# Modelado probabilístico del alumno en Entornos Inteligentes de Resolución de Problemas Educativos

**Tesis Doctoral**

*Presentada para optar al título de Doctor en Informática por:*

**D. Jaime Gálvez Cordero**

*Dirigida por:*

**el Dr. D. Eduardo Guzmán De los Riscos**
**y**
**el Dr. D. Ricardo Conejo Muñoz**

**Departamento de Lenguajes y Ciencias de la Computación**
**Escuela Técnica Superior de Ingeniería Informática**
**Universidad de Málaga**


**Málaga, noviembre de 2012**

Documento maquetado con TEXIS v.1.0+.

Copyright © Jaime Gálvez Cordero
Este trabajo ha sido financiado por la Consejería de Innovación, Ciencia y Empresa. Junta de Andalucía (España), P07-TIC-03243.

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

El Dr. D. Eduardo Guzmán De los Riscos, Profesor Titular de Universidad, y el Dr. D. Ricardo Conejo Muñoz, Catedrático de Universidad, ambos pertenecientes al Área de Lenguajes y Sistemas Informáticos de la E.T.S. de Ingeniería Informática de la Universidad de Málaga,

Certifican que,

D. Jaime Gálvez Cordero, Ingeniero en Informática, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo su dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

***Modelado probabilístico del alumno en Entornos Inteligentes de Resolución de Problemas Educativos***

Revisado el presente trabajo, estiman que puede ser presentado al tribunal que ha de juzgarlo, y autorizan la presentación de esta Tesis Doctoral en la Universidad de Málaga.

Málaga, noviembre de 2012

Fdo.: Dr. D. Eduardo Guzmán De los Riscos  Fdo.: Dr. D. Ricardo Conejo Muñoz

*A todas aquellas personas que*
*han hecho este trabajo posible*

# Agradecimientos

*La gratitud es la memoria del corazón*
Lao-tsé (570 a.C. - 490 a.C.)

Como reza la cita que encabeza esta sección, la gratitud es la memoria del corazón. De acuerdo con esta metáfora, agradecer es recordar. Es por ello, que debo recordar a todos aquellos que han ayudado a que este trabajo haya llegado hasta aquí.

En primer lugar a mis directores, el Dr. D. Ricardo Conejo y el Dr. D. Eduardo Guzmán, los cuales han dirigido y supervisado el trabajo realizado, dando sabios consejos que han hecho que esta tesis haya sido posible. Especialmente estoy en deuda con Edu, desde que comenzó nuestro periplo con el proyecto fin de carrera de la diplomatura de Informática; seguido del de la licenciatura; después con la tesina del máster de postgrado; hasta ahora, como director de esta tesis. Su labor ha sido la de un amigo, de igual a igual, que se ha involucrado en cada trabajo realizado, buscado en todo momento la manera de ayudar y hacer que este trabajo viera la luz. La gratitud que quiero transmitir no puede reflejarse solamente con estas palabras. Además, el hecho de que esta opinión sea algo compartido y generalizado entre los compañeros que trabajamos con él, demuestra que no es un trato preferente, sino una forma de ser, por lo que me siento afortunado de haber podido trabajar a su lado.

A mi familia agradezco el apoyo que me han dado constantemente. De forma más cercana, mis padres y mi hermano, pero también mis tíos y primos han contribuido a ello. Su ánimo durante la realización de este viaje ha sido más importante que cualquier otro, por venir de donde y de quien viene. Especialmente agradezco a mi madre que ha vivido el desarrollo de esta tesis como si fuera suya propia. Todos ellos han sufrido de una forma u otra las exigencias de realizar una tesis.

No puedo olvidarme de las personas que ya no están. Algunas por sus palabras de ánimo y sus altas expectativas sobre mí, las cuales han servido de motivación para superar los obstáculos encontrados. Otras por su ayuda, ya haya sido directa o indirectamente, en forma de inspiración y aliento para terminar.

Por su puesto, debo agradecer a una segunda familia durante este tiempo: mis amigos del 3.3.10. En las muchas horas que hemos compartido en el laboratorio ha habido un ambiente excelente que ha servido enormemente para sobrellevar las tareas que tocaban. Han sido unos años en los que la unión se ha visto reflejada también en multitud de experiencias fuera del laboratorio. Todo esto ha influido positivamente en el desarrollo de esta tesis. No quiero mencionar a todos, por no dejarme a nadie, puesto que han sido muchos los que han pasado por allí, pero sí quiero destacar especialmente a mi gran amigo Enrique, pues durante todo este tiempo hemos compartido las experiencias, ya no sólo de hacer una tesis, sino también de la vida; y a Martyna por sus revisiones

contra-reloj que han contribuido a la internacionalización de esta investigación.

Como no, también quiero agradecer a mis amigos de fuera del 3.3.10 que han aportado un grano de arena y han seguido en mayor o menor medida el desarrollo de este trabajo. Sin restar importancia a ninguno, agradezco especialmente a Juan Antonio por animarme desde que comenzamos los estudios universitarios hasta el día de hoy. No quiero dejar fuera de estas líneas a amigos que desde diferentes partes del mundo han servido de apoyo. Especialmente agradezco a Sisca, pues también ha compartido gran parte de las vicisitudes de este viaje conmigo.

Como parte de la tesis tuve el honor de realizar una estancia en Nueva Zelanda con uno de los grupos más importantes en el ámbito que abarca el trabajo realizado. Aquí, debo agradecer en primer lugar a la Dra. Tanja, por darme la oportunidad de trabajar con ella y su grupo, por acogerme desde que pisé Christchurch y por preocuparse de que todo fuera bien mientras estaba allí. Agradezco especialmente a Moffat, pues desde el primer momento, hasta el final, me trató como a un amigo de toda la vida. Sus charlas, los viajes compartidos y su trato hicieron que mi tiempo allí fuera como si estuviera en casa. También doy las gracias a Sagaya, por sus ánimos. En general, agradezco a todos los miembros del grupo ICTG de la Universidad de Canterbury por tratarme como si fuera uno más del grupo. Los diferentes problemas, principalmente asociados con las fuerzas de la naturaleza, el tiempo que jugaba en nuestra contra y otros muchos a los que nos tuvimos que enfrentar para llevar a buen término la conferencia AIED 2011, hicieron que nos uniéramos aún más y que la estancia fuera inolvidable.

En el hilo de las estancias realizadas, también agradezco a la Dra. Susan por permitirme realizar una estancia en la Universidad de Birmingham, cuyo grupo también destaca en la comunidad científica relacionada con la temática de esta tesis. La acogida brindada por ella, así como por parte de los miembros de su equipo con los que pude trabajar hizo mucho más llevadera mi visita a la ciudad inglesa.

Otra parte de mi agradecimiento va destinada a los alumnos de proyectos fin de carrera, cuyo trabajo ha servido para desarrollar varias partes de esta tesis. Igualmente, agradezco a aquellos compañeros que han colaborado con la investigación de alguna forma, ya sea mediante consejos o cediéndonos su tiempo y sus clases para que realizáramos los experimentos necesarios. También, agradezco a los doctores / doctoras que han aceptado formar parte del tribunal evaluador o que han colaborado elaborando informes externos, pues su participación contribuye a la finalización de este trabajo.

No puedo olvidarme tampoco de los compañeros científicos / investigadores de diversos países que he conocido en las conferencias a las que he tenido la suerte de asistir. Los consejos y palabras brindados por estas personas, tanto las que contaban con la experiencia de terminar un trabajo como este, como las que perseguían el mismo destino, resultaron alentadores. Es especialmente reconfortante encontrarse con personas que trabajan en algo parecido y que siguen tu trabajo desde otras partes del mundo.

Por último, pero no menos importante, me gustaría agradecer a la Junta de Andalucía por haberme concedido una beca de investigación (Ref. P07-TIC-03243), la cual me permitió realizar este trabajo. En relación con esta beca quiero mencionar la tarea del resto de becarios "excelentes" de la convocatoria 2007 de la universidad de Málaga. Con el grupo formado hemos conseguido cada objetivo propuesto, defendiendo nuestros derechos y movilizando a toda Andalucía. En especial, doy las gracias a los doctores Francisca y Víctor por su ayuda y por las experiencias compartidas.

A TOD@S, muchas gracias de corazón.

# Resumen

*Experiencia es lo que consigues*
*cuando no consigues lo que quieres*

Randy Paush (1960 - 2008)

Uno de los instrumentos de evaluación formal más extendidos es la *Teoría de Respuesta al Ítem* (TRI), una teoría psicométrica que se centra en las propiedades individuales de las preguntas, también conocidas como ítems. El elemento de la TRI que permite llevar a cabo la evaluación del alumno es una función de densidad que relaciona el conocimiento con la probabilidad de responder correctamente cada ítem y que se denomina *Curva Característica del Ítem* (CCI).

Aunque los ítems son el elemento que da la potencia a los tests para medir el conocimiento, éstos tienen una limitación importante. Debido a su forma, éstos normalmente proporcionan opciones que buscan medir el conocimiento asociado a conceptos teóricos o hechos, más conocido como *declarativo*. Cuando se trata de evaluar el conocimiento en tareas que requieran de la resolución de un problema o de la elaboración de una respuesta compleja, los ítems sólo proporcionan mecanismos para llevar a cabo tareas simples que no son más complejas que la ordenación o emparejamiento de elementos. En este sentido, no permiten la interacción apropiada para evaluar tareas complejas y no son el instrumento adecuado para diagnosticar el conocimiento del alumno en este tipo de dominios, llamados *procedimentales*.

Los *Sistemas Tutores Inteligentes* (STI) son entornos de aprendizaje en los que el proceso de instrucción se adapta a las necesidades del estudiante, empleando para ello técnicas de Inteligencia Artificial y de Psicología. De entre el amplio espectro de STI existentes hoy en día, están los denominados *Entornos Inteligentes de Resolución de Problemas* (EIRP) en los cuales los alumnos aprenden (y/o son evaluados) a través de la resolución de tareas complejas. En este tipo de entornos existen diversos paradigmas que establecen cómo modelar al alumno y cómo realizar el proceso de adaptación de la instrucción, de entre los cuales, uno de los más importantes es el *Modelado Basado en Restricciones* (MBR). El principal problema de este paradigma es la base sobre la que se asienta el proceso instructivo y de adaptación: la forma de diagnosticar el conocimiento del alumno, la cual, salvo alguna excepción, se basa en heurísticos.

Para solucionar las debilidades anteriormente mencionadas, esta tesis propone un modelo de evaluación formal en dominios procedimentales que se particulariza y formaliza usando el paradigma MBR con la TRI. Esta combinación es posible gracias a una similitud encontrada entre los elementos principales de las dos áreas usadas: las preguntas en los sistemas de tests, y las restricciones en los tutores MBR. Las restricciones pueden usarse para medir el conocimiento al representar evidencias correctas o

incorrectas, como sucede con las preguntas. Partiendo de esta analogía, se extienden los elementos típicos de evaluación mediante tests a sistemas MBR donde el alumno resuelve problemas. Aunque el modelo ha sido particularizado, éste puede generalizarse para diferentes paradigmas de modelado del alumno sobre tareas complejas en los que se puedan obtener evidencias del conocimiento y éstas puedan usarse para evaluar al alumno.

Además del uso del modelo para la evaluación del conocimiento, se proponen diferentes estrategias y procedimientos para un uso formativo del alumno. Las estrategias propuestas están basadas en la adaptación que puede realizarse mediante la TRI y que tiene su mayor exponente en los *Test Adaptativos Informatizados* (TAI). Este tipo de tests usan la TRI para determinar el ítem que más se adapta al nivel del alumno, haciendo que el cálculo del conocimiento sea más eficiente para obtener la misma fiabilidad que un test tradicional con un número predeterminado de preguntas. Dado que en el MBR una de las componentes principales involucrada en la instrucción del alumno es la adaptación y partiendo de la analogía entre las preguntas y las restricciones, se ha podido extender los mecanismos de adaptación de los TAI a sistemas MBR.

Para estudiar empíricamente la validez e idoneidad del modelo de evaluación se han implementado varios STI, los cuales han permitido la evolución y depuración del modelo. Este desarrollo del modelo ha tenido como fruto un marco de trabajo que ofrece servicios de evaluación a EIRP externos. De esta forma, cualquier sistema que proporcione la evidencia adecuada puede realizar la evaluación en dominios procedimentales.

Con el modelo de evaluación que esta tesis presenta se propone una forma de paliar la limitación de la evaluación formal en sistemas de tests mediante el uso de EIRP. A la vez, se cubren las debilidades existentes en sistemas MBR reemplazando los heurísticos por una técnica bien fundamentada de evaluación.

# Índice general

# Lista de figuras

# Lista de tablas

# Lista de definiciones

# Lista de símbolos

# Lista de acrónimos

CAT ..... *Computerized Adaptive Test*

CBM ..... *Constraint-Based Modeling*

CCC ..... *Constraint Characteristic Curve*

CCCR ..... *Curva Característica Complementaria de la Restricción*

CCI ..... *Curva Característica del Ítem*

CCIC ..... *Curva Característica de un Ítem Compuesto*

CCP ..... *Curva Característica de un Problema*

CCR ..... *Curva Característica de la Restricción*

CCT ..... *Curva Característica del Test*

CICC ..... *Composed Item Characteristic Curve*

CTT ..... *Classical Test Theory*

DBE ..... *Diseño Basado en la Evidencia*

ECD ..... *Evidence Centered Design*

EIRP ..... *Entorno Inteligente de Resolución de Problemas*

FII ..... *Función de Información del Ítem*

FIR ..... *Función de Información de la Restricción*

FIT ..... *Función de Información del Test*

ICC ..... *Item Characteristic Curve*

IRT ..... *Item Response Theory*

ITS ..... *Intelligent Tutoring System*

KT ..... *Knowledge Tracing*

MBR ..... *Modelado Basado en Restricciones*

MT ..... *Model Tracing*

OCCC ..... *Opposite Constraint Characteristic Curve*

PCC ..... *Problem Characteristic Curve*

PSLE ..... *Problem Solving Learning Environment*

SPP ..... *Selección por Problema Problemático*

SRP ..... *Selección por Restricción Problemática*

STI ..... *Sistema Tutor Inteligente*

TAC ..... *Test Administrador por Ordenador*

TAI ..... *Test Adaptativo Informatizado*

TC ..... *Traza del Conocimiento*

TCT ..... *Teoría Clásica de Tests*

TM ..... *Traza del Modelo*

TRI ..... *Teoría de Respuesta al Ítem*

# Parte I

# Introducción

En esta primera parte se introduce la problemática que motiva el trabajo de investigación realizado en esta tesis y se plantean los objetivos perseguidos para intentar paliar dicha situación.

# Capítulo 1

# Introducción

*La ciencia no es sino una perversión de sí misma a menos que tenga como objetivo final el mejoramiento de la humanidad*

Nikola Tesla (1856 - 1943)

**Resumen:** En este capítulo se detallan la motivación principal y los objetivos perseguidos con la investigación realizada. También, los contenidos de esta memoria son resumidos al final del capítulo.

La forma de transmitir los conocimientos de una generación a otra por medio de la enseñanza ha cambiado en los últimos años. En los tiempos que corren, el uso del computador se torna una herramienta indispensable en la labor de todo aprendiz. El abanico de posibilidades que éstos ofrecen facilita enormemente tanto la tarea del alumno como la del docente. En particular, los sistemas educativos a través del ordenador poseen una serie de características que los han convertido en los medios más utilizados en la enseñanza moderna. Estos sistemas representan el contexto más general en el que se enmarca el trabajo de investigación presentado en esta tesis.

## 1.1. Motivación

De la aplicación de técnicas de Inteligencia Artificial a los sistemas educativos surgen los denominados *Sistemas Tutores Inteligentes*(STI) (Nkambou et al., 2010), los cuales personalizan el proceso de instrucción adaptándolo a las necesidades del alumno de forma análoga a como lo haría un profesor. Estudios como el llevado a cabo por Bloom (1984) y numerosos trabajos previos en el campo (Polson y Richardson, 1988; Self, 1999; Murray, 1999) avalan la instrucción proporcionada por estos sistemas como una forma de aprendizaje más efectiva que la instrucción tradicional profesor-alumno. Shute y Psotka (1996) afirman que la inteligencia de estos sistemas radica principalmente en la determinación de los conocimientos del alumno, lo que permite determinar las fortalezas y debilidades del alumno que permitirán una adaptación de la instrucción más acorde con sus necesidades. Es en este punto donde toman importancia las palabras *evaluación* y *diagnóstico*.

El término evaluación es una palabra muy genérica, puesto que tiene usos muy diferentes y se puede aplicar a diferentes elementos pertenecientes a ámbitos muy variados. Según su acepción más general, la *Real Academia de la Lengua* (RAE) define la evaluación como "la acción de evaluar", que a su vez define evaluar como "Estimar, apreciar, calcular el valor de algo". Esta definición queda abierta a diferentes formas de evaluación. Para el tema que trata esta tesis, el ámbito de la evaluación educativa es el que se usará de ahora en adelante. En este área, una definición más precisa, adaptada a partir de (AFT (American Federation of Teachers) et al., 1990; Huba y Freed, 2000; Tissot, 2004), establecería la evaluación como el proceso de obtención de información con el fin de determinar los conocimientos de un estudiante, sus capacidades y competencias.

### 1.1.1. Tipos de evaluación

Independientemente del ámbito, el tipo o la forma, la evaluación puede ser agrupada en dos grandes categorías, que hacen referencia a la objetividad y las bases sobre las que se asienta la evaluación.

- Por un lado estaría la *evaluación informal* o *asistemática* (Navarrete et al., 1990), que se caracteriza por ser superficial, sin un plan elaborado, con una validez y fiabilidad no verificada y de un alto grado subjetivo. En la definición más general de la evaluación, este tipo es utilizado con frecuencia en la vida diaria para la toma de decisiones. En el ámbito educativo, es una valoración indicativa del nivel del estudiante en una materia de interés, y bajo las mismas propiedades mencionadas anteriormente.

- Por otro lado, se encuentra la *evaluación formal* o *sistemática*. Una definición que no puede faltar al hablar de evaluación formal es la dada por Scriven (1967), el cual plantea la evaluación como un proceso sistemático que persigue emitir un juicio de valor fundamentado objetivamente, basándose en un conjunto específico de valoración de metas que proporcionen una valoración comparativa o numérica, siendo necesario justificar a) los instrumentos o el criterio de recopilación de datos, b) las valoraciones y c) la selección de los objetivos. Según el autor, la evaluación es mejor si el evaluador desconoce los objetivos y es independiente, lo cual favorecerá la objetividad del juicio emitido. Esta definición de Scriven fue elaborada de forma general, sin especificar el ámbito de aplicación, pero se puede aplicar al entorno educativo considerando el alumno como el objeto de la evaluación y el aprendizaje como el objetivo general.

Según Scriven (1967), y ampliamente revisado por Taras (2005, 2007), la evaluación formal como parte del proceso de enseñanza y aprendizaje se divide en dos tipos: la evaluación formativa y la sumativa. Posteriormente, Bloom et al. (1971) agregan la evaluación diagnóstica a estas dos distinciones. Cada una de ellas no son excluyentes, sino que se complementan y atienden a los diferentes propósitos de la evaluación, las cuales responden a la pregunta *para qué* y está relacionado con la oportunidad *cuando* se evalúa:

- *Evaluación diagnóstica* o también llamada *inicial*: Se realiza para predecir un rendimiento o para determinar el nivel de aptitud previo al proceso educativo. Busca determinar cuáles son las características del alumno previo al desarrollo del programa, con el objetivo de ubicarlo en su nivel, clasificarlo y adecuar individualmente el nivel de partida del proceso educativo.

- *Evaluación formativa* o *de proceso*: Su objetivo es proporcionar la información para ajustar el proceso de enseñanza y el aprendizaje mientras éste está teniendo lugar. De acuerdo con Scriven (1967); Ramaprasad (1983); Sadler (1989); Black y Wiliam (1998), debe haber un refuerzo que indique las carencias existentes entre el nivel actual del elemento objeto de evaluación y el nivel requerido. De esta forma, este tipo de evaluación posibilita un doble refuerzo. Por un lado, indica al alumno su situación respecto de las distintas etapas por las que debe pasar para realizar un aprendizaje determinado; y por el otro, indica al profesor cómo se desarrolla el proceso de enseñanza y aprendizaje, así como los mayores logros y dificultades de los que aprenden.

- *Evaluación sumativa* o *final*: Ésta es un juicio que encapsula el compendio de toda la evidencia hasta cierto punto en el tiempo (de ahí el uso del término "summa" como parte del nombre). Tiene la estructura de un balance, que se realiza después de un período de aprendizaje en la finalización de un programa o curso, y busca calificar en función de un rendimiento, otorgar una certificación, determinar e informar sobre el nivel alcanzado.

Diversos autores (Black y Wiliam, 1998; Stiggins y Chappuis, 2006; Taras, 2007) se han referido en la literatura a las dos categorías de Scriven como *evaluación para el aprendizaje*, haciendo referencia a la evaluación formativa, en la que la evaluación se utiliza como una herramienta *para* mejorar el proceso de aprendizaje; y *evaluación del aprendizaje*, en referencia a la evaluación sumativa en la que el objetivo en sí es la evaluación *del* proceso de aprendizaje. Todos ellos parecen coincidir en que la evaluación sumativa puede ser únicamente sumativa, si la evaluación acaba con el juicio. Sin embargo, la evaluación formativa no puede existir por sí sola, ya que debe ir precedida de un juicio sumativo, que puede ser explícito o implícito. Taras (2005) además, puntualiza que toda evaluación comienza con una evaluación sumativa y la evaluación formativa es, de hecho, una evaluación sumativa con un refuerzo que es usado por el alumno.

Otra jerarquía sobre los tipos de evaluación se puede realizar atendiendo al modelo que se propone, el cual define qué se utiliza para realizar la evaluación y cuál es el proceso. Esta categorización responde al cómo se evalúa y distingue dos tipos de modelos:

- Los modelos *cuantitativos*, también llamados *objetivos* o *tecnológicos* parten de las ideas de Tyler (1942), el cual busca comparar unos objetivos preestablecidos con los logros alcanzados para determinar la congruencia entre ambos. Estos modelos evolucionan gracias a trabajos como el de Scriven (1967), que cuestionan el uso de las metas y dan mayor importancia a la emisión de un juicio de valor que podrá ser usado para paliar los déficits educativos.

- Los Modelos *cualitativos* son la evolución de los anteriores. Éstos señalan que se han de calcular los resultados finales junto al desarrollo de los procesos de enseñanza y aprendizaje para poder mejorarlos. Esto quiere decir que, al igual que el aprendizaje de los alumnos, también se debe evaluar la labor del profesor, los métodos didácticos, los materiales, etc. De esta forma, además de la medición cuantitativa, se tienen en cuenta elementos cualitativos que también influyen en el aprendizaje. Como inconveniente del modelo, destaca la falta de progreso

metodológico, la poca fiabilidad de los métodos para la obtención de datos, y su falta de objetividad.

Existen muchas otras dimensiones que permiten agrupar los tipos de aprendizaje, de entre las cuales, podemos destacar sin entrar en mucho detalle, al no ser tan relevantes para este trabajo, las siguientes: A) De acuerdo al criterio de evaluación, ésta puede ser *normativa*, si establece el juicio de un alumno en función del nivel del grupo al que pertenece; o *criterial* (también llamada *integral*), la cual pretende corregir la arbitrariedad de la evaluación normativa mediante el establecimiento de unos criterios externos, claros, y determinados, en funciones de los cuales se realizará la evaluación. B) Según el agente que interviene, se distingue la *autoevaluación*, si el sujeto que realiza la evaluación es también el objeto de la misma; la *evaluación externa*, si se realiza por agentes no vinculados al proceso de aprendizaje; y la *coevaluación*, la cual es una evaluación multilateral y combinada sobre las actividades llevadas a cabo por un grupo de alumnos.

### 1.1.2. Tipos de conocimiento

Como se mencionaba anteriormente, la evaluación educativa se aplica para determinar el conocimiento del estudiante. Aunque existen muchas clasificaciones del conocimiento, se pueden distinguir principalmente dos formas de conocimiento diferentes (Winograd, 1975; Cohen y Squire, 1980; Cauley, 1986):

1. El conocimiento *declarativo* o también conocido como *conceptual*, explica *qué* es algo. Este tipo de conocimiento se asocia a hechos o conceptos teóricos que el alumno posee y se puede decir o declarar. El término declarativo no se refiere al significado, sino al hecho de que la información puede ser representada simbólicamente.

2. El conocimiento *procedimental* o *cognitivo*, indica el *cómo* se hace algo y se asocia al proceso de acción o serie de acciones para la consecución de algún tipo de meta. El término procedimental se deriva de los procedimientos que se ejecutan para realizar una tarea.

Los tipos de conocimiento están íntimamente relacionados durante el aprendizaje de un estudiante pues tienen lugar en paralelo y se van modificando para dar lugar a otro nuevo. Este proceso se puede agrupar en varias fases o niveles: en primer lugar, a partir de alguna fuente de información el estudiante adquiere conocimiento declarativo o teórico (VanLehn, 1996); en una fase posterior, parte del conocimiento adquirido se transforma en procedimental, el cual se va reafirmando o corrigiendo con la práctica (Anderson, 1983); a un nivel posterior se produce un proceso de especialización que genera nuevo conocimiento a partir del análisis y deducción lógica resultante de la práctica (Piaget, 1970). Este proceso describe de forma muy general a qué parte del aprendizaje se corresponde. La relación exacta es mucho más compleja. De hecho, en la actualidad, los intentos por identificar la relación y la dependencia existente entre ellos, continúa siendo una cuestión sin resolver (Schneider y Stern, 2010).

Durante el trabajo presentado en esta tesis se utilizará el término declarativo o procedimental para describir la naturaleza de las materias / temas objetos de evaluación, o los dominios de aprendizaje. De esta forma, una materia o dominio es declarativa si el conocimiento que está asociado es de este tipo principalmente; y procedimental

si el dominio o materia contiene tareas o procedimientos que, inherentemente, están asociados con conocimientos procedimentales.

### 1.1.3. Instrumentos para la evaluación formal

Para llevar a cabo la evaluación, siempre es necesario un instrumento de medida a través del cual se recogerá la evidencia sobre el estudiante que permitirá determinar las aptitudes del mismo. De los muchos métodos e instrumentos de medida existentes, la investigación aquí presentada se centra en los asociados a la evaluación formal. En este ámbito, la herramienta por excelencia de medida, dado que es la más fiable y objetiva, son los *tests*.

Aunque el término test mental es acuñado por Cattell y Galton (1980), el primer test propiamente de inteligencia es creado a principios de siglo por los psicólogos franceses Binet et al. (1913). Su desarrollo fue impulsado por la armada de Estados Unidos durante la I guerra mundial para determinar si los reclutas eran aptos para el servicio (Boake, 2002). Aunque hoy día son utilizados ampliamente en todo el mundo, es principalmente en este país donde su uso está más extendido, formando parte del sistema educativo en procesos de promoción tales como el acceso a la universidad.

Los tests son instrumentos que intentan medir alguna de las habilidades o rasgos de una persona, tales como la inteligencia, grado de rendimiento, conocimiento, aptitud, etc. mediante una serie de preguntas denominadas, de forma genérica, *ítems*. En general, tienen la ventaja de que el tiempo requerido para aplicarse es mucho menor en comparación con otros instrumentos de evaluación. Además, normalmente se diseñan y administran mediante el ordenador, lo que reduce el tiempo requerido para su desarrollo y evaluación. En capítulos posteriores se ampliará información sobre este instrumento.

Dependiendo del grado de objetividad y formalidad en la construcción de sus ítems, los tests pueden ser estandarizados o no. Los no estandarizados suponen el uso más extendido en la educación. Normalmente son diseñados por profesores de acuerdo a unos objetivos pedagógicos concretos relacionados con la materia que desean evaluar. Su ventaja es la facilidad de aplicación y el bajo coste de desarrollo, tanto temporal como económico. El principal inconveniente de estos tests radica en el sujeto que los crea, puesto que la personalidad subjetiva del profesor-evaluador, sus prejuicios, opiniones, comprensión, e incluso sus caprichos e impulsos, prevalecen sobre la evaluación objetiva del evaluado (Pani, 2007). Esto conlleva que las propiedades de los tests se vean influenciadas negativamente. Las características que este tipo de tests debería cumplir para considerarse como apropiado para la evaluación son las siguientes:

- Validez: Esto quiere decir que el test mide lo que se supone que debe medir.
- Fiabilidad : Un test es considerado fiable si al realizarse de nuevo por el mismo conjunto de estudiantes, bajo las mismas circunstancias, el resultado medio es el mismo.
- Objetividad: Esta característica se encuentra en el hecho de que si el test, es evaluado por diferentes personas, el resultado será el mismo. En otras palabras, la evaluación no se ve afectada por el carácter subjetivo del evaluador.
- Comprensión: Un buen test debería incluir ítems las diferentes áreas de la materia siendo evaluada en él.

- Simplicidad: Significa que el test debería ser escrito en un lenguaje claro, correcto y evitando instrucciones ambiguas. Es importante que el método de preguntar sea lo más simple posible y que aún así, evalúe lo que debe evaluar.
- Puntuabilidad: Esta característica implica que cada ítem en el test tiene su propia puntuación, la cual tiene una influencia determinada en la puntuación total del test.

Los tests estandarizados sí que intentan cumplir con las características mencionadas anteriormente. Para ello, son diseñados a conciencia, normalmente por especialistas en evaluación, con un contenido escrutado y un proceso de administración y evaluación bajo ciertas normas o estándares (Hopkins y Stanley, 1981). El problema de éstos frente a los no estandarizados radica en que la evaluación proporcionada es principalmente de carácter sumativo, resultando difícil determinar los cambios que son requeridos para mejorar el aprendizaje. Además, al estar diseñados normalmente por organismos externos, es muy probable que no cumplan los objetivos educativos de las instituciones donde se administran y que conlleven un gasto económico mucho mayor.

### 1.1.4. Limitaciones de los tests en la evaluación

Uno de los grandes inconvenientes que se pueden mencionar sobre los tests se encuentra en el elemento que precisamente le proporciona su potencial, los ítems. Éstos imponen una limitación en cuanto al tipo del conocimiento que son capaces de evaluar, el cual está asociado con la naturaleza de la materia de evaluación.

El problema se encuentra en que los ítems, y consecuentemente los tests, están diseñados para obtener información mediante tareas simples, la cual se asocia principalmente al conocimiento declarativo. En este sentido, no permiten la interacción apropiada para la extracción de información relacionada con el conocimiento procedimental, la cual sería necesaria en la evaluación de tareas complejas. Por este motivo, la principal limitación de los tests se encuentra en que no son el instrumento adecuado para diagnosticar el conocimiento del alumno en dominios procedimentales.

Esta limitación es discutida en el trabajo de Marzano (1990), cuyo título resume perfectamente la problemática a la que nos referimos: *"Pregunta: ¿Miden los tests estandarizados habilidades cognitivas? Respuesta: No"*. De acuerdo a este trabajo, tan sólo un número muy reducido de operaciones cognitivas se usan normalmente en los tests y son operaciones simples como ordenar, comparar, representar. Cuando se trata de evaluar habilidades cognitivas, es necesario prestar atención no sólo al resultado dado, sino también a las acciones realizadas. La solución en sí tiene una naturaleza principalmente declarativa que puede mimetizar el verdadero conocimiento procedimental subyacente.

Este hueco evaluativo no está solamente presente en los tests, sino que, hasta la fecha, no se ha podido encontrar en la literatura existente ninguna metodología formal que permita realizar evaluación en los instrumentos educativos asociados a dominios procedimentales con cierta complejidad. Estos instrumentos son típicamente entornos educativos en los que el alumno puede aplicar el conocimiento procedimental mediante la resolución de problemas. En capítulos posteriores se dará más detalles de los mismos, a los que, de momento, nos referiremos como *Entornos Inteligentes de Resolución de Problemas* (EIRP).

La limitación explicada en este apartado, tanto en los tests para la evaluación, como en los EIRP da lugar a la principal motivación del trabajo de tesis que se presenta

en este documento. Dado que el conocimiento procedimental se sitúa en un nivel del proceso de aprendizaje superior al del declarativo, queda patente la importancia del mismo. La necesidad de estudiar si es posible paliar la limitación detectada, mediante el rigor de una metodología sistemática de evaluación, supone la principal justificación y el eje principal que mueve el trabajo realizado.

## 1.2. Objetivos

Tal y como se ha comentado, y como se detallará en capítulos posteriores, los sistemas existentes que permiten al alumno interactuar en tareas complejas de dominios procedimentales, carecen de mecanismos de evaluación formales para estimar el conocimiento del estudiante. Normalmente, las estimaciones realizadas se basan en heurísticos y fórmulas que no tienen una base bien fundamentada. Puesto que una evaluación formal permite modelar al alumno de una forma más precisa, la adaptación que el sistema puede proporcionar sería más efectiva al disponer de unas necesidades más fieles a la realidad. Por este motivo, se presenta como fundamental la elaboración de mecanismos de evaluación bien fundamentados, que garanticen un diagnóstico objetivo y fiable.

El objetivo principal de esta tesis es la elaboración de un modelo de evaluación del conocimiento en su dimensión general (de Jong y Ferguson-Hessler, 1996). Esto quiere decir, que el modelo no sea específico para un dominio concreto sino que pueda ser aplicado de forma genérica a diversos dominios. Concretamente, el modelo se centrará en dominios educativos de tipo procedimental con cierta complejidad, intentando así, cubrir el hueco de la evaluación educativa comentado anteriormente. Debido a la naturaleza de los métodos, procedimientos, y conceptos involucrados en este tipo de dominios, el proceso de evaluación se realizará a partir de tareas asociadas a la materia en cuestión. De esta forma, se busca una evaluación procedimental del conocimiento, que no una evaluación exclusiva del conocimiento procedimental. Esto quiere decir que, aunque se tratarán dominios procedimentales, la evaluación no va medir un tipo de conocimiento específico, ya que a la hora de elaborar una respuesta, el conocimiento declarativo también está involucrado. Es por ello que esta evaluación pretenderá evaluar el conocimiento del alumno de forma general en dominios procedimentales.

Las características que deberá cumplir el modelo, en relación con los tipos de evaluación mencionados en el apartado 1.1.1, son las siguientes:

- Debe ser un modelo de evaluación formal o sistemático, en el que la emisión de un juicio sobre el nivel estimado del alumno sea objetiva y se asiente sobre una base bien fundamentada de evaluación.
- Además, debe ser cuantitativo. Según este tipo de modelos de evaluación, se producirá un valor estimado del conocimiento sin tener en cuenta las características adicionales propias de los modelos cualitativos, los cuales se dejan fuera del trabajo de esta tesis por su subjetividad y por su falta de rigor.
- En cuanto a la finalidad, en primer lugar, la evaluación será sumativa. Con este tipo de evaluación, los mecanismos que se establezcan permitirán determinar lo que el alumno sabe en un momento determinado del tiempo. También, se estudiará cómo usar el modelo para proporcionar una evaluación formativa que permita usar la evaluación sumativa anterior para mejorar el proceso de aprendizaje de un alumno.

Este modelo dependerá claramente del instrumento de medida empleado, el cual, dadas las limitaciones vistas anteriormente en los sistemas de tests, consistirá en el uso de EIRP. Con este tipo de plataformas educativas se podrán llevar a cabo las tareas propias de los dominios procedimentales. La viabilidad de combinación de las técnicas formales existentes con los paradigmas de EIRP será crítica a la hora de llevar a la práctica el modelo teórico de evaluación. En este sentido, el segundo objetivo principal de la tesis será la implementación y la evaluación empírica de la viabilidad de uso práctico del modelo en diversos EIRP asociados a dominios concretos.

Como parte de la implementación del modelo de evaluación, se estudiarán las ventajas e inconvenientes que proporciona la utilización de las técnicas bien fundamentadas de evaluación en EIRP, tanto para el objetivo de evaluación en sí, como para los paradigmas existentes en este tipo de entornos. En este sentido hay varias posibilidades de uso que van más allá de la simple evaluación, tal y como se verá en capítulos posteriores, que pueden ser objeto de estudio.

En un tercer objetivo, se intentará abstraer, a partir de la implementación buscada por el objetivo anterior, características comunes que surjan de la aplicación del modelo inicial. El objetivo es generalizar el modelo en la medida de lo posible, para que éste no sea dependiente del dominio donde se aplique. Como parte final de este objetivo se pretende construir un marco de trabajo genérico que permita llevar a cabo la evaluación formal de acuerdo al modelo desarrollado. Es decir, que sea capaz de emitir un juicio sobre el conocimiento del alumno en cualquier dominio procedimental sobre el que se quiera realizar una evaluación. Este marco de trabajo debe disponer de las herramientas necesarias para la construcción de futuros EIRP que quieran incluir los mecanismos de evaluación presentados en esta tesis. Se busca con esto fomentar el uso de esta nueva metodología en la comunidad científica.

Para lograr estos objetivos la investigación realizada ha seguido una metodología de trabajo basada en un enfoque desde lo más específico a lo más general (conocido normalmente como *Bottom-Up*). En este sentido, primeramente se han construido diferentes EIRP individuales sobre dominios procedimentales específicos, abarcando desde dominios simples hasta dominios con tareas complejas. Para los diferentes sistemas se ha buscado el estudio empírico utilizando datos de estudios realizados a partir del uso en clase con estudiantes reales.

Como paradigma de construcción de EIRP, se ha usado el *Modelado Basado en Restricciones* (MBR) (Ohlsson, 1994), por las características de eficiencia, simplicidad de aplicación, y las facilidades que posee, las cuales serán expuestas en detalle en capítulos posteriores. Sobre este paradigma se ha estudiado la extensión de los modelos asociados para contemplar el uso de la *Teoría de Respuesta al Ítem* (TRI) (Thurstone, 1925) como mecanismo de evaluación bien fundamentado. Etas dos técnicas representan los dos grandes pilares sobre los que se asienta la tesis, como se podrá apreciarse a lo largo de este documento. La mayor parte del peso se lo llevan los EIRP que usan el segundo pilar como una herramienta para llevar a cabo los objetivos planteados.

Como etapa final, a partir de los sistemas desarrollados, se han extraído características comunes y patrones presentes en los mismos, generalizándolas en el marco de trabajo CBMEngine, una componente que puede ser reutilizable por cualquier EIRP que desee, siempre realice una serie de pasos y siga un protocolo definido. Los detalles concretos de este marco de trabajo, las integraciones realizadas con diversos entornos de aprendizaje, así como también, las experiencias realizadas con estudiantes reales, se explicarán en capítulos posteriores.

## 1.3. Estructura del documento

La memoria de tesis que se presenta en este documento está compuesta de 8 capítulos, estructurados en seis partes que se desglosan resumidamente a continuación.

### Parte I: Introducción

En la primera parte, compuesta únicamente por este capítulo, se han introducido los diferentes tipos de conocimiento del alumno. En base a éstos, se ha introducido el concepto de evaluación educativa y la problemática de realizar la evaluación bien fundamentada de tareas en dominios procedimentales. A continuación se han explicado las características de los EIRP como herramienta para poder afrontar la problemática anterior. Finalmente, se han planteado los objetivos perseguidos en este trabajo.

### Parte II: Antecedentes

Esta segunda parte está formada por los capítulos 2 y 3. En el primero de ellos, se detallarán los diferentes paradigmas existentes para la construcción de EIRP, haciendo especial hincapié en el MBR por ser uno de los pilares básicos de esta tesis. Para este paradigma se explicarán sus fundamentos teóricos, sus características, componentes, y se presentará cada trabajo existente en esta área, con el fin de detectar problemas que pudieran ser paliados con el uso de una evaluación formal.

En el capítulo siguiente, se detallan las metodologías de evaluación más populares, haciendo hincapié en otro de los pilares fundamentales del trabajo realizado: la TRI. Para esta metodología se explicarán sus fundamentos, características, requisitos para poderse aplicar y modelos existentes. También se detallará su uso en sistemas de tests y las ventajas e inconvenientes que aportan a este tipo de sistemas. Además, se hará un breve repaso a otros campos relacionados y marcos de trabajo genéricos que se adecuan a la problemática que se intenta resolver relacionada con la evaluación en EIRP.

### Parte III: Propuesta

La tercera parte contiene dos capítulos. En primer lugar, en el capítulo 4 se desarrolla un modelo teórico para realizar una evaluación sumativa mediante la TRI para sistemas MBR. El modelo tendrá en cuenta las particularidades de los EIRP basados en el MBR para modelar, de forma similar a como se hace en sistemas de tests, una base teórica necesaria previa a la aplicación de los mecanismos de inferencia y adaptación que la TRI proporciona. Además, se darán unas pautas generales sobre cómo el modelo puede ser generalizado para usarse en otro tipo de EIRP que no utilicen el MBR.

A continuación, en el capítulo 5, se explica cómo extender el modelo de evaluación sumativa con las características necesarias para poder llevar a cabo una evaluación formativa. Es decir, cómo usar la TRI para proporcionar aprendizaje, tanto en sistemas de tests como en EIRP bajo el MBR. En este nuevo modelo de evaluación formativa se proponen un amplio abanico de métodos asociados con diversas estrategias de aprendizaje. El capítulo incluye una revisión sobre otras utilidades que la TRI proporciona al aplicarse a sistemas MBR.

### Parte IV: Implementación

En esta parte, constituida por el capítulo 6, se mencionan las herramientas en las que se han implementado los modelos de evaluación anteriores. Para cada una de ellas se introduce el dominio educativo sobre el que está basada y los elementos de su arquitectura que permiten realizar la implementación. El elemento destacado de este capítulo es un marco de trabajo que proporciona a EIRP los mecanismos necesarios para aplicar la metodología de evaluación desarrollada.

### Parte V: Experimentación

En esta parte se estudia, de forma empírica, la validez de los modelos anteriores. Aquí se explica cada uno de los experimentos realizados utilizando las herramientas mencionadas en el capítulo anterior para determinar la viabilidad de la metodología de evaluación.

### Parte VI: Conclusiones

Para finalizar, esta última parte está compuesta por un único capítulo, en el que se presentan las conclusiones del trabajo de tesis. Dentro de este capítulo se mencionan las aportaciones principales de la investigación, las limitaciones existentes, y las líneas de investigación que quedan abiertas.

### Parte VI: Apéndices

Esta parte está compuesta por dos apéndices. El apéndice A extiende la información sobre el principal marco de trabajo implementado en esta investigación. El objetivo es detallar el conjunto de servicios Web que permiten a un sistema externo usar el modelo desarrollado en esta tesis. Como elemento requerido para optar al título de doctor con mención Internacional, el apéndice B presenta en lengua inglesa un resumen de los contenidos de la tesis junto con la traducción literal del capítulo de conclusiones.

# Parte II

# Antecedentes

Esta parte trata las dos áreas fundamentales sobre las que se asienta este trabajo de tesis. Concretamente se revisan los paradigmas existentes para el modelado del alumno en entornos de resolución de problemas y las metodologías relacionadas con la evaluación formal del conocimiento.

# Capítulo 2

# Entornos inteligentes de resolución de problemas

*Me lo contaron y lo olvidé;*
*lo vi y lo entendí;*
*lo hice y lo aprendí*

Confucio (551 a.C. - 479 a.C.)

**Resumen:** Atendiendo a la naturaleza procedimental de los dominios sobre los que se busca emitir un juicio del conocimiento, en este capítulo se hace una revisión detallada de los diferentes paradigmas existentes en la construcción de sistemas educativos en este tipo de dominios.

Como herramienta para proporcionar los medios necesarios para que el alumno pueda realizar la interacción objetivo de esta tesis se han utilizado los *Entorno Inteligente de Resolución de Problemas* (EIRP). El término *Entorno de Resolución de Problemas* (ERP) puede tener diferentes significados y aplicarse a diferentes áreas. De forma general, en la Informática este término se refiere a un sistema de ordenador que proporciona todas las facilidades computacionales requeridas para resolver un tipo de problema determinado en un entorno en el que el usuario no requiere otro tipo de conocimiento para alcanzar una solución, más que el asociado con el proceso de resolución (Gallopoulos et al., 1994). En el área de los sistemas educativos por ordenador, donde se sitúa el trabajo de esta tesis, la literatura asociada añade la característica inteligente a los ERP, utilizándose en su lugar el término EIRP o ERP de aprendizaje (en inglés normalmente se utiliza *Problem Solving Learning Environment* (PSLE)). De esta forma el término EIRP se refiere a un STI que centra su interacción con el alumno en la resolución de problemas de un dominio particular. Podría incluso decirse que los STI son EIRP, puesto que la gran mayoría centran su interacción en la resolución de problemas.

Dependiendo del conocimiento procedimental o las habilidades involucradas en la resolución de problemas, Mitrovic y Weerasinghe (2009) proponen una clasificación de los EIRP. Ésta se expresa en base a dos dimensiones ortogonales: el dominio en el que se ubica el EIRP y las tareas que se realizan en él. Tanto los dominios como las tareas pueden ser bien definidos o débilmente definidos (del inglés *ill-defined*), los primeros dependiendo de la teoría o conocimiento declarativo subyacente, y los segundos dependiendo de la complejidad de los procedimientos asociados. La combinación de los

diferentes valores da cuatro categorías que se muestran en la figura 2.1: dominios bien definidos con tareas bien definidas (etiquetado en la imagen como "DBTB"), dominios bien definidos con tareas débilmente definidas (etiquetado con "DBTD"), dominios débilmente definidos con tareas bien definidas (etiqueta "DDTB"), y dominios débilmente definidos con tareas bien definidas (etiquetado como "DDTD"). Esta clasificación será usada, no sólo en este capítulo, sino también, a lo largo de este documento, como punto de referencia para clasificar los EIRP que se expliquen.

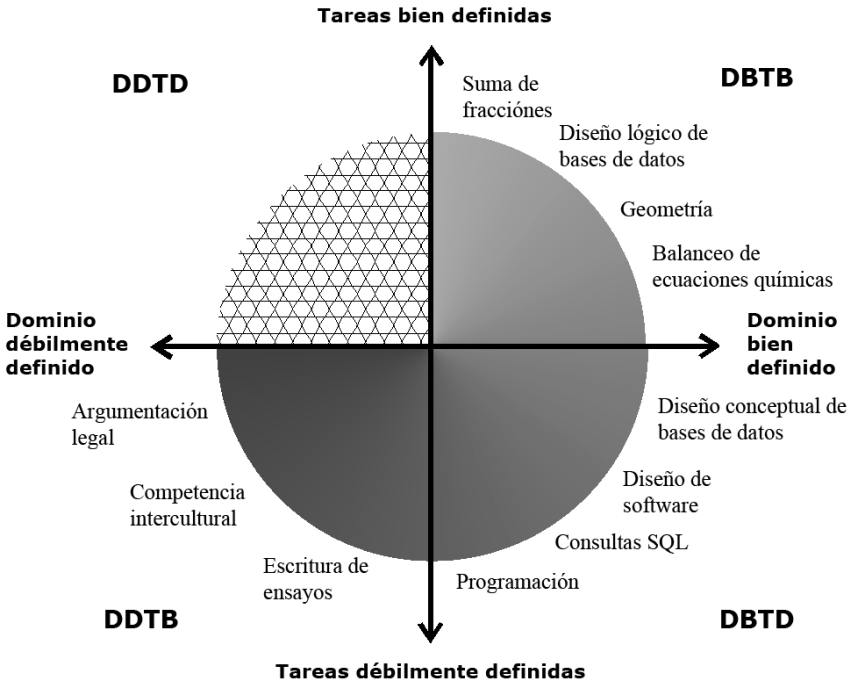



Figura 2.1: Espacio de dominios y tareas de instrucción.

Aunque el lector puede dar por supuesto que las tareas que se le plantean a un alumno en un EIRP son problemas, es necesaria una aclaración. La RAE define un problema como "el planteamiento de una situación cuya respuesta desconocida debe obtenerse a través de métodos científicos". Probablemente, el término español más apropiado para hablar de las tareas en un EIRP es "ejercicio", el cual se define en la RAE como "un trabajo práctico que en el aprendizaje de ciertas disciplinas sirve de complemento y comprobación de la enseñanza teórica". No obstante, dado que en la literatura relacionada con esta tesis se suele usar el término inglés *problem*, se toma por convenio el uso de la traducción literal al español. De esta forma, al hablar de problema, de ahora en adelante, se sobreentiende que se refiere a los ejercicios y tareas que pueden practicarse en un EIRP.

Este capítulo trata principalmente el paradigma base para la construcción de EIRP, haciendo primero una revisión del ámbito en el que se enmarca. En primer lugar, en la siguiente sección, se da una visión general sobre los STI y se presentan brevemente los paradigmas existentes que permiten a los STI realizar su función. De estos paradigmas dos destacan en el panorama actual y son tratados seguidamente. En la sección 2.2 se explican de forma global las características generales de los tutores cognitivos. La siguiente sección (2.3) explica el MBR. Dado que ésta supone uno de los dos pilares

básicos sobre los que se asienta esta tesis, se realiza una extensa y profunda revisión sobre el estado del arte asociado. Finalmente se muestran las conclusiones del capítulo.

## 2.1. Sistemas Tutores Inteligentes

De forma general, un *Sistema Tutor Inteligente* (STI) es un sistema cuyo objetivo es el de proporcionar una instrucción personalizada que ayude en el proceso de aprendizaje mediante el uso del ordenador. Estos sistemas son la evolución de los sistemas de instrucción asistida por ordenador o *Computer Assisted Instruction* (CAI). De hecho, antes de usar el término STI se utilizaba *Intelligent Computer Assisted Instruction* (ICAI) para referirse a la adicción de la inteligencia a los anteriores CAI. Los STI la combinan tres grandes disciplinas que se esquematizan en la figura 2.2: la Informática, la Psicología cognitiva, y la educación. Las principales áreas involucradas son la IA (*Inteligencia Artificial*), como base para hacer que los sistemas educativos se comporten como tutores humanos; la Psicología cognitiva, para dar explicación a los procesos mentales relacionados con el conocimiento; y la investigación educativa, que estudia la enseñanza y el aprendizaje humano.

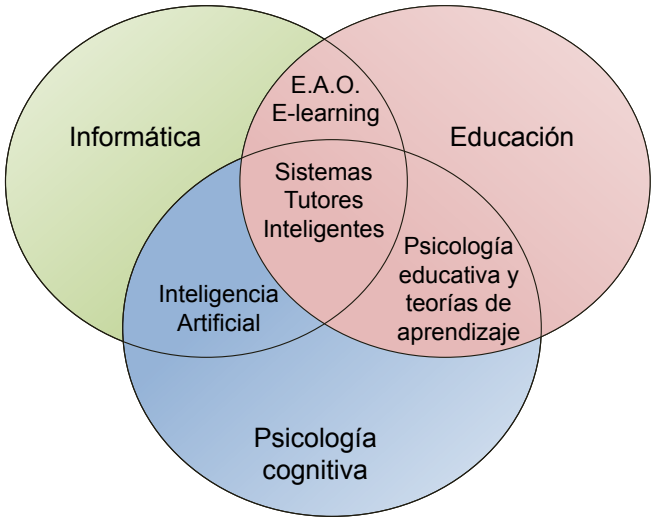



Figura 2.2: Disciplinas involucradas en los STI.

La dificultad clave de los STI reside en que la representación formal del conocimiento, heredada del campo de la IA, requiere modelos muy detallados y específicos del conocimiento del estudiante. Esta especificidad de los modelos es imposible de conseguir puesto que supone medir elementos no observables del conocimiento humano. Este problema es denominado por Mitrovic y Ohlsson (1999) como el *problema de la sobreespecificidad*. Aunque estudios como los de Ohlsson (1986); Self (1990); Holt et al. (1994) definen esta tarea como un problema inherentemente inmanejable, asumiendo que es complejo elaborar un modelo que se acople con exactitud a la realidad, aproximaciones a ésta son aceptadas ya que han probado ser útiles desde el punto de vista del aprendizaje (Self, 1990; Holt et al., 1994). De forma general, las técnicas de modelado funcionan porque limitan la especificidad de los modelos resultantes o porque restringen el escenario de instrucción de forma que se simplifica significativamente el

problema del modelado.

Aunque cada STI existente tiene sus particularidades, de forma general hay cuatro partes principales que se pueden destacar sobre el resto (Sleeman y Brown, 1982). Estas cuatro partes conforman la base utilizada por este tipo de sistemas realizar el proceso de enseñanza del alumno:

- Una *interfaz* mediante la cual se produce la interacción hombre-máquina. Cuando la interacción de la interfaz permite la resolución de problemas, entonces el STI es considerado un EIRP. La interfaz permite al sistema transmitir el conocimiento al alumno y obtener evidencias que serán usadas para estimar lo que el alumno sabe. Esta parte debería estar diseñada para que no fuese necesario otro conocimiento que el necesario con la materia que se enseña, de lo contrario puede hacer que haya una sobrecarga cognitiva (Sweller et al., 1998) y distraiga al alumno del proceso de aprendizaje.
- Un *módulo de dominio* o *módulo experto* en el que se modela el conocimiento experto asociado al dominio que se quiere enseñar. Este es el punto de referencia sobre el que se dirigen las estrategias instructivas y sobre el que se define la siguiente componente.
- Un *módulo del alumno* que intenta recoger lo que el alumno sabe del dominio. Este normalmente se realiza en base a las componentes del modelo de dominio. La estructura que almacena el estado que el sistema cree que tiene el alumno es el *modelo del alumno* en sí, mientras que al proceso de actualización del mismo se denomina *diagnóstico*.
- Un *módulo pedagógico* que se encarga de dirigir el proceso de instrucción. Éste actúa como instructor virtual que decide, en base al contenido del modelo del alumno y de los objetivos específicos de una estrategia instructiva, qué contenido presentar y en qué orden.

### 2.1.1. Modelado del alumno en STI

Como se mencionaba anteriormente, el modelo del alumno es una representación sobre las creencias que tiene el sistema acerca del conocimiento del alumno y, por tanto, una representación abstracta del mismo en el sistema (Holt et al., 1994).

Dado que la instrucción que un STI o un EIRP pueda proporcionar al alumno depende del contenido de su modelo, éste se presenta como una componente fundamental en la eficiencia instructiva del sistema. Existen diversas técnicas sobre las formas de construir el modelo, contenido del mismo y uso para la instrucción (Verdejo, 1994; Shute y Psotka, 1996). Una de las clasificaciones más comunes es la de Holt et al. (1994), que agrupa las técnicas básicas de modelado del alumno en las siguientes:

- *Modelo de superposición* (en inglés *Overlay Model*): Según esta técnica de modelado el conocimiento del alumno es considerado como un subconjunto del conocimiento de un experto en el dominio. Este enfoque supone que todas las diferencias entre el comportamiento del alumno y el del experto se explican como una falta de conocimiento del alumno. La instrucción según este modelo consiste en transmitir el conocimiento del experto que el alumno no sabe. El principal inconveniente de este modelo es que no considera que el alumno pueda tener conocimientos que no estén contemplados en el conocimiento del experto.

- *Modelo diferencial* (en inglés *Differential Model*): En este modelo el conocimiento del alumno es dividido en dos categorías, el conocimiento que el alumno debe saber, y el conocimiento que no se puede esperar que tenga. A diferencia del modelo de superposición, el modelo diferencial reconoce y trata de representar explícitamente tanto el conocimiento del alumno como las diferencias entre el alumno y el experto. Puede considerarse como un modelo de superposición, pero en lugar de superponer sobre el conocimiento del experto, se hace sobre un subconjunto de éste. No obstante, permiten más libertad que los anteriores al no ser tan estrictos en el modelado del alumno.

- *Modelo de perturbación* y *librería de fallos* (en inglés *Perturbation Model and Bug Library*): El modelo del alumno se forma mediante una combinación del modelo de superposición que almacena el conocimiento correcto del alumno, y una representación de los fallos cometidos (en inglés *bugs*) que son almacenados en una biblioteca de errores. Aunque la información adicional en un modelo de perturbación proporciona nuevas explicaciones del comportamiento del alumno, introduce también nuevos problemas: el esfuerzo necesario para construir y mantener el modelo del alumno es mucho mayor.

- *Modelo de conjuntos difusos* (en inglés *fuzzy set modeling*): Utilizan lógica difusa para trabajar con la incertidumbre acerca del conocimiento del alumno. La creencia del sistema acerca de que un alumno haya aprendido ciertas variables del conocimiento es modelada probabilidad difusa. Las variables del conocimiento representan habilidades y pueden ser agregadas en otras de mayor nivel. Las evidencias del alumno sirven para determinar el conocimiento del alumno en las variables más a bajo nivel, propagándose a los niveles superiores mediante mecanismos probabilísticos.

- Modelo basado en intervalos de confianza (en inglés *Bounded Models*): Se utilizan para modelar la incertidumbre del modelado mediante el establecimiento de intervalos de confianza sobre los límites inferior y superior del conocimiento del alumno.

- MBR: En el momento en que se realizó la clasificación, el MBR todavía estaba en desarrollo, pero ya entonces se vio claro su potencial. Esta técnica será explicada en detalle en la sección 2.3.

En una revisión reciente propuesta por Desmarais y Baker (2012) dos técnicas destacan en el panorama actual sobre el resto por su éxito y efectividad. Éstas son el MBR y los tutores cognitivos. Ambas son tratadas en secciones posteriores. En esta revisión se mencionan otro tipo de tutores como un tercer enfoque para modelar al usuario que actualmente está ganando interés: los tutores de secuenciación de contenido (en inglés *Content Sequencing Tutors*). La filosofía de éstos se centra en modelar al alumno mediante estados de conocimiento que son obtenidos a partir de una evaluación constante del alumno. En base a éstos se realiza la secuenciación, mostrando el contenido que el sistema estima irá cubriendo huecos en los estados de conocimiento.

Además de las mencionadas anteriormente, muchas otras técnicas han sido utilizadas en los STI, no sólo para el modelado, sino también para el diagnóstico del alumno. Algunas que se pueden destacar son:

- *Redes bayesianas* (Martin y VanLehn, 1995): Este es uno de los métodos clásicos para tratar con la incertidumbre del modelado del alumno. Las redes bayesianas realizan un razonamiento probabilístico sobre el estado del conocimiento del alumno basándose en su interacción con el sistema. Cada nodo de la red bayesiana mantiene una probabilidad asociada sobre la probabilidad de que el estudiante sepa esa pieza de conocimiento. A diferencia con los mencionados conjuntos difusos, las redes bayesianas se centran en el grado de creencia que cierta persona tiene acerca de si una variable está en un conjunto, mientras que la lógica difusa se centra en el grado de pertenencia de un elemento al conjunto.

- *Aprendizaje computacional* (en inglés *Machine Learning*): Este campo, subcategoría de la IA, trata con el diseño y algoritmos de técnicas que permitan a las máquinas aprender. Dada la enorme dimensión de este campo, los trabajos existentes utilizan diversas técnicas del mismo para múltiples aspectos. Por ejemplo, en (Langley et al., 1984) se utilizan árboles de búsqueda para generar modelos que mejor explican las acciones de un rendimiento particular. En un trabajo más reciente (Li et al., 2012) se utilizan técnicas específicas adquisición de conocimiento con el fin de obtener automáticamente el modelo de dominio a partir de ejemplos resueltos.

### 2.1.2. Áreas de investigación en los STI

El desarrollo actual de los STI abarca innumerables áreas de investigación y disciplinas, usando diferentes enfoques y metodologías. Puesto que nombrarlas todas sería una ardua tarea y no todas guardan relación con el trabajo que aquí se presenta, sólo mencionamos algunas de las más importantes o que de alguna forma están relacionadas con esta tesis.

Dado que el modelo que el sistema guarda sobre el alumno puede ser impreciso, una técnica para revisar este modelo y para favorecer el auto-aprendizaje, consiste en mostrar al alumno el modelo interno del sistema que refleja su conocimiento de una forma comprensible. El objetivo es proporcionarle una representación de su conocimiento que sirva de guía para el auto-aprendizaje, de forma que éste pueda decidir cómo actuar a fin de mejorar su conocimiento. A esta técnica de modelado se le conoce como *Modelo Abierto del alumno* (Kay, 1997; Mabbott y Bull, 2004; Bull, 2012). Al proceso de incorporar esta técnica a un sistema se le conoce como *abrir el modelo* del alumno. Las representaciones del conocimiento existentes son muy variadas, desde el diagrama de barras que indican el nivel del alumno en un tema concreto (Bull, 2004) hasta elementos mucho más complejos como los mapas conceptuales (Dimitrova, 2003; Bontcheva y Dimitrova, 2004).

El aprendizaje mediante la interacción social en el que distintos grupos de alumnos colaboran mediante un proceso determinado, es otro de los campos que los STI abarcan. Para hacer referencia a este tipo de aprendizaje en el dominio del aprendizaje mediante computador, se suele usar el acrónimo inglés CSCL (*Computer-Supported Collaborative Learning*). En este sentido, la interacción se realiza a través del ordenador mediante diversos elementos y procesos (Hoppe et al., 2007). El aprendizaje colaborativo es especialmente beneficioso cuando el objetivo de la instrucción es mejorar el pensamiento crítico y las habilidades relacionadas con la resolución de problemas Gokhale (1995). Inaba y Mizoguchi (2004) hacen una buena recopilación de los diversos beneficios educativos que esta forma de aprendizaje proporciona, como el desarrollo de habilidades

de auto-expresión, cognitivas, y meta-cognitivas.

Una de las áreas de investigación que están en auge en la actualidad se ubica en los **diálogos tutoriales**. Estos tienen surgen con el fin de imitar una de las características de los tutores humanos que se considera tienen mayor efecto en el aprendizaje: la conversación entre profesor y alumno. Mediante esta conversación el profesor fomenta la reflexión en el alumno a la vez que realiza un diagnóstico. Esta estrategia ha sido probada con efectividad en el aprendizaje en un número de sistemas que la implementan, algunos de los cuales son *AutoTutor* (Graesser et al., 2005), un sistema para transmitir conocimiento mediante conversación en dominios cuya naturaleza es predominantemente cualitativa (el conocimiento está contenido normalmente en expresiones verbales o del lenguaje natural que no son analíticas); *CIRCSIM-Tutor* (Evens y Michael, 2006), que trata sobre fisiología cardiovascular en relación con la regulación de la presión sanguínea; *Geometry explanation Tutor* (Aleven et al., 2003), una extensión de un tutor cognitivo en el dominio de la geometría que añade capacidad para que el alumno explique sus pasos; *Research Methods Tutor* (Arnott et al., 2008), centrado en las técnicas de investigación psicológicas en estudiantes universitarios; o Why2-Atlas (Vanlehn et al., 2002), en el dominio de la física matemática.

Otro de los terrenos importantes dentro de los STI es el de las **herramientas de autor** cuyo fin es el de facilitar la tarea de construcción de estos sistemas. Esta área se presenta casi como fundamental en cualquier paradigma importante de los STI dado que ayuda al desarrollo del paradigma y fomenta su uso. Su importancia queda patente también en el amplio número de herramientas que se pueden encontrar de este tipo (Murray, 1999, 2003). Algunas de estas herramientas serán mencionadas en las secciones 2.2.3 y 2.3.7.6.

Como se anticipaba al principio de este apartado, sólo hemos mencionado las áreas que guardan mayor relación con las técnicas utilizadas en la investigación que se presenta. No obstante, además de las mencionadas, existen muchas otras áreas de interés en los STI que pueden ser consultadas en (Nkambou et al., 2010; Cerri et al., 2012). Algunas que no se han mencionado y que caben destacar son: la *metacognición* (en inglés *metacognition*), que se encarga de usar técnicas para que el alumno reflexione sobre lo que sabe o lo que está haciendo, como la *auto-regulación* o la *auto-explicación* (en inglés *self-regulation* y *self-explanation*, respectivamente); la *minería de datos en la educación* (en inglés referido como *educational data-mining*), que estudia la aplicación de técnicas de minería de datos como parte de los STI; *ingeniería de ontologías* (*ontology engineering*), para la representación de elementos abstractos como el conocimiento; el *procesamiento del lenguaje natural* (*natural language processing*), como método para obtener información del alumno; o la *interacción hombre-máquina* (*human-computer interaction*), que explora las formas de comunicación entre el alumno y el ordenador.

## 2.2. Tutores cognitivos

Los tutores cognitivos (Anderson et al., 1995) representan uno de los enfoques de modelado del alumno más importantes hasta la fecha, en conjunción con el MBR que se explicará en la sección 2.3. Este tipo de tutores, desarrollados en su mayoría en la Carnegie Mellon University, tratan de modelar el dominio mediante un modelo cognitivo de competencias que el alumno debería saber. En este sentido, el modelo se centra en las habilidades cognitivas o conocimiento procedimental necesario para resolver los

problemas de un determinado dominio. Este tipo de tutores surgen inicialmente como un medio para estudiar y validar las teorías de Anderson en lugar de para desarrollar un nuevo tipo de sistemas tutores. A continuación se mencionan brevemente las teorías que motivan la aparición de estos sistemas y la forma en que se realiza el modelado del alumno.

### 2.2.1. Las teorías de Anderson

La primera teoría de Anderson es la conocida como ACT (del inglés *Adaptive Control of Thought*) (Anderson, 1983) que introduce la distinción de dos tipos de conocimiento involucrados en la resolución de problemas. Esta teoría establece que el proceso del pensamiento puede ser modelado usando conocimiento declarativo, que corresponde a cosas que somos conscientes que sabemos; y conocimiento procedimental, que se muestra en el comportamiento pero que no somos conscientes de él. El conocimiento procedimental o habilidades cognitivas vendrían representados por reglas de producción. La teoría ACT evolucionó a la ACT* (Anderson, 1983) que tenía una teoría más elaborada sobre cómo ocurre el proceso de adquisición de las habilidades cognitivas. Según esta última, la adquisición supone la formulación de reglas que relacionan objetivos y estados de una tarea con acciones y consecuencias, lo cual es modelado mediante reglas de producción del estilo *SI la meta es X ENTONCES realiza alguna acción y establece como subobjetivo Y.*

Posteriormente, la teoría ACT* dio lugar a la teoría ACT-R de Anderson (del inglés *Adaptive Control of Thought-Rational*) (Anderson, 1993; Anderson y Lebiere, 1998) que establece que el proceso de aprendizaje tiene lugar en dos pasos. En primer lugar el alumno debe adquirir el conocimiento declarativo apropiado para, después, convertirlo en procedimental. Por tanto, la teoría asume que las reglas de producción que representan el conocimiento procedimental sólo pueden ser aprendidas usando el declarativo. Esto quiere decir que las reglas de producción asociadas al conocimiento procedimental sólo pueden ser aprendidas haciendo y no simplemente mediante la recopilación de información. Este último motivo es la razón principal que motiva la implementación de este tipo de tutores para probar las teorías de Anderson. Aunque existen otras antecesoras a la ACT-R y varias versiones de esta última, las cuales todavía hoy en día siguen evolucionando, sólo se han mencionado las más importantes y conocidas de cara a dar una visión global. Para ampliar información sobre la teoría ACT-R se recomienda al lector visitar la página Web dedicada a ésta [1], en donde se detallan todas las publicaciones relacionadas, software disponible, y mucha más información.

### 2.2.2. Fundamentos

Los tutores cognitivos basados en las teorías anteriores siguen varias ideas clave. En primer lugar, el dominio se modela mediante reglas de producción que permitirían resolver problemas presentados al estudiante en la forma en que se espera que éste los resuelva. En cualquier estado de la fase de resolución de un problema, el modelo es capaz de generar, usando las reglas de producción, un conjunto de secuencias que representan diferentes formas de generar soluciones al problema. En segundo lugar, las reglas que modelan el conocimiento son de dos tipos: las que representan acciones válidas en el dominio y aquellas que representan acciones incorrectas. A estas últimas

[1] http://act-r.psy.cmu.edu/

se les denomina reglas incorrectas (del inglés *buggy rules*). En tercer lugar, el sistema es capaz de proporcionar dos tipos de instrucción: si el estudiante realiza una acción reconocida por una regla defectuosa, se le muestra un refuerzo explicando el error; por otro lado, si el estudiante solicita ayuda, se le muestra un mensaje en base a las reglas de producción correctas.

Usando los tres elementos anteriormente mencionados, el sistema comprueba cada paso de la resolución de un problema, intentando identificar una regla, bien sea correcta o defectuosa, que pudiera haber generado ese paso. A esta forma de instrucción se denomina *Traza del Modelo* (TM) (en inglés *Model Tracing*). Su nombre se refiere al proceso de intentar relacionar las acciones que el alumno realiza en el sistema para resolver un problema con alguna secuencia de reglas de producción activadas en el modelo cognitivo del dominio (Anderson et al., 1990; Koedinger et al., 1997).

El modelo del estudiante en los tutores cognitivos se realiza en base a las reglas del modelo cognitivo del dominio y pretende representar el aprendizaje el alumno sobre estas reglas. A esta forma de modelado se le denomina *Traza del Conocimiento* (TC) (en inglés *Knowledge Tracing*) (Anderson et al., 1995; Corbett y Anderson, 1995). La TC utiliza probabilidades de que un alumno haya aprendido cada regla del modelo cognitivo. A diferencia con la TM, cuyo objetivo se centra en ayudar al alumno a resolver un problema de forma satisfactoria, la TC pretende modelar el conocimiento cambiante del estudiante que serviría como base para proporcionar estrategias instructivas como la adaptación.

La TC normalmente utiliza redes bayesianas para modelar las probabilidades, existiendo todavía numerosas investigaciones sobre el mejor modo de determinar las probabilidades, entre las que se incluyen la maximización de la esperanza o la búsqueda mediante fuerza bruta, tal y como se recoge en (Desmarais y Baker, 2012). Otra alternativa al modelado bayesiano se encuentra en el llamado *análisis del factor de aprendizaje* (en inglés *Learning Factors Analysis*) (Cen et al., 2006; Pavlik et al., 2009a). Este enfoque utiliza un modelo logístico de regresión múltiple para modelar el aprendizaje combinado con técnicas de búsqueda combinatoria para calcular los parámetros del modelo. El principal problema es que el modelo no parece poderse usar para proporcionar adaptación. Para solucionar este problema el *análisis del factor de rendimiento* (en inglés *Performance Factors Analysis*) (Pavlik et al., 2009b), añade parámetros indicativos del rendimiento del alumno en un nuevo modelo.

Aunque la aplicación de la técnica de la TM es sencilla puesto que se reduce a una comparación de patrones, el proceso de construcción del modelo de dominio es similar al de la librería de fallos, ya que requiere de identificar los posibles fallos que el alumno podría cometer. Además, estos fallos se suelen identificar en base a un estudio del comportamiento de una población, tal y como se muestra en (Anderson et al., 1985), por lo que podrían no ser útiles para otra población. Sin embargo, el principal problema que tiene la técnica es que las acciones del alumno que no pueden ser relacionadas con ninguna regla del sistema no son interpretables por el sistema. Por este motivo es necesario un esfuerzo considerable en la creación del dominio para garantizar cubrir el máximo posible de acciones posibles.

### 2.2.3. Sistemas tutores y herramientas de autor

Los primeros tutores cognitivos desarrollados bajo este paradigma fueron dos sistemas implementados en el dominio de la programación declarativa y de la Geometría.

El primero, llamado *LISP tutor* (Anderson et al., 1984) permitía resolver pequeños programas en el lenguaje LISP. El segundo, llamado *Geometry tutor* (Anderson et al., 1981), se centra en las demostraciones usando las definiciones y postulados geométricos y su representación en forma de grafos. Ambos tenían como objeto el estudio de los principios que rigen el aprendizaje y la adquisición de habilidades en la resolución de problemas para probar las teorías de Anderson mencionadas anteriormente.

Otro de los tutores más prolíficos de este paradigma es el tutor PAT (*PUMP Algebra Tutor*) (Koedinger et al., 1997), cuyo nombre proviene del proyecto bajo el que se fraguó (*Pittsburgh Urban Mathematics Project*). El dominio educativo que trata es el de la resolución de problemas de Álgebra. Además de las características básicas de refuerzo ante errores o bajo demanda, el sistema posee una interfaz mediante la cual se proporcionan herramientas de representación, para hacer gráficas, y resolver ecuaciones. Su éxito se refleja en que, de las tres escuelas que se mencionan en la publicación inicial, su uso actual aumenta a más de dos mil en Estados Unidos (Koedinger y Aleven, 2007).

Muchos otros dominios han sido utilizados para la construcción de tutores cognitivos y muchas extensiones han sido realizadas sobre diversas áreas de investigación. Por nombrar sólo algunos de los tutores más destacados: el tutor Andes (VanLehn et al., 2005), en el dominio de la Física; SlideTutor (Crowley et al., 2005), para el diagnóstico de enfermedades de la piel mediante imágenes; Excel Tutor (Anderson et al., 1995), en la enseñanza de la conocida hoja de cálculo; Stoichiometry Tutor (McLaren et al., 2008) en la resolución de ecuaciones estoquiométricas, las cuales relacionan los elementos involucrados en una reacción química; y un largo etcétera.

También, con el objetivo de facilitar la tarea de la construcción de tutores cognitivos, principalmente para hacer extensible la creación de éstos a personas que no sepan de programación, se han desarrollado numerosas herramientas de autor. En esta área, la mayoría se basan de algoritmos de aprendizaje automático mediante programación por demostración para la adquisición de conocimiento y la creación de reglas de producción.

Una de las herramientas más conocidas para el propósito de la autoría es *Cognitive Tutor Authoring Tools* (CTAT) (Koedinger et al., 2004). Esta herramienta se basa del uso de programación por demostración para la construcción de dos tipos de tutores. Por un lado, se pueden construir los tradicionales tutores cognitivos, con el problema de que es necesario para ello conocimientos de programación en la creación del modelo cognitivo. Por otro lado, se pueden crear los denominados tutores de traza de ejemplos (del inglés *Example-tracing tutors*) (Aleven et al., 2006), los cuales no requieren conocimientos de programación y pueden utilizar tareas complejas, pero se centran en un problema concreto. La herramienta predecesora que da lugar a CTAT es Demonstr8 (Blessing, 2003), cuyo objetivo es la elaboración de tutores relacionados con la aritmética. Demonstr8 permite construir una interfaz para el nuevo tutor mediante un entorno gráfico que utiliza un algoritmo de programación por demostración para la generación automática de reglas de resolución. Sin embargo, con esta herramienta parece ser que no se construyó ningún sistema real.

Recientemente, se está trabajando en una meta más ambiciosa con la herramienta SimStudent (Li et al., 2012). Aunque se basa también de extracción de conocimiento mediante programación por demostración, el objetivo es, además de obtener las reglas de producción, la construcción del modelo de dominio de forma genérica. El algoritmo se encargaría de detectar a partir de los ejemplos las estructuras típicas y los elementos involucrados en los problemas para generar este conocimiento, ahorrando la tarea de especificar las estructuras involucradas a la hora de crear el sistema tutor.

Algunas otras herramientas que se pueden encontrar en la literatura sobre los tutores cognitivos son Tutor Development Kit (Anderson y Pelletier, 1991), el cual proporciona un entorno para la creación de reglas de producción y de la interfaz del sistema, pero su usabilidad queda en entredicho al compararse con las herramientas actuales en las que se obtiene conocimiento automáticamente; Cognitive Model SDK (Blessing et al., 2007) que, en lugar de usar programación por demostración, intenta usar otras representaciones más fáciles de entender para usuarios no familiarizados con tutores cognitivos y busca la reutilización de interfaces de tutores ya existentes; Assissment Builder (Razzaq et al., 2009), la cual es una simplificación de la herramienta CTAT en el sentido de que se elimina del proceso de construcción la escritura de las reglas de producción, pero que, al ser una simplificación, presenta limitaciones en comparación con CTAT, como la incapacidad para determinar la estrategia de resolución usada.

## 2.3. Modelado Basado en Restricciones

El *Modelado Basado en Restricciones* desarrollado por Ohlsson (1992, 1993, 1994) es una de las técnicas más populares y exitosas de modelado del alumno. Esta técnica ha sido impulsada por la Doctora Antonija Mitrovic, líder del grupo *Intelligent Computer Tutoring Group* de la Universidad de Nueva Zelanda, con el que hemos tenido el honor de colaborar en parte del trabajo realizado en esta tesis. La eficiencia de este paradigma ha sido probada a través de la implementación de un conjunto muy extenso de EIRP (Mitrovic et al., 2001, 2007; Mitrovic, 2012) que constituyen un obligado punto de referencia. Además, Mitrovic y su grupo han realizado una gran variedad de investigaciones sobre diferentes aspectos educativos de la técnica de modelado.

Esta técnica de modelado surgió para intentar solventar algunas de las limitaciones de la TM de los tutores cognitivos, comentada anteriormente. Concretamente, se buscaba reducir la complejidad requerida para desarrollar el conjunto de reglas de producción para generar una solución. Este desarrollo se complicaba si se quería proporcionar un refuerzo inteligente sobre los errores del alumno. Para ello, se hace necesaria la creación de reglas erróneas que permitan detectar los pasos erróneos. De no poseer estas reglas, cualquier paso erróneo no contemplado pasaría por correcto. Esto supone de un esfuerzo considerable en la construcción de un conjunto lo suficientemente completo para dejar sin contemplar el menor número de errores posibles.

De forma general, el MBR define un paradigma para la construcción de STI en dominios de tipo procedimental. El objetivo de esta técnica es mejorar el proceso de enseñanza haciendo que los estudiantes aprendan de sus propios errores mientras resuelven problemas asociados al dominio de enseñanza. La teoría sobre la que se asienta para alcanzar este objetivo es la conocida como *teoría de Ohlsson de aprendizaje a partir de los errores prácticos* (Ohlsson, 1996) (la traducción literal pierde el sentido completo de la original en inglés, conocida como *Ohlsson's theory of learning from performance errors*). Según esta teoría, un alumno comente errores, incluso si se le ha enseñado correctamente la forma correcta de hacer las cosas, porque el conocimiento declarativo que ha aprendido no ha sido introducido como parte del procedimental. Con la práctica de las tareas, si el alumno es advertido acerca de sus errores, éste va revisando y modificando su conocimiento de forma que se corrige el incorrecto y se va ampliando el procedimental. En resumen, el aprendizaje según la teoría de Olhsson es un proceso de dos pasos: en el primero, los errores se detectan mientras se está reali-

zando una actividad; en el segundo paso, se llevan a cabo las acciones tutoriales para corregir el error detectado.

El MBR se puede considerar un paradigma para el desarrollo de entornos educativos de resolución de problemas que facilita la construcción de los mismos. Esto es así puesto que establece un marco teórico centrado en modelar el dominio de enseñanza de una forma más simple que los demás paradigmas existentes. Este modelo del dominio simplifica también el modelo del estudiante asociado y las diferentes componentes de un sistema inteligente relacionadas con la instrucción. Como consecuencia de esta simplificación, la aplicación práctica del marco teórico se torna una de sus ventajas principales, dada su facilidad para llevarse a cabo en un sistema real, como se podrá ver a lo largo de esta sección.

### 2.3.1. Fundamentos

El propósito del MBR es intentar dar una solución al problema de la sobreespecificidad mencionado al principio del capítulo. Para ello, Ohlsson (1994) se guía por dos consideraciones a la hora de desarrollar este paradigma: la primera establece que en un sistema de aprendizaje, no es necesario reconocer todos los errores de un alumno, a menos que haya acciones pedagógicas para cada uno de ellos. En su lugar, para modelar al alumno se deberían usar clases equivalentes de situaciones (o estados del aprendiz) para las que hay una acción pedagógica apropiada. La segunda consideración establece, que dada la inmensidad del conocimiento falso o erróneo, se hace necesario emplear abstracciones para manejarlo. Estas dos consideraciones implican el uso de clases de equivalencia sobre las soluciones a construir en el sistema y que tendrían asociada la misma respuesta instructiva.

En la determinación sobre lo que deben ser las clases de equivalencia anteriores, Ohlsson (1994) menciona que, a diferencia del enfoque de Traza del Modelo, no todos los pasos que el alumno utilice para llegar una solución se deben tener en cuenta. Esta afirmación se justifica en que no todos los pasos de una solución reflejan de la misma forma el conocimiento del alumno. Algunos pasos serán más cercanos que otros a la conceptualización de las habilidades requeridas para resolver el problema. De esta forma, la información pedagógica útil no recae en la secuencia de acciones que el alumno realice, sino en el estado resultante el problema. Éste, se dice será *informativo diagnósticamente*, o que proporciona información para diagnosticar el conocimiento, si viola uno o más principios fundamentales del dominio.

De acuerdo con lo anterior, el conocimiento de un dominio se expresaría como un conjunto de principios o *restricciones* que ninguna solución a un problema dentro de ese dominio debería ser violar. Cada restricción define dos clases de equivalencia sobre las soluciones que se pueden construir: el conjunto de soluciones erróneas que violan la restricción y el conjunto de las que no. Estas clases de equivalencia suponen una abstracción sobre los pasos realizados para llegar a una solución que viole / satisfaga la restricción, lo que refuerza la afirmación de que no es importante la secuencia de acciones, sino la solución en la que éstas desembocan. A su vez, cada una de estas clases de equivalencia requiere una respuesta tutorial diferente, es decir, un tipo de remedio determinado para corregir la situación de error asociada.

### 2.3.2. Representación formal de las restricciones

Para aunar las ideas anteriormente detalladas, se utiliza una representación formal introducida por Ohlsson y Rees (1991), y denominada *restricción de estado* (en inglés *state constraint*). Este concepto es la clave del MBR como se podrá comprobar no sólo en el contenido de esta sección, sino también, en toda la investigación realizada durante esta tesis. A continuación se dan tres definiciones para expresar formalmente las restricciones de estado, que de ahora en adelante serán referidas simplemente como restricciones.

**Definición 2.1** (Restricción de estado)**.** *Una restricción de estado $i$ asociada a un dominio concreto de un sistema MBR, viene definida formalmente como un par ordenado de condiciones, el cual tiene la forma $\langle \varrho_i, \varsigma_i \rangle$. En este par, $\varrho_i$ representa la condición de relevancia y $\varsigma_i$ la condición de satisfacción de $i$. Ambas condiciones se definen a continuación.*

**Definición 2.2** (Condición de relevancia de una restricción)**.** *Dada una restricción $i$, la condición de relevancia asociada, $\varrho_i$, es una expresión lógica que se usa para determinar el tipo de problemas y los estados de las soluciones para los cuales la restricción se puede aplicar.*

**Definición 2.3** (Condición de satisfacción de una restricción)**.** *La condición de satisfacción $\varsigma_i$ de una restricción $i$ es una expresión lógica que modela la situación de estado erróneo vinculada con un principio del dominio que no debería ser violado.*

Ambas restricciones actúan conjuntamente para determinar cuándo un principio del dominio es violado. De esta forma, cuando la condición $\varrho_i$ de una restricción es verdadera para un determinado estado solución de un problema, se dice que la restricción es relevante para ese problema, puesto que ese estado podría contener un error. Es entonces cuando la condición de satisfacción se evalúa para determinar si la restricción es violada. De no satisfacerse la condición $\varsigma_i$, la restricción se ha violado, y por tanto, el principio del dominio asociado.

Para implementar las restricciones no se impone una manera específica de hacerlo, sino que se deja abierto. Las diversas formas de implementarlas o codificarlas son igualmente válidas, siempre y cuando se comporten de acuerdo al funcionamiento teórico descrito anteriormente que permita detectar los errores. Según Ohlsson (1994), la forma más obvia y conveniente de codificar una restricción, es mediante un par de patrones en el que cada patrón es una lista de proposiciones que pueden estar combinadas mediante conjunción o disyunción booleana. Algunos ejemplos de cómo realizar esta codificación se dan en (Ohlsson y Rees, 1991; Ohlsson, 1993; Mitrovic, 1998a; Martin y Mitrovic, 2000; Baghaei et al., 2006; Mitrovic y Ohlsson, 2006), entre otros.

Para la implementación realizada en la investigación asociada a esta tesis se han utilizado diferentes representaciones, las cuales se explican detalladamente en el capítulo 6. En las primeras, y siguiendo con los ejemplos encontrados, las restricciones se codifican como reglas de producción en las que el antecedente de la regla contiene las condiciones de relevancia y satisfacción. En las representaciones finales, se ha desarrollado una nueva forma mejorada usando reglas en las que el antecedente sólo contiene la condición de relevancia. La justificación de este cambio y el objetivo perseguido se explica en el apartado 6.5.1.

En las primeras investigaciones y estudios realizados sobre el MBR, se puntualiza que el planteamiento original de las restricciones permite representar solamente conocimiento sintáctico, es decir principios independientes del dominio (Mitrovic, 1998a,b,c). Esto tiene el inconveniente de que en algunos dominios, aunque las soluciones construidas no violan ningún principio del dominio, la interpretación semántica de la solución propuesta no permite resolver el problema planteado. Como ejemplo de esta situación, el dominio en el que surge esta apreciación es el del lenguaje SQL, en el que una consulta solución sintácticamente correcta puede ser errónea si no obtiene correctamente los datos requeridos al ejecutarse. En esta situación es necesario también alertar al alumno, por lo que es necesario un nuevo tipo de restricciones que permitan llevar a cabo esta tarea, las cuales Mitrovic denomina restricciones *semánticas*. Surge así la distinción entre restricciones *sintácticas*, que se asocian a los principios fundamentales del dominio, y las restricciones semánticas que se encargan de comprobar si la solución del alumno es la correcta.

Posteriormente a la distinción entre restricciones semánticas y sintácticas, se añade la terminología de *restricciones camino* (Mitrovic y Ohlsson, 2006), cuya referencia original corresponde al término inglés *path constraints*. Éstas comprueban que determinados pasos de la resolución se realicen en un orden concreto. Según los autores, este tipo de restricciones son necesarias en dominios procedimentales donde la ejecución de determinados pasos según una secuencia dada, es un principio del dominio, y por tanto debe modelarse en las restricciones (Mitrovic y Ohlsson, 2007). Su objetivo es similar al de las reglas de producción que se utilizan en los tutores cognitivos.

### 2.3.3. Componentes de un sistema MBR

En la literatura existente, cada sistema desarrollado suele tener particularidades en la arquitectura del mismo. Esto sucede sobre todo en los primeros sistemas implementados, como es natural, dada la novedad y el escaso grado de madurez de la técnica de modelado. Con la experiencia adquirida en el desarrollo de nuevos tutores (Mitrovic et al., 2004) y, más adelante, con el uso de herramientas de autor que ofrecen un marco de trabajo para la autoría de sistemas (ver apartado 2.3.7.6 para una revisión detallada de las herramientas existentes), se vislumbra una arquitectura en la que de una forma u otra, están presentes ciertas componentes que hacen que los sistemas MBR realicen su función educativa. Estas componentes se pueden ver en la figura 2.3, obtenida de (Mitrovic, 2006).

#### 2.3.3.1. Modelo del dominio

Empezando desde el núcleo de la arquitectura, la parte más importante es el *modelo de dominio*. Este modelo es implícito y está asociado a *lo que se enseña*. Explícitamente, está compuesto por los elementos relacionados con la materia o dominio que se va a enseñar en el sistema. El elemento central en este modelo es el conjunto de restricciones asociadas a cada principio del dominio y que son la base para detectar errores del alumno. Dentro, también se engloban los problemas que se pueden presentar al alumno, los cuales contendrán la información semántica necesaria para poder corregir la solución, tal como la solución ideal a cada problema. Aunque la solución ideal sirve de referencia para analizar la del alumno, ésta no siempre es necesaria y otras veces, cuando es necesaria, puede no ser útil porque haya muchas otras soluciones igualmente válidas.

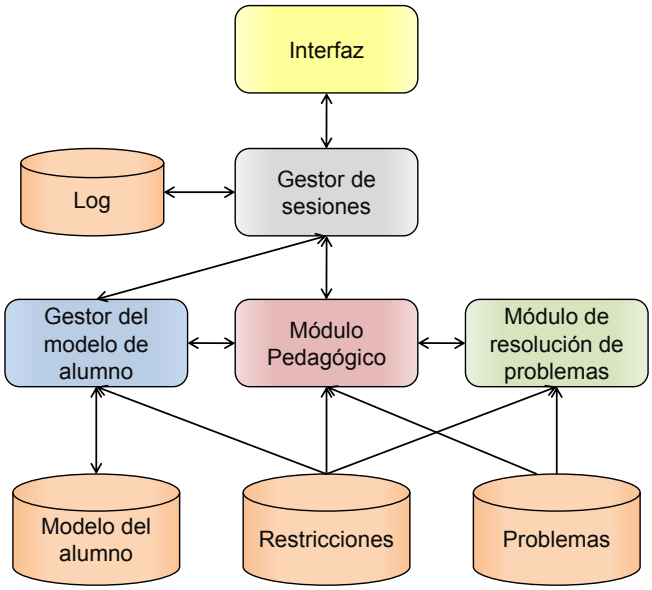



Figura 2.3: Arquitectura típica de sistemas MBR.

Concretamente, en dominios en los que no hay una solución concreta, o la solución puede expresarse de diversas formas, es necesario contar con algún mecanismo que resuelva el problema, lo cual es similar al proceso de resolución requerido por la técnica de traza del modelo.

El modelo de dominio debería ser codificado por un experto en la materia, pues requiere de identificar los principios involucrados y cuáles son los problemas que permiten aplicarlos. En relación con las restricciones, éstas deberían ser creadas directamente en el lenguaje usado, de acuerdo a la forma elegida para codificarlas, propia de cada sistema. En las primeras implementaciones realizadas, este proceso se hacía manualmente por una persona con los conocimientos de programación asociados. Con la aparición de las herramientas de autor, que se detallan en el apartado 2.3.7.6, este proceso se simplifica, capacitando a expertos en el dominio para realizarlo sin requerir conocimientos específicos de programación. Esto es posible por el uso de un mecanismo de aprendizaje automático (o *machine learning*) para la generación de las restricciones de forma automática a partir de una ontología del dominio y de un conjunto de problemas, definidos usando los elementos de la ontología (Suraweera et al., 2005). Este mecanismo puede llegar a producir restricciones incorrectas, inválidas o poco útiles, por lo que requiere de una etapa de supervisión de las mismas.

Respecto a los problemas del dominio, lo ideal sería que éstos contemplaran el conjunto completo de restricciones de forma que cada una estuviera contemplada por diversos problemas y formando diferentes combinaciones. En algunos sistemas en los que el número de restricciones es muy elevado, la definición de problemas puede ser un inconveniente. En este sentido, en (Martin y Mitrovic, 2002b) se contempla la generación automática de problemas a partir de un conjunto de restricciones. El mecanismo desarrollado utiliza la condición de relevancia de las restricciones a considerar para ir construyendo una solución ideal. Aunque el mecanismo permite construir una solución ideal, la generación del enunciado es una tarea muy compleja que puede dar lugar a problemas sin solución o a enunciados sin sentido. Por este motivo, la tarea, se debe

realizar de manera supervisada por un experto. El mecanismo no es parte de las herramientas de autor, lo que implica que no se integra con el de generación automática de las restricciones, de hecho, son en principio incompatibles, pues lo que se genera en uno es la fuente para generar el otro y viceversa.

Las componentes que se han mencionado sobre el modelo de dominio se suelen almacenar en formato LISP, mediante una serie de hechos declarativos para la definición de las restricciones y de los problemas. Aunque normalmente se utilizan ficheros LISP que posteriormente se cargan en la memoria de trabajo del motor de inferencia, también es posible el mantenimiento de la información en una base de datos, con registros dentro de tablas que representan cada elemento. Por ejemplo, una de las versiones de SQL-Tutor que funciona como aplicación independiente y que está disponible para descarga en el portal Database Place[2] (Mitrovic, 2006), almacena los problemas en ficheros LISP, mientras que las restricciones están incluidas dentro de la base de datos del sistema.

#### 2.3.3.2. Modelo del alumno

La segunda componente importante del MBR es el *modelo del alumno*, la cual modela *lo que éste sabe* en base al modelo de dominio. Puesto que el conocimiento del dominio está codificado en este paradigma mediante las restricciones, el conocimiento del alumno vendrá determinado por el resultado de aplicar cada restricción a las soluciones que éste ha construido. En algunos casos, y dependiendo de los problemas que se hayan intentado, puede que existan restricciones que no hayan sido relevantes en ninguno de los problemas resueltos. Si este es el caso, ese conjunto de restricciones no aporta nada sobre el conocimiento del estudiante. El resto de restricciones serán informativas diagnósticamente puesto que su resultado reflejará el conocimiento del alumno. Así pues, el modelo se reduce al conjunto de evidencias que se encuentra en el resultado de las restricciones relevantes, el cual estará constituido por evidencias negativas de conocimiento, en aquellas restricciones violadas, y/o evidencias positivas, si las restricciones han sido satisfechas. Además de los resultados, se almacena un modelo de superposición sobre las restricciones del dominio que contiene el porcentaje de veces que cada una ha sido satisfecha por el estudiante.

Como parte de este modelo del estudiante también se almacena la información asociada con la actividad realizada por éste que da lugar a la satisfacción o violación de reglas. Esta información consiste en una jerarquía de elementos que agrupan las actividades a distinto nivel. En primer lugar, y en un nivel más general, se sitúan las sesiones, las cuales comprenden toda actividad realizada en cada sesión de trabajo. La información que permite identificar el periodo de actividad de una sesión es almacenada en este modelo. En un nivel inferior, se sitúan los problemas que el estudiante ha intentado resolver en cada sesión. Dentro de cada problema habrá uno o varios intentos, cada uno con una solución que el alumno ha construido y que también es registrada. En el nivel más bajo de la jerarquía se sitúa la información atómica que representa realmente el modelo del estudiante en sí: las restricciones violadas y satisfechas en cada solución propuesta.

El modelo del estudiante es completado mediante una estimación del nivel de conocimiento. Normalmente se utiliza un valor numérico que se sitúa dentro del intervalo [1, 9] y que es proporcional al número de restricciones que el alumno sabe (Mitrovic y Martin, 2004). Para determinar el número de restricciones que el alumno sabe se

[2] http://www.aw-bc.com/databaseplace/

aplica un sencillo heurístico a cada restricción (Barrow et al., 2008). Dependiendo del momento de utilización del sistema, se calcula de una forma u otra. Al principio de la interacción con el sistema, se toman los 5 primeros intentos realizados por el alumno. Si el alumno ha satisfecho una restricción más del 70 % de las veces que ésta ha sido relevante, se considera que la sabe, en caso contrario está todavía por aprender. Al final de la interacción con el sistema, se toman los 5 últimos intentos y se realiza un cálculo similar, considerando que para aquellas restricciones que el alumno no sabía inicialmente, si se han satisfecho más del 70 % de las veces que han sido relevantes, entonces se han aprendido. Esta parte del modelo sería equivalente a la TM de los tutores cognitivos, con la salvedad de que en el MBR es mucho más simple y no está bien fundamentada.

En algunos estudios (Mayo y Mitrovic, 2000, 2001; Mitrovic et al., 2002), en lugar de usar el heurístico anterior, para determinar el conocimiento del estudiante se usa un modelo probabilístico. Concretamente, se emplean redes bayesianas para modelar la probabilidad de que el alumno sepa cada restricción en función del resultado en intentos anteriores y de la probabilidad de saber otras restricciones con las que se mantenga alguna relación. Este enfoque no tiene como resultado una evaluación numérica o un nivel del estudiante como tal, sino que el conocimiento está distribuido en las probabilidades de saber cada una de las restricciones del dominio. Esto es así puesto que el objetivo de esta forma de modelado no es proporcionar una evaluación sumativa. Lo que se pretende es mejorar la adaptación del sistema a las necesidades individuales para mejorar la instrucción del alumno, tal y como se verá en el apartado 2.3.5.

Al igual que con los elementos del modelo de dominio, el modelo del alumno se puede almacenar tanto en ficheros como en la base de datos del sistema. Normalmente, y motivado por el manejo más sencillo dentro del motor de inferencia utilizado en los sistemas existentes, se suele almacenar en ficheros escritos en lenguaje LISP, uno por estudiante. El fichero almacena la información que compone el modelo del alumno y que se ha mencionad anteriormente (resultado de cada restricción en aquellos problemas en los que ha sido relevante, así como la estimación del nivel del alumno y otras medidas internas).

#### 2.3.3.3. Componentes de gestión y control

En un nivel intermedio, y por encima del modelo de dominio y del estudiante, se sitúan tres módulos que realizan operaciones de gestión y control sobre los modelos anteriores:

- El primero de los módulos es el **gestor del modelo del alumno**, el cual se encarga de manipular el modelo asociado. Este módulo crea e inicializa el modelo para alumnos que entran por primera vez, consultar la información existente, y asociar el modelo del alumno a la sesión correspondiente. Dentro de esta componente también se sitúa el motor de inferencia que se ocupa de comprobar las restricciones violadas y satisfechas por las soluciones del alumno, las cuales se actualizan directamente en el modelo del alumno.

- La segunda componente de control, el **módulo pedagógico**, tiene la tarea de aplicar la estrategia pedagógica de acuerdo a las necesidades individuales contenidas en cada modelo del alumno. En el MBR, la pedagogía se encuentran en los mecanismos de adaptación proporcionados por dos mecanismos: el mecanis-

mo de información de los errores mediante refuerzo, conforme se explica en la sección 2.3.4; y el mecanismo de selección del siguiente problema a sugerir al alumno.

- La tercera componente que se distingue en la arquitectura, a este nivel, es el **módulo de resolución de problemas**. Este no se utiliza en todos los tutores MBR. Los objetivos de este módulo son dos, el primero se centra en proporcionar un mecanismo que sea capaz de generar una solución correcta a partir de la que el alumno ha proporcionado (Martin y Mitrovic, 2000). Con este mecanismo se pretende generar un refuerzo que informe de la solución correcta, partiendo de la del alumno. Para ello, se reemplazan las partes incorrectas con las componentes asociadas de la solución ideal. El segundo objetivo es generar problemas automáticos a partir de un conjunto de restricciones (Martin y Mitrovic, 2002b), como se comentaba anteriormente, en aquellos dominios donde el número de restricciones es muy elevado.

- Por encima de estos módulos de control se sitúa el **Gestor de sesiones**, el cual proporciona soporte a múltiples usuarios concurrentes, manteniendo el estado de cada uno durante su interacción con el sistema. El estado el alumno no es más que información sobre la sesión de trabajo, como el identificador del alumno o el problema que se está resolviendo. Además, esta componente sirve de punto de acceso y de comunicación entre la interfaz y los componentes más internos. Así pues, se encarga de organizar y orquestar los componentes con la información proveniente desde la interfaz. Por ejemplo, cuando un alumno envía la solución en la interfaz, ésta es pasada a través del gestor de sesiones al módulo pedagógico, el cual invocará al modelo del estudiante para comprobar los errores y generará un refuerzo que será devuelta de nuevo al gestor de sesiones para mostrarla en la interfaz. Esta componente realiza otra labor que es muy importante: registra toda la información asociada a la actividad del alumno y a las respuestas del sistema en ficheros de actividad o de log para cada alumno. La información almacenada incluye la solución que el alumno construye en cada intento; la lista de restricciones violadas o satisfechas; algunos razonamientos procedentes del módulo pedagógico, en cuanto a la secuenciación de acciones pedagógicas; y otras medidas internas del sistema.

- La componente más exterior y la que permite la interacción con el alumno es la **interfaz del sistema**. En ella se le proporciona al alumno los mecanismos adecuados para construir una solución a los problemas del dominio. Puesto que la evaluación de las restricciones requiere la utilización de la información asociada al estado solución construido por el alumno a un problema dado, la interfaz debe mantener una representación interna de la solución que se pasará al gestor de sesiones y éste gestionará con los módulos internos. Al igual que no se establece una forma concreta de implementar las restricciones, no se especifica una forma concreta de implementar esta representación, la cual dependerá de la primera implementación. Así pues, si se ha usado una lista de proposiciones elementales formando las diferentes condiciones de relevancia y satisfacción, lo más idóneo es usar una lista de proposiciones elementales representando cada componente de la solución. Esta representación debería ser actualizada conforme el alumno va construyendo la solución.

En determinados dominios procedimentales con tareas que requieren un conjunto de pasos determinados, es posible algunos de los pasos correspondan a estados de la solución altamente diagnósticas, pues representan un punto en los que se puede medir el conocimiento del alumno. En estos casos, el MBR puede aplicarse a cada uno de los pasos (Ohlsson y Mitrovic, 2006). De esta forma, es posible que la representación de la solución en estos pasos difiera de la utilizada para la solución final, siendo la interfaz la responsable de mantener la representación adecuada a cada paso de resolución.

### 2.3.4. Aprendizaje a partir de los errores

El proceso de comprobación de las restricciones es casi trivial y se reduce a un proceso de comparación de patrones en el que, primero, se usan los valores del estado solución para comprobar las restricciones que cumplen su condición de relevancia; y segundo, para estas restricciones se comprueba la condición de satisfacción que determinará la violación o satisfacción de la misma. Este proceso es computacionalmente rápido si se utiliza un algoritmo similar a las redes RETE (Forgy, 1982). Como resultado de esta comprobación, si una restricción es violada, uno o más errores han producido un estado incorrecto, lo cual permitirá alertar al alumno mediante refuerzo, que es el remedio en el que se basa la teoría de Ohlsson (1996). Este proceso sería similar a la TM en los tutores cognitivos, puesto que el objetivo es detectar situaciones erróneas y proporcionar un refuerzo para éstas.

El refuerzo consiste en un mensaje o una acción tutorial en la que se dé a conocer el error del estudiante. De acuerdo con la teoría psicológica base del MBR, el contenido del mensaje, para que éste sea efectivo, debe informar al alumno sobre dónde está el error en su solución, qué es lo que está provocando ese error, y cuál es el principio que se está violando. El hecho de que el contenido sea desarrollado según el criterio de un tutor humano experto en la materia no es una condición necesaria para alcanzar la máxima efectividad pedagógica. Esto queda reflejado en el estudio realizado por Zakharov et al. (2005), en el que se rediseña el refuerzo humano ajustándose a la guía anterior. Las conclusiones muestran que la eficiencia es mayor si se hace de acuerdo a los fundamentos psicológicos de la técnica de modelado.

Normalmente, los sistemas MBR presentan diversos niveles de refuerzo en los que cada uno es diferente en relación con la cantidad de información mostrada para guiar al alumno. Típicamente, se utilizan seis niveles que se van adaptando progresivamente (Mitrovic y Martin, 2000; Mitrovic et al., 2002; Mathews et al., 2008): 1) simple: el cual se presenta la primera vez que el alumno envía una solución y sólo se le avisa de si ésta es correcta o no; 2) aviso de error: tras varios envíos incorrectos, se le muestra la parte de la solución que contiene el error; 3) pista: si el error persiste, se le muestra un mensaje de refuerzo asociado a una de las restricciones violadas y se mantiene en este nivel, mostrando cada advertencia de cada restricción todavía violada; 4)solución parcial: en caso de que el alumno lo desee, puede solicitar este tipo de ayuda que mostrará la versión correcta de la primera restricción violada; 5) todos los errores: bajo demanda, al igual que en nivel anterior, se muestran las pistas asociadas a todas las restricciones violadas por la solución; 6) solución completa: como nivel más alto, y también bajo demanda del alumno, se le mostrará la solución ideal del profesor. Cambiando la forma de presentar esta ayuda daría lugar a nuevas estrategias de adaptación basadas en el refuerzo.

En el caso en el que varias restricciones sean violadas, como se ha mencionado an-

teriormente, sólo se muestra una restricción con el objetivo de hacer que el alumno se centre en un único principio cada vez. El sistema debe determinar sobre qué restricción mostrar el refuerzo. La forma más simple es establecer un orden determinado en las restricciones del dominio y usar este mismo para la presentación del refuerzo. En (Mitrovic, 1997, 1998b) se sugiere usar como factor discriminador para ordenar las restricciones la diferencia entre el número de veces que la restricción ha sido violada y el número de veces que ésta ha sido satisfecha. Sin embargo, Mitrovic (2012) afirma que esta opción es difícil de implementar. Otros investigadores han propuesto enfoques similares mediante la asignación de pesos sobre las restricciones y usar estos valores para la selección (Le et al., 2009).

En su formulación original, el refuerzo se proporciona ante restricciones violadas. Este hecho de presentar información ante la presencia de errores no es exclusivo de los sistemas MBR, sino que se puede encontrar también en otros STI (Anderson et al., 1990; Koedinger et al., 1997; VanLehn, 2006). En (Barrow et al., 2008) se compara este tipo de refuerzo, el cual denominan *refuerzo negativo*, con el *refuerzo positivo*, presentado ante acciones que dan lugar a soluciones correctas. Según diversos estudios (Ohlsson et al., 2007; Cade et al., 2008; Fossati, 2008; Di Eugenio et al., 2009), los tutores humanos también usan un refuerzo positivo sobre las respuestas correctas con el fin de reafirmar el conocimiento del estudiante, llegando a usarse con mayor frecuencia que la negativa. En el estudio realizado, el refuerzo no se presenta en cada satisfacción de las restricciones puesto que podría llegar a saturar al estudiante. Por el contrario, sólo se presenta en casos donde el refuerzo positivo puede ser beneficioso como la resolución correcta de una restricción considerada como difícil o si el sistema estima que el estudiante tiene dudas sobre una restricción y la resuelve correctamente. Siguiendo estas pautas, el estudio concluye que los estudiantes que reciben el refuerzo positivo aprenden el doble de rápido que los que sólo reciben negativa.

### 2.3.5. Adaptación

De las diferentes formas de secuenciación o andamiaje (en inglés *scaffolding*), la utilizada en los tutores MBR se basa en la adaptación que se realiza sobre el alumno. De acuerdo a Brusilovsky (1999), el andamiaje puede ser *activo* si existe un objetivo de aprendizaje o *pasivo*, si simplemente actúa en base a las acciones del estudiante proporcionando adaptación apropiada, de acuerdo al modelo del estudiante. Atendiendo a esta diferenciación, la adaptación en el MBR es pasiva y está presente mediante dos elementos principales: el refuerzo proporcionado cuando el alumno comete errores, y la selección del siguiente problema a realizar.

En cuanto al refuerzo, éste se proporciona en base al rendimiento inmediato del alumno. Como se explicó anteriormente, normalmente, se aplican seis niveles, de los cuales, los tres primeros se van adaptando dependiendo de si el alumno sigue cometiendo errores en la solución de un problema dado. En este sentido la adaptación no se realiza en base al modelo, sino que solo tiene en cuenta la secuencia de respuestas para un problema. Cuando el refuerzo se encuentra en el nivel de pista, en el cual sólo se muestra información sobre una de las restricciones violadas, el refuerzo a mostrar es seleccionado por el módulo pedagógico del sistema. Como norma general, esta decisión se realiza en base a un orden preestablecido, seleccionando la primera restricción violada según ese orden. En (Mitrovic, 1997, 1998b) se menciona que la adaptación se realiza ordenando las restricciones en base a las que presentan mayor problema (diferencia entre veces

relevante y veces satisfecha).

En (Martin y Mitrovic, 2005, 2006), se utiliza un mecanismo que intenta adaptar el refuerzo al modelo particular del estudiante proporcionando mensajes correspondientes a conceptos más generales. El mecanismo utiliza curvas de aprendizaje sobre los modelos de estudiante para determinar cuáles serían las mejores generalizaciones de refuerzo para, posteriormente, usar estas generalizaciones en base a las restricciones que se violan. Esto quiere decir que no se tiene en cuenta el modelo del alumno en tiempo real para proporcionar el refuerzo, sino que si un alumno viola ciertas restricciones, se le proporciona la correspondiente generalización establecida a priori. Los resultados muestran que esta generalización es efectiva en los dos primeros problemas, siendo necesario una más específica a continuación. Los autores proponen que, dados los resultados, para realizar el refuerzo en tiempo real, el sistema se debería basar en los modelos de todos los alumnos del sistema, siendo difícil la adaptación a un modelo particular del alumno por la reducida cantidad de información que éste pudiera tener.

Otra de las estrategias de adaptación en el refuerzo (Martin y Mitrovic, 2000) se basa en la solución que el estudiante ha construido, en lugar de utilizar el modelo del estudiante. La adaptación se realiza en el nivel más alto del refuerzo, correspondiente a mostrar la solución completa, si el alumno lo ha demandado. El mecanismo desarrollado utiliza la solución del estudiante, en conjunción con la solución ideal, para generar una solución correcta que será utilizada para proporcionar el refuerzo.

El segundo elemento sobre el que se proporciona adaptación consiste en la selección del siguiente problema a realizar. En los sistemas existentes, el siguiente problema no es impuesto al alumno, sino que existe también la posibilidad de que sea éste el que decida libremente el problema a realizar. Esta opción carece de adaptación, pero se intenta guiar al alumno presentado información como la lista ordenada por dificultad de los problemas o el modelo abierto que refleja las debilidades del alumno (Hartley y Mitrovic, 2002). La opción adaptativa se presenta cuando el alumno pide un problema al sistema, de forma general, o en base a un tipo de problemas que se desea resolver. Entonces, el sistema busca la restricción en la que el alumno presenta más problemas (Mitrovic, 2003a), o el concepto con mayor número de restricciones violadas (Suraweera y Mitrovic, 2004). El razonamiento para esto es que si el estudiante ha violado la misma restricción varias veces, sería adecuado considerarla (Mayo y Mitrovic, 2000). A partir de esto, se busca el problema que contiene esa restricción o conceptos, y que se adecua mejor al nivel del estudiante, lo que equivale a elegir un problema dentro de la zona de desarrollo próximo (Vigotsky, 1978).

Para determinar si un problema es adecuado para un alumno, éstos tienen una dificultad asociada. En los sistemas iniciales esta dificultad era fijada por el experto en el dominio con valores entre 1 y 9. Posteriormente, en (Mitrovic y Martin, 2004) y motivados por el trabajo de Brusilovsky (1992), en el que se distingue entre *complejidad estructural*, con un valor fijo y *dificultad del problema*, calculada dinámicamente en base al modelo del alumno, se introduce el cálculo de la complejidad de manera dinámica. Esta dificultad se calcula como la suma ponderada de la probabilidad de que el estudiante haya aprendido cada restricción relevante en el problema. La ponderación se determina a partir del número de predicados que contiene la restricción, y la probabilidad de saber una restricción se determina aplicando el método explicado en el apartado 2.3.3.2. Una vez realizado este cálculo para los problemas que el alumno no ha intentado todavía, se escoge aquel que difiere en una unidad el nivel del modelo del estudiante.

En otros estudios realizan diferentes estrategias de adaptación. Por ejemplo, en (Mitrovic y Martin, 2003) se utiliza el modelo abierto para justificar la elección del sistema en el siguiente problema a realizar si el estudiante tiene un nivel bajo y su elección de problema no coincide con la del sistema. Los resultados muestran que la asistencia a la hora de elegir el problema es beneficiosa para el aprendizaje. En otro estudio (Martin y Mitrovic, 2002b) se propone un mecanismo para generar problemas de manera adaptativa, de acuerdo a las restricciones más violadas por el estudiante, pero no se termina de aplicar porque los problemas requieren de un enunciado apropiado, cuya generación se dificulta al necesitar procesamiento de lenguaje natural.

Otras versiones más bien fundamentadas utilizan un enfoque probabilístico mediante redes bayesianas (Mayo y Mitrovic, 2000, 2001) en el que para seleccionar el problema se mira el número de refuerzos que es más probable se produzca. Este cálculo se realiza mirando cada restricción de un problema y viendo la probabilidad de violarse. Si el número de refuerzos es el más cercano a un valor óptimo, establecido en el nivel del estudiante más dos, entonces, se selecciona el problema. El valor óptimo está pensado para que a usuarios con poco nivel se les presenten problemas que generen menos refuerzos.

### 2.3.6. Otros estudios realizados

La investigación realizada en el campo del modelado del alumno mediante las restricciones no se centra sólo en la prueba de la efectividad instructiva del paradigma. Las estrategias tutoriales y los campos explorados son muchos. Los trabajos realizados en estas ramas de los STI se presentan a continuación.

No mucho después de los primeros estudios realizados en la Universidad de Canterbury, aparecieron los primeros estudios que utilizaban la técnica del **modelo abierto** (Mitrovic y Martin, 2002; Hartley y Mitrovic, 2002). En ellos, se demostraba que abrir el modelo del alumno era beneficioso, principalmente en estudiantes con bajo nivel. Sobre esta técnica también se ha investigado la eficiencia de modelos abiertos negociables, en los que si el alumno no está de acuerdo con la estimación del sistema, intenta demostrar su nivel en un tema concreto mediante alguna actividad adicional (Thomson y Mitrovic, 2010). Los resultados, aunque parten de una evaluación subjetiva, son prometedores, siendo requerido más trabajo en este área, al igual que pasa con el uso de representaciones complejas mediante mapas conceptuales, donde los resultados no son significativos pero prometedores (Duan et al., 2010). Otros estudios han analizado la eficiencia de las distintas representaciones del modelo, mediante técnicas de *seguimiento de los movimientos oculares* (en inglés *eye-tracking*) para determinar la utilidad de cada elemento contenido en la representación (Mathews et al., 2012). En general, para abrir el modelo del estudiante en sistemas MBR, especialmente en aquellos con un número elevado de restricciones, se hace necesario mostrar el conocimiento en base a agrupaciones de restricciones en conceptos más generales. El motivo es que, si se presenta el conocimiento en base a restricciones puede ser contraproducente para el alumno si el número de restricciones no es muy reducido y representa conceptos claros para éste.

La **colaboración** en la resolución de tareas también ha sido tenida en cuenta en el MBR. En (Baghaei, 2006; Baghaei y Mitrovic, 2006) se propone un modelo en el que se permite realizar una resolución colaborativa, tras responder de manera individual al problema. En este proceso de colaboración, el grupo debe proporcionar una solución al

problema de manera conjunta a la vez que justifica y discute con sus compañeros sobre las aportaciones realizadas. Además de las restricciones típicas para la comprobación de la corrección de la solución, se introduce el concepto de *meta-restricciones*, las cuales proporcionan refuerzo en base a las actividades realizadas por el estudiante y un modelo ideal de colaboración. El modelo consiste en comprobar que se realizan determinadas actividades relacionadas con varios aspectos de la colaboración (contribución al diálogo, contribución a la solución grupal, diferencias entre la solución individual y grupal, y monitorización del plan de resolución del problema). Posteriormente, en (Baghaei y Mitrovic, 2007; Baghaei et al., 2007), se muestra el estudio realizado sobre la efectividad de este modelo de colaboración. Según los resultados, el uso de las meta-restricciones para proporcionar refuerzo en las tareas relacionadas con la colaboración resultó significativamente efectiva, en comparación con la colaboración sin ayuda, y los alumnos mejoraron también significativamente con la colaboración. Otro estudio posterior sobre los efectos aislados de cada categoría de meta-restricciones (Holland et al., 2011), muestra que éstas son efectivas al favorecer el aprendizaje.

El **aspecto afectivo** de los STI también ha sido tenido en cuenta en esta técnica de modelado. El primer trabajo realizado en este campo es (Zakharov et al., 2007). Aquí, se añade un avatar al sistema que hace las veces de un mentor, el cual reacciona con diferentes estados emocionales de acuerdo a las acciones del alumno. El modelo utiliza una serie de reglas de comportamiento que dictan las acciones a realizar por el avatar, las cuales se presentan en forma de refuerzo verbal y expresiones faciales. El funcionamiento del sistema, en relación con la afectividad, consiste simplemente en reaccionar a soluciones correctas, incorrectas, o estados de neutralidad, mostrando caras sonrientes, tristes intentando empatizar con el estudiante.

Posteriormente, en (Zakharov et al., 2008), este modelo es extendido mediante el reconocimiento de los rasgos faciales del estudiante con el fin de mejorar el mecanismo para detectar su estado emocional y poder actuar en consecuencia. Los resultados del estudio subjetivo realizado en este último trabajo, mediante encuestas de opinión, muestran la preferencia de los usuarios del avatar afectivo frente a un avatar sin esta característica. Aunque no hay mucho trabajo objetivo que muestre la efectividad de los agentes afectivos desarrollados, se sigue trabajando en esta línea actualmente.

Otra de las ramas exploradas en el grupo neozelandés ha sido el uso de **diálogos tutoriales** para la mejora del proceso instructivo. Desde el trabajo realizado por Weerasinghe y Mitrovic (2002), en el que se planteaba una metodología teórica para tratar los errores más frecuentes del alumno mediante diálogos tutoriales; pasando por los estudios de Mitrovic (2005a); Weerasinghe y Mitrovic (2006), en los que se mostraba que tales diálogos permitían un proceso de aprendizaje más rápido; varios trabajos han seguido esta línea de investigación hasta la actualidad. Inicialmente, los diálogos se basaban en los errores frecuentes, cada uno de los cuales se asociaba con una componente de una jerarquía en la que se agrupaba la violación de restricciones como representantes del error. En trabajos posteriores (Weerasinghe et al., 2008, 2009), se extendió el modelo para contemplar selección adaptativa de los diálogos. Para realizar esto, se utiliza el modelo de usuario y se presenta el diálogo asociado a los errores en los que el alumno tiene mayor dificultad. Además, la adaptación tiene en cuenta la longitud de los diálogos y su contenido. En estudios más recientes realizados por Weerasinghe et al. (2010, 2011), se mostró que el uso de esta técnica mejoraba el aprendizaje de los alumnos que la recibían.

### 2.3.7. Herramientas existentes

El *Modelado Basado en Restricciones* ha sido aplicado como técnica de modelado del alumno y del estudiante en diferentes entornos de aprendizaje o sistemas inteligentes. Pero además, su uso no sólo se queda ahí, sino que ésta se ha convertido en un paradigma que dicta una serie de pautas para la construcción de sistemas educativos. Esto ha desembocado en la creación tanto de diversos tutores inteligentes como de herramientas de autoría para la creación de nuevos sistemas tutores. En este apartado se muestran algunas de las más importantes.

#### 2.3.7.1. Sistemas tutores

Aunque el nacimiento de la formulación teórica del MBR se sitúa a principios de los 90, no es hasta finales de la misma década cuando surgen las primeras implementaciones. Desde entonces, ha sido demostrada su eficiencia y versatilidad al aplicarse con éxito en diferentes dominios de enseñanza. La mayoría de los estudios realizados se centran en la temática de las bases de datos y han sido desarrollados por la Doctora Mitrovic y su grupo de investigación. A continuación se muestran las herramientas más importantes desarrolladas por el grupo neozelandés y algunas que han sido realizadas por otros investigadores externos al mencionado grupo.

Para cada herramienta existente se introducirá el dominio de aplicación sobre el que están basadas, mencionando el tipo de problemas o actividades educativas que se pueden realizar, y ubicándolas de acuerdo a la clasificación que se explicó al principio de este capítulo, realizada por Mitrovic y Weerasinghe (2009).

#### 2.3.7.2. SQL Tutor

Uno de los sistemas más prolíficos que existen en el paradigma del MBR es SQL-Tutor (Mitrovic, 1998b, 2003a; Mitrovic y Ohlsson, 1999). Sobre él se han llevado a cabo multitud de estudios de diferentes componentes del sistema con fines pedagógicos diversos (Mitrovic, 2012). Uno de los indicadores de su éxito se puede encontrar en el número y fuente de usuarios, que no sólo se limita a los estudiantes que desde 1998 utilizan el sistema en la Universidad de Canterbury, Nueva Zelanda, en donde éste ha sido desarrollado, sino que, está disponible a nivel mundial mediante el portal Web Database Place[3]. La editorial Addison Wesley ha sido la creadora de dicho portal, que incorpora, además de SQL-Tutor, otros dos sistemas relacionados con el aprendizaje de las bases de datos (Mitrovic, 2006), los cuales se mencionan posteriormente.

El tutor es un EIRP enfocado en el lenguaje SQL, que es el predominante de consultas sobre bases de datos relacionales. Los problemas que se le plantean al alumno consisten en la construcción de una consulta en lenguaje SQL que, dado un esquema de una base de datos, permita obtener cierta información de ésta. Aunque es un lenguaje simple y bien estructurado, surgen algunos problemas con ciertos conceptos avanzados del mismo y, sobre todo, con la sobrecarga cognitiva (Sweller et al., 1998) asociada a este tipo de problemas. Esta última es el resultado de tener que mantener mentalmente muchos detalles asociados con el problema que se está resolviendo, tales como el esquema de la base de datos, el objetivo de la consulta, las restricciones sobre los atributos de las tablas, etc. Esto hace que, aunque el dominio sea bien definido, las tareas

[3] http://www.aw-bc.com/databaseplace/

pertenezcan a la categoría de débilmente definidas, de acuerdo con la clasificación de Mitrovic y Weerasinghe (2009).

En su versión inicial, el sistema funcionaba como una aplicación de escritorio, implementada mediante Allegro Common Lisp (Franz Inc, 1998), que desempeñaba sus funciones, de manera independiente, en máquinas Windows o estaciones Sun. Posteriormente, se desarrolló una versión Web que, aunque seguía usando Common Lisp como motor de inferencia, cambió la implementación a la versión CL-HTTP (Mallery, 1994). Esta nueva versión se llamó SQLT-Web (Mitrovic, 2003a) y utilizaba una arquitectura cliente-servidor. El cliente era la interfaz del sistema y el servidor contenía el módulo pedagógico, el modelo de dominio, todo el mecanismo de razonamiento, y la lógica de negocio. La evolución de la interfaz y la extensión de la base de conocimiento asociada a las restricciones del dominio, es el sistema que hoy conocemos como SQL-Tutor.

Como podrá observarse en la sección 7.5, una parte de la investigación desarrollada ha tenido la oportunidad de utilizar datos provenientes de este sistema en la experimentación realizada.

SQL-Tutor utiliza como elemento pedagógico adicional al MBR el *descubrimiento guiado*, uno de los estilos de enseñanza usuales de los STI. El descubrimiento guiado se basa en la idea de que los estudiantes puedan explorar y descubrir cosas por sí mismos, en lugar de ser obligados a hacer algo. Esta metodología favorece un mejor aprendizaje y la retención del conocimiento adquirido durante más tiempo (Anderson, 1993). Por otro lado, la exploración sin restricciones, no es recomendable, principalmente en principiantes, ya que los estudiantes pueden perder demasiado tiempo sin rumbo. La solución se encuentra en la orientación, en forma de pistas bajo demanda o de forma automática, tal y como se mencionaba en el apartado 2.3.4.

La interfaz del sistema, tal y como puede observarse en la figura 2.4, está diseñada para tratar los problemas de la sobrecarga cognitiva inherente, mencionados anteriormente. Este objetivo es alcanzado principalmente por la parte etiquetada con (D), la cual muestra información detallada del esquema de la base de datos, tanto de las tablas, como de sus atributos. De esta forma, se evita al alumno el problema de la sobrecarga cognitiva asociada a mantener dicha información en la memoria de trabajo mental (Sweller et al., 1998). En la figura, el elemento superior, etiquetado con (A) muestra el enunciado del problema a resolver. También, bajo la etiqueta (B), el sistema presenta las partes de la solución que el alumno debe construir estructuradas en las diferentes componentes de una consulta SQL. Esta forma de presentar huecos para rellenar por el estudiante simplifica el problema en diferentes subobjetivos, cada una asociada a una componente de la solución completa. La última parte de la interfaz, etiquetada con (D), presenta información sobre las los errores, o en caso de que proceda, los aciertos que la solución haya producido sobre las restricciones del dominio.

El modelo de dominio del sistema está compuesto por un conjunto de más de 700 restricciones que abarcan casi todos los principios involucrados en el lenguaje SQL. Aunque el conjunto de restricciones no es completo, en relación con todas las sentencias que se pueden construir en el lenguaje, esto no supone un problema, puesto que permiten diagnosticar las soluciones posibles en los problemas que forman parte del dominio, cuyo cardinal supera los 300 (Mitrovic, 2012). En cuanto al modelo del alumno, la versión más reciente contempla el modelo básico de almacenar el histórico de las soluciones y una estimación heurística del estudiante del mismo modo que se explicó en la sección 2.3.3.2.

La ayuda que reciben los estudiantes al comprobar la corrección de una solución

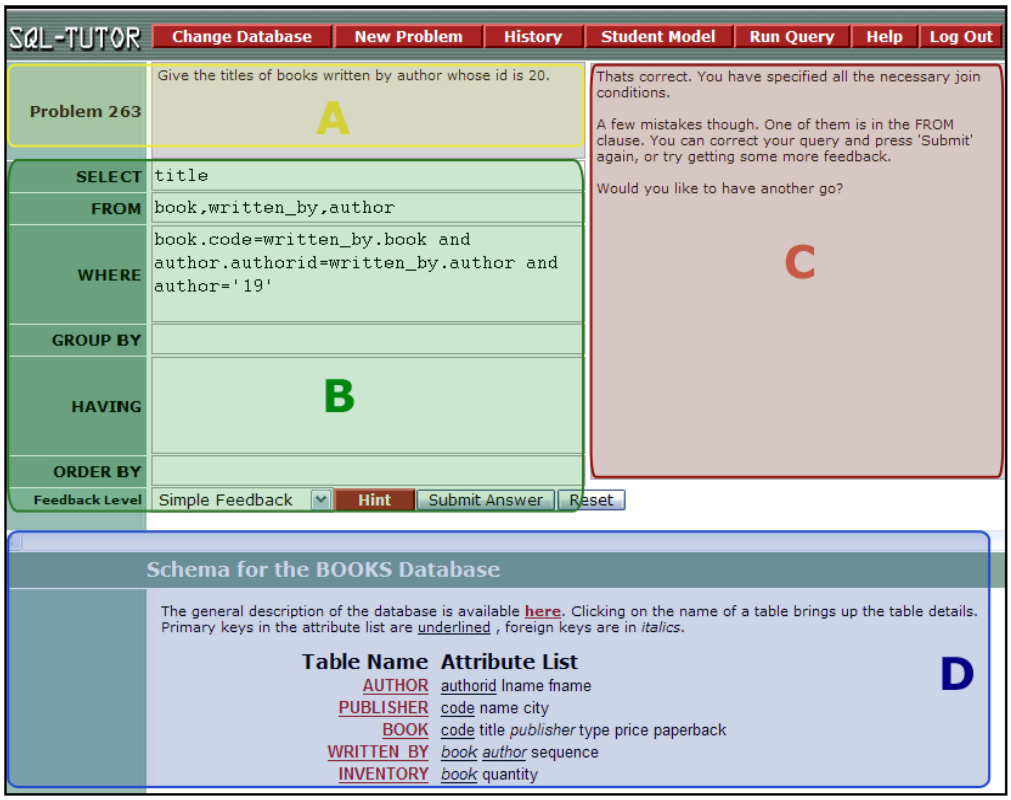



Figura 2.4: Interfaz del sistema SQL-Tutor.

tiene seis niveles que abarcan desde un aviso simple, indicando si hay algún error, hasta la solución completa, tal y como se describe en detalle en el apartado 2.3.4. El estudiante puede solicitar bajo demanda los niveles de ayuda más altos en caso de que no sepa como continuar. Además, el sistema dispone de refuerzo sobre restricciones que el alumno satisface pero que todavía no domina, con el objetivo de favorecer el aprendizaje de éstas.

Sobre este sistema se han realizado multitud de investigaciones como el estudio de la efectividad de la refuerzo positiva en comparación con la negativa de Barrow et al. (2008); estudio de la efectividad de proporcionar refuerzo para agrupación de restricciones, en comparación con restricciones individuales de Martin y Mitrovic (2005, 2006); comparación de la adaptación mediante el cálculo de la dificultad de los problemas de forma dinámica, frente a una dificultad estática de Mitrovic y Martin (2004); generación de plantillas de problemas realizada por Mathews y Mitrovic (2007); apertura del modelo del alumno (Mitrovic y Martin, 2002); y muchas otras que se mencionan a lo largo de esta sección.

#### 2.3.7.3. EER-Tutor

Otra de las tres herramientas más importantes desarrolladas por el grupo neozelandés en el dominio de las bases de datos es el sistema EER-Tutor (del inglés *Enhanced-Entity Relationship Tutor*) (Zakharov et al., 2005). Esta herramienta se centra en el dominio del diseño conceptual de bases de datos mediante el uso del modelo Entidad

Relación (Chen, 1976). Los problemas que se plantean al estudiante en este sistema consisten en un enunciado a través del cual se especifica una serie de entidades, cada una con unos atributos concretos y relaciones, a su vez, con otras entidades. El proceso de resolución consiste en la construcción de un esquema que represente las entidades, sus propiedades, y las relaciones existentes entre ellas. Aunque el dominio es bien definido, pues el modelo es relativamente simple, las tareas asociadas son débilmente definidas, pues para llegar hasta una solución se pueden seguir infinidad de caminos.

El sistema sigue una arquitectura cliente servidor similar a la explicada en el apartado 2.3.3. Por un lado, el cliente está formado por un entorno gráfico, mediante Applets Java, que proporciona al alumno las herramientas necesarias para construir la solución al problema. Concretamente, se pueden añadir entidades al modelo solución, especificar sus atributos, y relacionarlas con otras entidades. Por otro lado, el servidor realiza la comprobación de la solución utilizando el motor de inferencia Allegro Common Lisp, actualiza el modelo del alumno, y pasa la refuerzo a la interfaz para presentarla al alumno.

Originalmente, al sistema se le llamó *KERMIT* (*Knowledge-based Entity Relationship Modeling Intelligent Tutor*) (Suraweera y Mitrovic, 2001, 2002, 2004). KERMIT era una aplicación de escritorio desarrollada en el lenguaje Visual Basic. Posteriormente, se implementó una versión Web de la aplicación con la herramienta de autor WETAS (ver apartado 2.3.7.6) que se denominó *ER-Tutor* (*Entity Relationship Tutor*) (Suraweera y Mitrovic, 2002, 2004). Finalmente, se extendió el modelo de dominio de ER-Tutor para contemplar nuevos elementos, ampliar el conjunto restricciones y mejorar la funcionalidad, dando lugar al actual EER-Tutor.

Algunas de las versiones implementadas han acuñado diferentes nombres, como e-KERMIT (Hartley y Mitrovic, 2002), la cual es una versión que abre el modelo del alumno de KERMIT; y KERMIT-SE (Weerasinghe y Mitrovic, 2002), que incorpora auto-explicación a la versión básica (de ahí el prefijo SE, del inglés *Self-Explanation*). Otras versiones mantienen el nombre de su original pero tratan diversas ramas pedagógicas como la evaluación de modelos abiertos del alumno de EER-Tutor (Duan et al., 2010; Mathews et al., 2012); implementación de un modelo abierto del alumno que puede ser negociado en EER-Tutor (Thomson y Mitrovic, 2010); efectividad del aspecto afectivo de EER-Tutor (Zakharov et al., 2007, 2008); implementación y evaluación de diálogos tutoriales en EER-Tutor (Weerasinghe et al., 2009, 2010, 2011); o efectividad de la ayuda bajo demanda (Mathews et al., 2008).

#### 2.3.7.4. NORMIT

La tercera de las herramientas que forman el trío de aplicaciones sobre bases de datos lo completa NORMIT (*NORMalization Intelligent Tutor*) (Mitrovic, 2002). El dominio educativo que trata es el de la normalización de bases de datos relacionales (Codd, 1974). Los problemas asociados a este dominio consisten en organizar los campos y las tablas de una base de datos con el fin de minimizar la redundancia y la dependencia. Para realizar este proceso, se va refinando la estructura de la base de datos de acuerdo a varias formas normales, las cuales establecen requisitos que las tablas deben cumplir y que se asocian a distintos niveles sobre los cuales se puede evitar la redundancia y la dependencia. El dominio educativo asociado a NORMIT es bien definido, al igual que las tareas que es necesario realizar, pues existe un algoritmo bien definido para hacer que la base de datos cumpla cada una de las formas normales.

La particularidad de este sistema es que es el primero que se centra sobre dominios procedimentales en los que la secuencia de pasos a realizar para alcanzar la solución está bien definida y es importante realizarla en el orden apropiado. En la versión inicial de NORMIT la secuencia de pasos es fija y se asocia a una tarea que el alumno debe realizar, no dejando libertad sobre el orden. Posteriormente, en (Mitrovic y Ohlsson, 2006) se menciona el uso de las anteriormente citadas *restricciones camino*, las cuales, permiten comprobar la ejecución correcta de los pasos en un entorno en el que el alumno tiene libertad para realizar los pasos en el orden deseado.

NORMIT, al igual que EER-Tutor, sigue una arquitectura Web idéntica a la explicada en la sección 2.3.3, implementada en Allegro Common LISP que funciona sobre el servidor Web AllegroServe[4]. La interfaz consiste simplemente en páginas HTML que mandan la información al servidor, el cual se encarga de invocar al módulo de resolución de problemas para comprobar la corrección de la solución del alumno. En este proceso, además de aplicar las restricciones sobre la solución del alumno, el módulo de resolución de problemas generará la solución ideal que será comparada con la del alumno. El conjunto de restricciones del dominio usado para detectar los errores, en su versión inicial, consta de 53 restricciones.

Algunas de las versiones del sistema se corresponden a investigaciones realizadas sobre áreas de los STI como el modelo abierto del usuario (Mitrovic, 2003b); la autoexplicación correspondiente a la versión NORMIT-SE (Mitrovic, 2005a,b); los diálogos tutoriales con el alumno (Weerasinghe et al., 2011); y la generación automática de restricciones a partir de ontologías (Suraweera et al., 2010), mecanismo que se utiliza posteriormente en la implementación de las herramientas autor (ver apartado 2.3.7.6).

#### 2.3.7.5. Otros sistemas tutores

Además de los dominios mencionados anteriormente, el grupo neozelandés también ha contemplado otros ámbitos para el desarrollo de sus sistemas. Algunos de los más destacados son los siguientes:

- Collect-UML (Baghaei et al., 2005) es un sistema que se enmarca en el dominio del diseño orientado a objetos mediante el lenguaje de modelado unificado (UML). Los problemas que se le presentan al estudiante requieren que éste construya un diagrama UML que represente las entidades, atributos y relaciones especificados en un enunciado. Este sistema se enmarca dentro de los dominios bien definidos pero con tareas débilmente definidas. La característica más destacada de Collect-UML, es que es el primero de los sistemas MBR, sobre el que se incorpora la colaboración entre estudiantes como parte del proceso de aprendizaje (Baghaei et al., 2007; Holland et al., 2011).

- CAPIT (Mayo et al., 2000) trata el dominio de la gramática inglesa. Concretamente, se ocupa del uso de las mayúsculas y de los signos de puntuación, proporcionando una o varias frases para que el alumno complete huecos con la adecuada capitalización de las letras o signos de puntuación. Este tipo de dominio es bien definido y, a su vez, las tareas son bien definidas. Como característica principal del sistema, para modelar al alumno y realizar el proceso adaptativo, se utilizan redes bayesianas (Mayo y Mitrovic, 2001). Como se vio anteriormente, este modelo se puede aplicar en sistemas con un modelo de dominio muy reducido.

[4] http://allegroserve.sourceforge.net/

- ERM-Tutor (Milik et al., 2006a,b) pertenece al tipo de dominios en el que la tarea de resolución tiene unos pasos asociados bien definidos. En concreto, el dominio se sitúa en el paso de diagramas que siguen el modelo Entidad Relación a esquemas relacionales de bases de datos. Este proceso está bien definido y sigue un algoritmo con unos pasos concretos, por lo que, tanto el dominio, como las tareas de resolución, son bien definidas. Aunque este sistema también corresponde a temas de bases de datos, no ha madurado tanto como los tres sistemas explicados anteriormente, probablemente porque los estudios realizados no presentan datos significativamente importantes. Sobre este sistema se han realizados algunos estudios en el ámbito de los diálogos tutoriales (Weerasinghe et al., 2008, 2009).

- J-LATTE (Holland et al., 2009), cuyo nombre proviene de las siglas del inglés *Java Language Acquisition Tile Tutoring Environment*, se centra en el dominio de la programación Orientada a Objetos mediante el lenguaje Java. Los problemas que los estudiantes pueden resolver consisten en implementar un método java, el cual puede ser de tres tipos diferentes: escritura de un resultado, devolución de un resultado lógico de verdad o falsedad, o devolución de un resultado mediante el uso de sentencias iterativas. Aunque el dominio del sistema es bien definido, pues las componentes que se pueden utilizar para construir los métodos pertenecen a un conjunto reducido, la tarea de implementación es débilmente definida, ya que la forma de alcanzar una solución implica infinidad de caminos o combinaciones de sentencias.

Aunque en su mayoría, los sistemas existentes han sido desarrollados en el grupo neozelandés, también existen otros trabajos de otros autores que son recogidos en (Mitrovic, 2012) y que abarcan diversos dominios educativos como: la electrónica (Billingsley et al., 2004); diagramas UML (Le, 2006); Física (Mills y Dalgarno, 2007); algoritmos de enseñanza (Petry y Rosatelli, 2006); análisis del habla en entornos colaborativos (Rosatelli y Self, 2004); diseño arquitectónico (Oh et al., 2009); aprendizaje de la lengua (Menzel, 2006); y matemáticas discreta (Billingsley y Robinson, 2005).

#### 2.3.7.6. Herramientas de Autor

Dentro de las herramientas desarrolladas en la Universidad de Canterbury, también se contemplan las de autor. Con el objetivo de facilitar la construcción de nuevos tutores, bajo el paradigma del MBR, se han realizado varios estudios. Cada estudio realizado es la evolución o mejora de uno anterior. Este hilo argumental que tiene como fin el de cumplir el objetivo anteriormente mencionado, ha tenido como resultado diversas herramientas para la autoría o elicitación de sistemas tutores. En este apartado se mencionan estas herramientas y el trascurso de la investigación, la cual comienza con WETAS, y evoluciona a través de CAS, ASPIRE, y, finalmente, VIPER.

**WETAS** (*Web-Enabled Tutor Authoring Shell*) (Martin y Mitrovic, 2002c,a) es un entorno diseñado para proporcionar todas las funciones independientes del dominio a la hora de construir de sistemas tutores con una interfaz sencilla basada únicamente en elementos HTML. Estas funciones independientes del dominio se corresponden con aquellos mecanismos presentes en cualquier sistema tutor MBR. Concretamente, proporciona las funcionalidades de selección de problemas, evaluación de la solución

generada por el alumno, modelado del alumno, proporción de refuerzo, y generación automática de la interfaz.

De acuerdo con la poca información existente para esta herramienta, el autor necesita sólo especificar las componentes específicas del dominio correspondiente al sistema que se quiere construir. Estas componentes incluyen la estructura del dominio, en caso de que tenga alguna estructura de pasos concreta que se deba seguir para la resolución; el modelo del dominio, en forma de restricciones; el conjunto de problemas del dominio; cualquier información necesaria sobre la secuenciación del material educativo; y software encargado de procesar la entrada, si es requerido algún procesamiento previo.

La principal limitación de WETAS es su falta de apoyo en el proceso de autoría de un sistema tutor (Suraweera et al., 2004a). Esto es así porque WETAS no proporciona ninguna asistencia para desarrollar las diversas componentes del modelo del dominio. Por el contrario, las restricciones y los problemas, deben ser codificados en algún editor externo antes de utilizarse en sistema, lo cual puede ser muy engorroso. Básicamente, WETAS es un servidor Web, basado en Allegro Common Lisp sobre AllegroServe, que da cabida a las componentes específicas del modelo de dominio, las cuales son usadas por los elementos comunes del MBR para permitir la ejecución de diversos sistemas tutores. Con la herramienta WETAS se han construido dos sistemas tutores (Martin y Mitrovic, 2002a): LBITS (*Language Builder ITS*), un sistema tutor en el dominio de la lengua inglesa mediante puzles, y una versión de SQL-Tutor.

Con el objetivo de asistir en el proceso de elicitación del modelo del dominio y de reducir la complejidad de esta tarea, Suraweera et al. (2004a) presentan un trabajo en el que se utilizan ontologías. Aunque la idea es utilizar las ontologías como elemento inicial en un proceso de generación automático de restricciones, el trabajo presentado se centra sólo en extender WETAS con un entorno de trabajo para la construcción de la ontología. Los autores muestran que, aún creando las restricciones manualmente, el uso de la ontología y la reflexión asociada sobre los conceptos del dominio facilita la construcción de las mismas. Este trabajo es extendido en (Suraweera et al., 2004b), en el que se presenta el mecanismo de generación automática para la generación de restricciones sintácticas a partir de la ontología.

Posteriormente, el mecanismo de generación automático es completado mediante la generación de restricciones semánticas, además de las sintácticas. La extensión de WETAS con el mecanismo de generación automática de las restricciones se llamó **CAS** (*Constraint Acquisition System*) (Suraweera et al., 2005). El proceso de generación completo se compone de cuatro fases. En la primera, el autor especifica una ontología con cada concepto del dominio. En la segunda fase, se generan las restricciones sintácticas a partir de la ontología definida anteriormente. A continuación, se generan las restricciones semánticas mediante un algoritmo de aprendizaje automático que usa ejemplos de problemas y sus soluciones para corregir su base de conocimiento; la última fase, todavía en este trabajo no estaba implementada completamente, pero consistía en la revisión de las restricciones generadas proporcionando nuevos ejemplos de problemas y soluciones. La efectividad de la herramienta se demuestra en un trabajo posterior (Martin et al., 2007), en el que destaca el poco tiempo requerido para la construcción de sistemas tutores.

El mecanismo propuesto en CAS para mejorar WETAS siguió desarrollándose hasta completar todas las fases comentadas anteriormente y extendiendo su proceso hasta

contar con un sistema completo, al cual se llamó **ASPIRE**[5] (Authoring System for Developing Constraint-Based tutors) (Mitrovic et al., 2006, 2009). ASPIRE, además de ser una herramienta de autor, proporciona el entorno necesario para ejecutar los sistemas tutores construidos.

El proceso de autoría es el mismo que el mencionado para WETAS, con la particularidad de que en ASPIRE el proceso está completamente integrado y se refina a fases de desarrollo más específicas. La primera etapa consiste en especificar las características del tutor. Esto engloba, la definición de la estructura de dominio o secuencia de pasos, en dominios fuertemente procedimentales; construcción de la ontología del dominio siguiendo el mismo entorno desarrollado inicialmente para WETAS; modelado de las estructuras asociadas a los problemas y las soluciones; diseño de la interfaz, la cual puede ser textual, generada automáticamente por el sistema, o un applet específico proporcionado por el autor; y la adicción de problemas y soluciones ejemplo. La segunda gran etapa consiste en la generación semi-automática del modelo del dominio. Primeramente se generan las restricciones sintácticas y semánticas, y como paso necesario, éstas son supervisadas por el autor. Finalmente, el tutor generado se despliega en el servidor Web de ASPIRE y queda disponible para su uso. Las evaluaciones iniciales realizadas sobre ASPIRE muestran que la herramienta es capaz de generar en torno al 90 % de las restricciones que son requeridas para cubrir el modelo del dominio (Mitrovic et al., 2009). Lógicamente, en algunos dominios muy complejos, el mecanismo puede no llegar a generar la mayoría de las restricciones.

La utilización de ASPIRE no sólo se limita al grupo ICTG, sino que está disponible vía Web para usarse desde cualquier parte del mundo, previa autorización de los administradores. Esta disponibilidad ha motivado la construcción de muchos sistemas tutores, de los cuales, algunos de los más importantes, sobre los que se ha evaluado su efectividad, se pueden mencionar un tutor en el dominio del estudio de inversiones (Mitrovic et al., 2008); DM-Tutor (Decision Making Tutor) (Amalathas et al., 2010), en el dominio de la toma de decisiones para la gestión de plantaciones de aceite de palma; o Thermo-Tutor (Mitrovic et al., 2011), un sistema para la enseñanza de termodinámica.

Posteriormente, surge VIPER (*Virtual Interactive Practice Environment Resource*) (Martin y Mitrovic, 2008; Martin et al., 2009), una extensión de ASPIRE que trata de cubrir las necesidades de dominios en los que son requeridos elementos más específicos. Concretamente, permite construir dominios en los que la interacción está basada en imágenes, mediante la identificación de elementos dentro de la imagen o la comparación de imágenes. La ventaja del sistema es que permite especificar una plantilla, mediante una ontología a más alto nivel que la necesaria para las restricciones. La plantilla puede ser usada por otros dominios a la hora de construir sistemas tutores que comparten la misma estructura del dominio, interfaz, y lógica.

## 2.4. Conclusiones del capítulo

En este capítulo se han mencionado las características generales de los EIRP, destacando brevemente los tutores cognitivos, como uno de los paradigmas predominantes en la construcción de estos sistemas. El grueso del capítulo se centra en una revisión profunda del MBR, el otro paradigma predominante en el modelado del alumno en EIRP. Ambos paradigmas no se limitan al modelado del alumno, sino que establecen

[5] http://aspire.cosc.canterbury.ac.nz:8001/login

una forma de crear el resto de elementos de los STI, por lo que su utilización para la construcción de STI es preferente a cualquier otra técnica de modelado del alumno.

Tras la extensa revisión realizada sobre el paradigma MBR, la eficiencia de éste queda patente en la cantidad de estudios realizados que ponen de manifiesto su idoneidad como herramienta pedagógica (Mitrovic et al., 2001, 2007; Mitrovic, 2012). Estos estudios, como se ha podido ver, se extienden a diversas áreas de los STI con resultados satisfactorios, en las investigaciones más maduras, o prometedores, en las que todavía están en desarrollo. La principal ventaja es la relación eficiencia / simplicidad que proporciona. La simplicidad hace referencia a la facilidad requerida para aplicar la técnica, ya que se reduce a la definición de un conjunto de restricciones que modelan el conocimiento a enseñar y la utilización de un motor de inferencia que permita detectar los errores del alumno en las soluciones que éste construya.

Las herramientas más importantes usadas en los estudios revisados se resumen en la tabla 2.1. Para cada una se menciona: su nombre; el dominio de enseñanza donde se aplica; la clasificación del dominio y sus tareas en relación con la figura 2.1 (columna *Tipo*); y el número de restricciones del dominio (columna *Num. Rest*), el cual puede variar en puesto que se ha obtenido de las publicaciones existentes sobre estos sistemas y podrían existir nuevas versiones con distinto número.

| **Nombre** | **Dominio** | **Tipo** | **Num. Rest.** |
|---|---|---|---|
| SQL-Tutor | Consultas de bases de datos | DBTD | >700 |
| EER-Tutor | Diseño de bases de datos mediante modelo Entidad-Relación | DBTD | 39 |
| Collect-UML | Diseño Orientado a Objetos mediante UML | DBTD | 133 |
| CAPIT | Uso de mayúsculas y signos de puntuación en la gramática Inglesa | DBTB | 25 |
| ERM-Tutor | Paso de modelo Entidad-Relación a bases de datos relacionales | DBTB | 121 |
| J-LATTE | Programación Orientada a Objetos en Java | DBTD | 88 |
| WETAS | Multidominio: entorno de ejecución de sistemas MBR | - | - |
| ASPIRE | Multidominio herramienta de autor genérica | - | - |

Tabla 2.1: Resumen de herramientas MBR.

Comparando las dos técnicas principales de modelado se pueden observar claramente diversas similitudes y diferencias. Por un lado, el mecanismo para proporcionar la instrucción inmediata, sea la TM en los tutores cognitivos, o el refuerzo ante las restricciones en el MBR, no difiere en su aplicación, pues ambos se basan en realizar

una comparación de patrones. En cuanto a la construcción del modelo de dominio es evidente que se requiere menos esfuerzo en el MBR que en los tutores cognitivos, puesto que codificar todos los principios del dominio es más fácil que modelar todos los posibles pasos erróneos, lo cual podría ser una tarea muy costosa o incluso imposible de completar en los dominios débilmente definidos. Desde un punto de vista más técnico, la implementación de la interfaz y la puesta en común de las diversas componentes puede ser la parte más costosa del MBR, a menos que se utilicen algunas de las herramientas de autor mencionadas, pero en ese sentido, cualquier otra técnica requiere el mismo o más esfuerzo (Mitrovic et al., 2003). La diferencia principal entre las dos técnicas radica en el modelo del alumno que se usa como base para la instrucción: mientras que en los tutores cognitivos la TM utiliza un enfoque probabilístico, en los sistemas MBR esta parte utiliza heurísticos susceptibles de mejora.

Es en el mecanismo que diagnostica el conocimiento del estudiante donde se encuentra la limitación más importante del MBR. Éste está basado en una función heurística muy subjetiva que, además, es extremadamente simple. Tal y como se explicaba en la sección 2.3.3, este nivel es proporcional al número de restricciones que el alumno sabe. A su vez, se considera que se sabe una restricción si se ha satisfecho más del 70 % de las veces que ésta ha sido relevante en un periodo de tiempo, lo cual es bastante subjetivo dado que 1) el umbral que determina si se sabe la restricción podría ser inferior, superior, o incluso dinámico de acuerdo a diversas características del estudiante; y 2) el periodo de tiempo se toma o bien al principio o bien al final, sin estar justificado el porqué es idóneo ese intervalo. Ohlsson y Mitrovic (2006) ya señalan que, para permitir que los sistemas mejoren la toma de decisiones pedagógicas en el MBR, es necesario contar con un modelo del estudiante a largo plazo en lugar del mencionado, con una naturaleza más a corto plazo. Esta componente susceptible de mejora es similar a la TC de los tutores cognitivos, con el inconveniente de que, en el caso del MBR, no sigue una base bien fundamentada.

La forma de determinar si un alumno sabe una restricción afecta también a la adaptación de los sistemas MBR, pues la estimación de la dificultad de los problemas se realiza en base al mismo principio mencionando anteriormente, al usar la suma ponderada de la probabilidad de que el estudiante haya aprendido cada restricción relevante en el problema. Esta forma de determinar la dificultad de un problema, no refleja su verdadera dificultad y hace que, al utilizarse para seleccionar el problema siguiente en un entorno real, se seleccionen problemas inadecuados, o bien muy simples, o muy complejos, como mencionan Mayo y Mitrovic (2000). El problema se encuentra, además de en la forma de decidir si se sabe una restricción, en que los pesos de la ponderación se determinan a partir del número de predicados que contiene la restricción, lo cual no es un reflejo objetivo, ni formal, de la importancia o influencia que pueda tener una restricción en la dificultad del problema.

En un intento por corregir esta situación se utilizaron técnicas objetivas que combinan la teoría de decisión estadística (Savage, 1954) con las redes bayesianas (Bayes, 1763). La investigación realizada por Mayo y Mitrovic (2000, 2001); Mitrovic et al. (2002), aunque utiliza un enfoque formal que resulta efectivo, tiene varios inconvenientes que la hacen impracticable en sistemas reales. Por un lado, la aplicación de este método va en detrimento de una de sus mayores ventajas en su formulación original: la facilidad de aplicación y el esfuerzo requerido para poner en práctica el modelo es elevado. Por otro lado, una limitación existente, en cuanto a la escalabilidad del modelo, hace que sólo sea aplicable a sistemas con un número reducido de restricciones.

El uso de otras técnicas formales para determinar la dificultad de un problema objetivamente y para realizar el diagnóstico del estudiante, se plantea como un elemento necesario para hacer que el paradigma del MBR proporcione un proceso de enseñanza objetivo y bien fundamentado con una mayor efectividad tutorial. Principalmente, sería necesario utilizar otros mecanismos para estimar el nivel del alumno, lo cual reforzaría la base sobre la que se asienta todo el proceso de adaptación del paradigma, el cual incluye la selección adaptativa de problemas, presentación de refuerzo adecuado y cualquier otra toma de decisiones pedagógicas que fuera necesaria.

Otras limitaciones mencionadas anteriormente y de menor importancia sobre el paradigma del MBR se encuentran en el uso de las restricciones semánticas que en algunos dominios requiere de un módulo de resolución de problemas, al igual que pasa con los tutores cognitivos. Además, para que el MBR alcance su máximo potencial es necesario contar con soluciones altamente informativas diagnósticamente, las cuales proporcionan información suficiente para realizar el diagnóstico del conocimiento. Por ejemplo, una solución compuesta de un único número no es altamente informativa porque no contempla los principios involucrados en el proceso de resolución. En dominios con este tipo de soluciones se suelen usar pasos de resolución y *restricciones camino* que, como se explicaba en el apartado 2.3.2, se corresponden a las reglas de producción de la TM. Éstos son pequeños inconvenientes que no reducen la flexibilidad, eficiencia y aplicabilidad del MBR (Mitrovic, 2012).

# Capítulo 3

# Mecanismos de evaluación

*Lo importante es no dejar de hacerse preguntas*
Albert Einstein (1879 - 1955)

**Resumen:** Dado que uno de los objetivos de la tesis es diseñar una metodología de evaluación formal del conocimiento, en este capítulo se hace una revisión de los mecanismos existentes que permiten realizar este tipo de evaluación.

La evaluación y la determinación de aptitudes psicológicas del ser humano, en general, es una característica que ha sido explorada desde tiempos remotos. En uno de los primeros intentos de medir las características de la personalidad humana, sobre el 400 a.C., Hipócrates clasificaba el temperamento en cuatro tipos diferentes en base a los diferentes humores del cuerpo. En el estudio de las habilidades mentales destaca Sir Francis Galton, en el siglo XIX, donde sus estudios comparativos de la habilidad se consideran el origen de la Psicología diferencial. Estos estudios motivan la posterior aparición de los tests mentales, término que utiliza por primera vez James McKeen Cattell y que es ampliado posteriormente en los estudios de Binet et al. (1913) mediante su escala para determinar la inteligencia usando la edad mental.

Esta tradición de elaboración de escalas y determinación de las habilidades mentales de las personas se ha mantenido hasta nuestros días. En la actualidad, el campo que se encarga de la medición de las dimensiones o rasgos psicológicos de la persona como la inteligencia, el dominio en una cierta materia, cierto rasgo de la personalidad, etc.; es la Psicometría. Esta disciplina, situada dentro de la Psicología, además del estudio de métodos para la medición de rasgos psicológicos, se ocupa del estudio de la fiabilidad y validez de los mismos con el objetivo de hacer que los instrumentos de medida posean la mayor calidad, formalidad y objetividad posible.

Dentro de la Psicometría, la herramienta por excelencia para llevar a término su objetivo es el *test*. Un test es una prueba que pretende medir una cierta característica de un individuo. Para ello, se utilizan una serie de preguntas como instrumento de medida, las cuales plantean una cuestión al individuo y recogen una respuesta que sirve para realizar la medición. Atendiendo a lo que se desea medir se distingue una gran variedad de tipos de tests. Así pues, los tests de personalidad o normativos miden componentes afectivos de la persona; los tests clínicos comprueban si hay trastornos psicológicos; los tests de logros miden el grado de consecución de ciertos objetivos;

los tests de aptitud determinan la capacidad de una persona en el desempeño de una tarea; los tests proyectivos buscan predecir el comportamiento de una persona; los tests de inteligencia miden el cociente intelectual; los tests ipsativos, normalmente usados en procesos de selección, determinan las preferencias de un individuo; y así un largo etcétera (Olea et al., 2010).

Para esta tesis, como se podrá observar, el tipo de test no es importante, sino su utilización como instrumento de medida, el cual estará orientado a la medición del conocimiento en un ámbito educativo. Es por ello que se puede establecer el siguiente convenio de términos que serán usados principalmente en este capítulo y, en menor medida, en el resto de la tesis: para referirnos a los individuos de un test, se usará el termino sujeto, alumno, individuo, estudiante, o evaluado; las preguntas son comúnmente referidas en el campo como ítems; y el rasgo a medir, como normal general, es el conocimiento.

Aunque tradicionalmente los tests se administraban mediante papel, y todavía es una práctica extendida en la actualidad, con el avance de la tecnología, a partir de los 80, se comenzó a utilizar el denominado *Test Administrador por Ordenador* (TAC), el cual se refiere a la administración y realización de los tests en el ordenador. El salto de los test a la plataforma tecnológica proporciona numerosas ventajas frente a la administración tradicional a papel (Hontangas et al., 2000; Olea, 2002; Olea et al., 2010) tales como: la posibilidad de estandarizar las condiciones de aplicación de los tests para todos los evaluados; se obtiene un procesamiento rápido de los datos, proporcionando una evaluación inmediata. Además, permite establecer controles para preservar la seguridad de la prueba; registrar información útil para la evaluación, como los tiempos de respuesta; minimizar errores de corrección; proporcionar refuerzo en los ítems; y añaden las características de accesibilidad y disponibilidad propia de las plataformas Web, haciendo que los tests estén disponibles desde cualquier parte y en cualquier momento.

El principal inconveniente que presentan los tests informatizados es que no permiten evaluar tareas de resolución complejas, o dominios procedimentales (Marzano, 1990). El uso del ordenador favorece la elaboración de ítems innovadores (Parshall et al., 2010), las cuales permiten utilizar elementos multimedia y proporcionan cierta complejidad en la interacción con el alumno como la ordenación de elementos, emparejar, interacción con imágenes (Conejo et al., 2004). No obstante, estos ítems distan mucho de los procesos de resolución complejos que cualquier dominio puede contener. Esta problemática es una de las principales que motivan el trabajo presentado en esta tesis.

El hecho de añadir la dimensión informática a los tests posibilita la aparición de nuevos modelos de administración de los tests (en inglés *test delivery models*), entre los que destacan (Luecht y Sireci, 2011): los *tests lineales* o de forma fija, los cuales tienen una serie predeterminada de ítems a mostrar, independientemente del nivel del alumno; los tests *lineales al vuelo* (en inglés *linear-on-the-fly*) los cuales son de forma fija pero se seleccionan todos los ítems que lo componen en el momento en que un estudiante comienza un test; los *testlets* (Rosenbaum, 1988), que son un conjunto de ítems administrados juntos y considerados como una unidad; los *test adaptativos*, los cuales se adaptan al rendimiento del estudiante; y los *tests en la sombra* (en inglés *shadow tests* (van der Linden, 2010), que son como los tests adaptativos pero la selección se basa en la construcción de un test completo por cada ítem candidato, usando el ítem cuyo test "en la sombra" maximiza un criterio concreto.

Como en cualquier instrumento de medida, donde debe haber una forma de estable-

cer la precisión y de determinar cómo de fiables son las mediciones realizadas, los tests también deben tener esta característica, dado que las decisiones que se toman en base a los resultados pueden afectar la vida de las personas. Es por ello que el funcionamiento de los tests debe estar guiado por teorías bien fundamentadas que tengan en cuenta los errores que se pueden producir durante la medición. Estas teorías, denominadas *teorías de tests*, definen modelos estadísticos y matemáticos que, bajo una serie de suposiciones y planteamientos, permiten estudiar las propiedades psicométricas como la precisión de las mediciones (Suen, 1990). Hasta nuestras fechas dos son las teorías de tests que destacan: la *Teoría Clásica de Tests* (TCT), centrada en las propiedades y características de los tests; y la *Teoría de Respuesta al Ítem* (TRI), centrada en las propiedades de los ítems. Ambas teorías serán tratadas posteriormente.

En este capítulo se presenta una revisión de los métodos formales que permiten determinar el conocimiento del alumno. Primeramente, la siguiente sección hace un breve repaso sobre la TCT, lo cual introducirá al lector en el funcionamiento de las teorías de tests y permitirá apreciar mejor las ventajas de la TRI. Esta última, dada su importancia para esta tesis, será explicada en detalle en la sección 3.2. Posteriormente, en la sección 3.3 se explicarán los tests adaptativos, centrándose en la forma de uso de la TRI y las aportaciones que esta teoría proporciona a este tipo de tests, las cuales están directamente relacionadas con esta tesis. Seguidamente, en la sección 3.4, se revisan otras técnicas utilizadas para la evaluación y que están relacionadas con el trabajo de esta tesis. Finalmente, se presentan las conclusiones de este capítulo en la sección 3.5.

## 3.1. Teoría Clásica de los Tests

La *Teoría Clásica de Tests* (TCT) o también conocida como *Teoría de puntuación verdadera* (Muñiz, 2003) (en inglés *Classical Test Theory*), es un compendio de teorías de test, las cuales son teorías y definiciones de modelos que permiten determinar las puntuaciones en los tests. Aunque sus orígenes se sitúan a principios del siglo XX con los trabajos de Spearman (1904, 1907), no es hasta la década de los 60 cuando se publica el modelo clásico final en (Lord y Novick, 1968).

En los trabajos iniciales que motivaron el desarrollo de la TCT se intentaba explicar un fenómeno que había sido observado empíricamente: algunos conjuntos de ítems parecían dar unos resultados más consistentes que otros. La explicación que se dio es que la puntuación total observada en un instrumento de medida, $X$, estaba compuesta de la suma de dos variables latentes no observables: una *puntuación verdadera*, $PV$, y un *error de medida* $E$. Este error de medida representa errores no sistemáticos como distracción, motivación, ansiedad, etc., los cuales tienen un efecto negativo sobre la puntuación verdadera, algunas veces, y positivo otras. La relación entre la puntuación observada y las variables no observables se expresa mediante la ecuación siguiente:

$$X = PV + E \tag{3.1}$$

La puntuación verdadera puede ser vista como la componente sistemática de la puntuación total. La TCT supone que las mediciones se dispersan uniformemente sobre la media, lo que significa que las desviaciones ocurren equitativamente a ambos lados de la puntuación verdadera. Como consecuencia, la puntuación verdadera es en realidad la puntuación media. Esto significa que el error medio significa una distribución normal de media 0, ya que los valores positivos y negativos de las desviaciones se cancelan entre sí; y varianza desconocida.

El modelo mencionado es conocido como *modelo lineal clásico* de Spearman. Según esta formulación, se tiene un valor, el proporcionado por $X$, pero dos incógnitas ($PV$ y $E$). Para poder avanzar y obtener el valor de las incógnitas Spearman añade tres supuestos y una definición (Muñiz, 2010):

- En el primer supuesto se define la puntuación verdadera como la esperanza matemática de la puntuación observada. Es decir, $PV = E(X)$. Conceptualmente esto significa que se define la puntuación verdadera de una persona en un test como aquella puntuación que obtendría como media si se le pasase infinitas veces el test, lo cual es algo teórico.
- El segundo supuesto consiste en asumir que no hay ninguna relación entre la puntuación verdadera de una persona y los errores que afectan a esas puntuaciones. Es decir, puede haber puntuaciones verdaderas bajas / altas con errores bajos o altos, indiferentemente, sin existir una conexión entre las dos variables. Esto se puede expresar como $r(pv, e) = 0$.
- El tercer supuesto postula que los errores de medida que tienen lugar en un test no están relacionados con los que ocurren en otro. Es decir, $r(e_i, e_j) = 0$.
- La definición que añade Spearman es la de los *test paralelos*. Estos tests son aquellos que miden lo mismo pero con diferentes ítems, lo que implica que la puntuación verdadera y las varianzas de los errores de medida en los tests paralelos tienen el mismo valor.

Los tres supuestos son razonables pero no se pueden comprobar empíricamente de forma directa, sino que las deducciones que luego se hagan a partir de éstos permitirán confirmarlos o denegarlos.

### 3.1.1. Fiabilidad de un test

El objetivo de la TCT es evaluar la calidad de la puntuación observada del test mediante la fiabilidad de la medición, la cual se basa en los supuestos anteriores. El concepto de la fiabilidad en esta teoría establece que si una persona, que realiza el mismo test en un corto espacio de tiempo, tiene variaciones en la puntuación observable, entonces este test es más probable que refleje los errores no sistemáticos en lugar de la puntuación verdadera y, por tanto, sería menos fiable. Teóricamente la fiabilidad se representa mediante la siguiente ecuación (Abad et al., 2006):

$$Fiabilidad(X) = \frac{\sigma^2(PV)}{\sigma^2(X)} \tag{3.2}$$

No obstante, dado que no se conoce la puntuación verdadera, es necesario usar estimadores de la fiabilidad. De forma general, se utiliza un método para obtener varias mediciones diferentes del conocimiento y se estudia la correlación entre los resultados, la cual equivale a la fiabilidad (Lord y Novick, 1968). Estas mediciones se pueden conseguir de diferentes formas:

- Usando medidas repetidas mediante tests paralelos, los cuales son difíciles de construir porque tiene que garantizarse que los ítems deberían ser iguales en términos de lo que miden, pero no pueden ser idénticos para evitar que formen el mismo test.

- Repitiendo el mismo test. En este caso debería haber el tiempo suficiente para prevenir que los individuos recuerden el test, pero no demasiado para garantizar que no hay cambio en el conocimiento entre las dos presentaciones del test, de lo contrario, el uso de la correlación para calcular la fiabilidad no sería válido. Este último requisito es muy fácil de violarse, ya que es muy probable que los alumnos intercambien opiniones y "aprendan" de una vez para otra acerca de un mismo test.

- Usando una única medida, pero dividiendo el test en dos mitades aleatorias iguales en longitud y preferiblemente en dificultad (mitades paralelas). Esta forma es más eficiente que el uso de varias medidas y permite determinar la fiabilidad de las dos mitades mediante la correlación entre las puntuaciones. Para calcular la fiabilidad del test completo es necesario usar alguna corrección como la fórmula de predicción de Spearman-Brown (Surhone et al., 2010).

- Para prevenir problemas ocasionados por dos mitades no paralelas en el método anterior, se puede usar el método de consistencia interna que se basa en considerar los ítems como si fueran tests. Tras la administración de un test completo se comparan cada par de ítems mediante algún método de consistencia como el coeficiente $\alpha$ de Cronbach (1951), el coeficiente $K - R20$ de Kuder y Richardson (1937), o los *límites inferiores de la fiabilidad* de Guttman (1945).

Además, de la fiabilidad, otro índice fundamental en la TCT es el *error estándar de medida* (EEM). Aunque ambos son estimaciones matemáticas del error, contienen diferente información que puede ser utilizada para estudiar la consistencia un test. El EEM se calcula como la relación entre la desviación típica del error y la desviación típica de puntuación observada en el test, la cual es estimada a partir de la fiabilidad mediante la fórmula de la ecuación 3.3. Esta media refleja la discrepancia entre la puntuación observada y la verdadera, y es también un buen índice de la consistencia del test.

$$EEM = \frac{\sigma(E)}{\sigma(X)} = \sqrt{1 - Fiabilidad(X)} \tag{3.3}$$

### 3.1.2. Alternativas a la TCT

Además del enfoque clásico propuesto mediante el modelo lineal de Spearman, surgen otros modelos que, basándose del anterior, modifican las componentes implicadas y, por tanto, establecen un conjunto propio de métodos para la determinación de la fiabilidad y validez. Entre los modelos más relevantes se pueden destacar:

- La *teoría de la generalizabilidad* (Webb y Shavelson, 2005), también conocida como teoría G, en lugar de usar la variable $E$ del modelo clásico, considera que la variabilidad de la puntuación puede deberse a varias fuentes de error, denominadas facetas. La ventaja frente al modelo clásico se encuentra en que se puede estimar qué proporción de la varianza total del resultado es debida a cada faceta. Esta teoría utiliza un coeficiente de generalizabilidad que hace las veces de la fiabilidad en el modelo clásico de la TCT.

- Los *tests referidos al criterio*, originados por Glaser (1963), tienen como objetivo determinar si las personas dominan un criterio concreto o tema de conocimiento.

En lugar de discriminar entre las personas, como la mayoría de los tests psicológicos, se centran en evaluar el grado del conocimiento en el tema de conocimiento denominado criterio.

- La *teoría fuerte de la puntuación verdadera* es una extensión de la TCT en donde los modelos que se definen utilizan supuestos más restrictivos. Uno de los modelos más populares en este enfoque es el *binomial de Lord* (Lord, 1955), el cual se puede considerar el precursor de la teoría fuerte. Este modelo es conocido por estimar el error estándar condicional de las puntuaciones observadas en los tests y establece un supuesto más restrictivo sobre la distribución de los errores.

- El *análisis factorial* (Child, 1990), que difiere de la TCT en que los modelos asociados consideran una descomposición multidimensional de las variables de puntuación verdadera en factores. No todos los modelos dentro de esta categoría están basados en la TCT. En general, este enfoque implica la obtención de un conjunto de $n$ medidas sobre los mismos individuos, calcular una matriz de correlación entre estas medidas de tamaño $n \times n$, y usar técnicas de análisis factorial para identificar un número reducido de factores.

### 3.1.3. Ventajas e inconvenientes de la TCT

La principal ventaja de esta teoría proviene de su simplicidad de aplicación, puesto que los modelos matemáticos sobre los que se define son relativamente sencillos de llevar a cabo. También, a diferencia con la TRI, el tamaño muestral requerido para realizar los análisis puede ser mucho menor. Otra ventaja se encuentra en la denominación común de los modelos de la TCT, la cual los define de forma general como *modelos débiles*, en el sentido de que los supuestos de esta teoría no son muy estrictos y son fácilmente satisfechos por los métodos de evaluación tradicionales (Hambleton y Jones, 1993).

Una de las principales limitaciones de la TCT se encuentra en que las características del test y de las personas no pueden ser separadas. Esto es así porque la puntuación de una persona se define como el número de ítems que acierta y la dificultad de un ítem como la proporción de personas que lo responden correctamente en un determinado grupo. Esto hace que los resultados sean dependientes del grupo donde se aplique cierto test y que la puntuación no pueda ser comparable con la obtenida en otro test diferente, ya que cada uno tiene una escala propia en la que los ítems tendrán una dificultad diferente.

Otro inconveniente de la TCT está relacionado con la fiabilidad del test, la cual se mide realizando el test varias veces de forma paralela. El problema está en que la construcción de los tests paralelos es bastante difícil, por no decir imposible. Además, la fiabilidad de un test influye también sobre el error estándar de medida, que es considerado igual para todos los individuos que realizan el test, asunción que es inadmisible (Hambleton et al., 1991).

Una última limitación de esta teoría es que se centra en el test en su conjunto, imposibilitando un análisis individual sobre los ítems. Es decir, el valor alcanzado por el individuo en el test sólo permite dar una valoración global sobre dicho test, pero impide realizar predicciones sobre el comportamiento de las personas ante un ítem concreto o determinar la probabilidad de que una persona responda correctamente a un ítem determinado. Además, la TCT contempla solamente los errores no sistemáticos, descartando los errores sistemáticos que afectan de igual manera a mediciones diferen-

tes. Estos errores se asocian a ítems no adecuados o incorrectamente diseñados, siendo imposible detectarlos con la formulación de la TCT.

## 3.2. Teoría de Respuesta al Ítem

La *Teoría de Respuesta al Ítem* (TRI) (referida en inglés como *Item Response Theory*) (Hambleton et al., 1991; Embretson y Reise, 2000; Baker, 2001; DeMars, 2010) es un desarrollo relativamente reciente de la rama de la Psicometría. Al ser una teoría de tests, se refiere a un modelo para la medición de aspectos de la mentalidad humana a partir de la respuesta a ítems. Como menciona Bock (1997), los fundamentos de esta teoría se encuentran en los trabajos de Thurstone (1925, 1927), con la elaboración de la primera escala en la que situar los ítems de un test y el denominado método de *juicio comparativo* para realizar la evaluación de tests; Lazarsfeld (1950a,b), donde se introdujo el término "rasgo latente", en referencia a los rasgos mentales no observados; y Lord (1952), que estudió la relación entre las variables latentes con las respuestas a los ítems y las puntuaciones obtenidas en los tests.

La popularidad de la TRI comienza en los 60 y, de acuerdo a Georgiev (2008), sigue dos líneas separadas de desarrollo: la primera línea se remonta a (Lord y Novick, 1968) donde se introducían métodos de medida precisos; y la segunda rama es la iniciada por Rasch (1960) mediante una familia de modelos para la medición y el desarrollo de tests. El trabajo de Rasch fue continuado por otros importantes psicómetras en el área que extendieron su trabajo como Fisher o Wright. En cualquier caso, la consistencia de sus resultados respecto de la TCT, ha convertido a esta teoría en una de las más importantes de la Psicometría. La principal ventaja respecto a su antecesora radica en que la estimación del conocimiento se centra en las propiedades individuales de los ítems, haciendo que el modelo sea más preciso y fiable y no sea dependiente de la población donde se aplica. Una comparación detallada se presenta en la sección 3.2.3.

La TRI se basa en dos postulados principales (Hambleton et al., 1991): primero, el conocimiento de un estudiante en un ítem de un test puede explicarse mediante una serie de factores llamados rasgos latentes o habilidades, los cuales se asocian a los rasgos de la personalidad no observables y que se pretenden medir con un test; y segundo, la relación entre la probabilidad de dar un tipo de respuesta (variable observable) y el rasgo implícito en la respuesta a un ítem (variable no observable) puede describirse mediante una función denominada *Curva Característica del Ítem* (CCI). Esta función representa las propiedades individuales de los ítems que se mencionaban anteriormente y que le confieren las ventajas sobre la TCT. El concepto de CCI es el central de la TRI, ya que sobre éste se apoyan el resto de elementos de la teoría. Concretamente, la CCI se define como sigue:

**Definición 3.1** (Curva Característica de un Ítem)**.** *La CCI es una función monótona creciente que indica la probabilidad condicional de que un estudiante, con cierto rasgo latente estimado ($\theta$), responda correctamente a un ítem. La CCI de un ítem $i$, denotada por $P_i(\theta)$, es una función de densidad de probabilidad y, como tal, se define en el rango de los reales $(-\infty, .., \infty)$ y tiene como dominio el intervalo $[0, 1]$.*

A la hora de utilizar la TRI hay tres características que son requeridas para que la aplicación de la misma sea correcta. Estas características son referidas normalmente como los *supuestos de la TRI* (Hambleton et al., 1991), algunos de los cuales deben cumplirse ya que la formulación de los mecanismos de la TRI los dan por supuestos.

- Independencia local: Este supuesto establece que cuando las habilidades que influyen en el resultado de un test se mantienen constantes, las respuestas a cualquier par de ítems son estadísticamente independientes (Hambleton et al., 1991). En otras palabras, no debería haber relación entre el éxito o fallo de un alumno entre dos ítems cualesquiera. Este supuesto se puede formalizar considerando $\theta$ el conocimiento influyente en las respuestas de un alumno en un test mediante la siguiente fórmula:

$$P(U_1, U_2, ..., U_n|\theta) = P(U_1|\theta)P(U_2|\theta)...P(U_n|\theta) \tag{3.4}$$

  Donde, $P(U_i|\theta)$ es la probabilidad de que un alumno responda al ítem $i$. La igualdad de la ecuación establece que la probabilidad de que ocurra un patrón de respuestas, asociado a un conjunto de ítems en un test, es igual al producto de las probabilidades asociadas con las respuestas del examinado a los ítems individuales.

- Unidimensionalidad: Esta suposición determina que sólo un rasgo latente es medido por un conjunto de ítems en un test. Este supuesto no puede cumplirse estrictamente porque diversos factores cognitivos de la personalidad y asociados al proceso de realización del test afectan al resultado. Entre estos factores se encuentra la motivación, ansiedad, tendencia a responder aleatoriamente, habilidades cognitivas, concentración, etc. Dado que éstos siempre están presentes en mayor o menor medida, para que este supuesto sea adecuadamente satisfecho, la condición requerida es que un rasgo latente influya en mayor medida al resto en el resultado del test, siendo éste el que se desea medir.

- Invariancia: Muchos autores sólo identifican como supuestos los dos ya mencionados, dejando este elemento como una propiedad deseable. Sin embargo, otros autores sí que lo identifican como un supuesto requerido por la TRI. Esta propiedad establece que el rasgo latente que se esté midiendo para un alumno debe ser el mismo, independientemente del test que sea usado para ello. En modelos paramétricos, que se explicarán a continuación, esta invariancia se aplica también a la estimación de los parámetros. De esta forma, los valores estimados para los parámetros representando la CCI de un ítem deberían ser los mismos independientemente del grupo de alumnos que se utilicen, siempre y cuando la muestra poblacional usada para la estimación sea representativa. Además, durante el proceso de medición del alumno no debería haber aprendizaje para garantizar que el rasgo latente que se está midiendo permanece constante.

Los modelos de la TRI son robustos a violaciones menores de estos supuestos. Además, ningún dato real cumple los supuestos a la perfección (Reeve y Fayers, 2005), con lo que el objetivo principal de estos supuestos es cumplirlos lo mejor posible.

### 3.2.1. Modelos de la TRI

Tal y como se recoge en trabajos como los de Baker (2001); DeMars (2010), existen numerosos modelos de la TRI que condicionan el funcionamiento de los mecanismos usados para la evaluación del alumno. A lo largo de este apartado se mencionan los modelos más importantes, clasificándolos en base a sus características generales. De esta forma, se distinguen tres criterios de clasificación: 1) dependiendo del número de

rasgos latentes que modelan; 2) según el tratamiento de la respuesta; 3) según la función utilizada para aproximar la CCI. Nótese que dado un modelo concreto, éste estará dentro de una de las posibles categorías que cada criterio de clasificación establece, pues son características no excluyentes.

#### 3.2.1.1. Según el número de rasgos latentes

Dependiendo del número de rasgos latentes que un modelo permite determinar a partir de los ítems de un test, se distinguen dos tipos de categorías: los modelos unidimensionales, que sólo evalúan un rasgo latente; y los multidimensionales que permiten la determinación de varios rasgos latentes. Los primeros son referidos como *UIRT* (del inglés *Unidementional Item Response Theory*), mientras que los últimos se nombran mediante *MIRT*, por su correspondiente terminología en inglés.

En los modelos MIRT las CCI son multidimensionales, con tantas dimensiones como rasgos latentes estén implicados en el modelo, lo cual hace que su manejo sea mucho más complejo que los UIRT. Aunque los modelos MIRT son la combinación de diversas ideas provenientes de la Psicología, Psicometría, desarrollo de tests, educación, y estadística, las principales fuentes de las que parten son los modelos UIRT y las técnicas de análisis factorial (Reckase, 2009). Es por ello que los modelos UIRT han sido estudiados más en profundidad y son más populares que los MIRT.

Como se mencionaba en las alternativas a la TCT, la determinación de los diversos rasgos latentes se realiza a partir de un conjunto de $n$ medidas sobre los mismos individuos. Una vez se obtienen estas medidas, se construye una matriz de correlaciones que permite, aplicando técnicas de análisis factorial, medir un número reducido de rasgos latentes (Rupp y Templin, 2010). Una descripción muy detallada de los modelos MIRT, su utilización en sistemas de tests, y una recopilación de los últimos avances en el área se puede encontrar en (Reckase, 2009).

#### 3.2.1.2. En base al tratamiento de la respuesta

Otra de las diferenciaciones que distinguen a los modelos de la TRI es la forma en que se trata una respuesta asociada a un ítem. Atendiendo a este criterio, los modelos posibles son:

- *Dicotómicos o binarios*: Dentro de este tipo se engloban la mayoría de los modelos de la TRI y son los más usados por ser más simples de aplicar. Estos modelos consideran que la respuesta dada a un ítem puede tomar dos valores posibles, o bien ésta es correcta, o bien es incorrecta. Así pues, la opción que el alumno selecciona como respuesta no se contempla en estos modelos, sino el resultado asociado.

- *Politómicos*: Los modelos de este tipo tienen en cuenta la respuesta seleccionada por un alumno. Por este motivo, cada respuesta que puede darse en un ítem tiene asociada una curva característica que expresa la probabilidad de que un alumno, con un nivel de conocimiento $\theta$ dado, seleccione esa respuesta. Estos modelos suelen ser utilizados en tests de personalidad o aptitud, pues no se supone que existan respuestas correctas o incorrectas, sino que la diferencia entre las alternativas de respuesta es la intensidad con la que se debe poseer el rasgo medido para responder cada una de ellas.

Aunque los primeros modelos politómicos datan de la década de los 60, no es hasta la aparición de la herramienta MULTILOG (Thissen et al., 2003) y del desarrollo tecnológico cuando los modelos politómicos se hacen computacionalmente aplicables. Inicialmente fueron diseñados para modelar los ítems de opción múltiple de forma que se contemplase la respuesta dada por el alumno. Sin embargo, en la actualidad se utilizan sobre todo en ítems con respuestas ordenadas en diferentes categorías, como los tipo Likert.

La principal desventaja de los modelos politómicos se encuentra en la dificultad de estimación de los parámetros de las opciones de cada ítem, ya que es necesario un conjunto de datos lo suficientemente amplio como para realizar una calibración correcta. En (Hontangas et al., 2000) se menciona que son necesarios del orden de cinco individuos por cada parámetro que se desea estimar en los modelos politómicos, motivo por el que los modelos dicotómicos son los que se utilizan principalmente en los TAI. Como contrapartida, la principal ventaja de los modelos politómicos es que proporcionan más información que los dicotómicos puesto que se añade un nivel más de información que permite analizar las respuestas de los ítems.

Los modelos dicotómicos más importantes serán explicados posteriormente en el apartado 3.2.1.3. En relación con los politómicos, aunque hay diversos modelos, como puede consultarse en (Nering y Ostini, 2010), sólo se mencionan a continuación los tres más destacados:

- *Modelo de Respuesta Graduada General* (En inglés *General Graded Response Model*) (Samejima, 1969, 2010): Este tipo de modelos fue creado para analizar ítems politómicos cuyas respuestas pertenecen a categorías ordenadas tipo Likert. Originalmente llamado *Modelo de Respuesta Graduada*, es una generalización del modelo 2PL, que será explicado en el apartado 3.2.1.3, en la que los ítems pueden tener un número diferentes de categorías de respuesta. Estos modelos introducen las denominadas curvas características de funcionamiento que representan la probabilidad de que un alumno responda a una categoría superior o igual a una dada. Un ítem según este modelo es tratado como varios ítems dicotómicos, tantos como categorías posibles, donde los dos valores de respuesta se asocian a que la respuesta sea una categoría o el resto. A partir de esta formulación, y con algunas restricciones, se determinan las curvas características de funcionamiento y posteriormente las curvas características de la respuesta. El principal problema de estos modelos es que requieren la estimación de una gran cantidad de parámetros.

- *Modelo de Respuesta Nominal* (en inglés *Nominal Response Model*) (Bock, 1972; Thissen et al., 2010): En este modelo se tratan ítems con dos o más categorías nominales, es decir, no se supone que exista orden en las categorías que pueden darse como respuesta. El problema del modelo es que exige el cálculo de muchos parámetros, por lo que es necesaria una muestra grande para que las estimaciones sean precisas. Como ventaja, la imposición de pocas restricciones a los datos tiende a ajustarse frecuentemente a las respuestas de los individuos.

- *Modelo de crédito parcial generalizado* (en inglés *Generalized Partial Credit Model*) (Muraki, 1982): Es un modelo en el que los ítems son conceptualizados como una serie de pasos ordenados donde el alumno recibe crédito parcial por realizar correctamente un paso concreto. De esta forma se modelan ítems que están

asociados a un proceso de resolución con diferentes pasos para alcanzar una respuesta. El modelo se formula bajo el supuesto de que la probabilidad de elegir la categoría i-ésima respecto de la categoría (i-1)-ésima es modelada mediante un modelo dicotómico. Como su nombre indica, es una generalización del *modelo de crédito parcial* de Masters (1982, 2010), con la diferencia de que se relaja el supuesto de discriminación uniforme proveniente del modelo dicotómico. Para ello, se introduce un parámetro de discriminación en la probabilidad de respuesta de un individuo. Dicho parámetro representa la facilidad con que se puede pasar de una categoría a otra en cada ítem según el valor del rasgo latente que posean los individuos.

#### 3.2.1.3. Según el tipo de función

Esta clasificación discrimina los modelos en base a la función utilizada para representar la probabilidad de la CCI. Aunque teóricamente, atendiendo a este criterio, se pueden construir infinitos modelos, puesto que funciones matemáticas para emplear hay muchísimas, se pueden agrupar en varias categorías que representan los más utilizados. De forma general, se puede distinguir entre modelos paramétricos y no paramétricos, dependiendo de si la CCI está definida por parámetros o por valores, respectivamente. Dentro de los primeros destacan dos categorías principales: los *modelos normales*, que usan la función de distribución normal para el modelado de la curva; y los *modelos logísticos*, que usan la función de distribución logística.

##### Modelos paramétricos

En esta categoría entran las funciones que definen la probabilidad de responder correctamente mediante una serie de parámetros. Normalmente se suelen usar tres parámetros típicos, cada uno tiene un significado y su valor afecta de manera diferente a la curva del ítem:

- $a_i$ o índice de discriminación del ítem, el cual es un valor proporcional a la pendiente de la recta tangente en el punto de inflexión de la CCI, el cual corresponde al valor de la curva para $\theta = b_i$. Cuanto mayor sea este valor (mayor discriminación), mayor será la distinción para separar los estudiantes en niveles de conocimiento diferentes. Aunque teóricamente se define en el rango $(-\infty, \infty)$, valores negativos no se tienen en cuenta, ya que indicaría una curva decreciente. Los ítems que tienen valores negativos tienen algún tipo de error y son descartados del test, siendo necesaria su revisión. Usualmente no se obtienen valores mayores que 2, por lo que el rango normal en el que se sitúa este valor es $(0, 2)$. El efecto de este valor en la curva se puede ver en la figura 3.1. Aquí se puede observar que un valor más pequeño de $a_i$ hace que la curva sea menos pendiente, mientras que valores altos hacen que el cambio de la probabilidad de no saber un ítem a saberla sea mucho más abrupto.

- $b_i$ o índice de dificultad del ítem, que coincide con el valor de $\theta$ para el que la probabilidad de responder correctamente es la misma que la de hacerlo incorrectamente, descontando el parámetro $c_i$. Valores mayores de este parámetro requieren un conocimiento mayor del estudiante para tener la misma probabilidad de responder incorrecta y correctamente un ítem. Cuando los valores de habilidad de

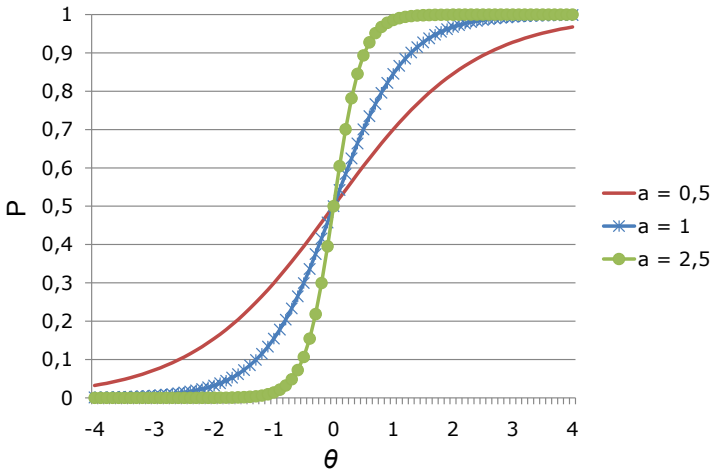



Figura 3.1: Parámetro $a_i$ (discriminación del ítem).

un grupo son normalizados a una media 0 y varianza 1, este parámetro toma valores en el intervalo $(-2, 2)$. Esto quiere decir que valores más cercanos a 2 corresponden a ítems más difíciles que aquellos que tienen un valor más cercano a $-2$. En la figura 3.2 se puede ver que el efecto que tiene este parámetro en la curva es el desplazamiento horizontal de la misma hacia la izquierda (ítems más fáciles) o hacia la derecha (ítems más difíciles).

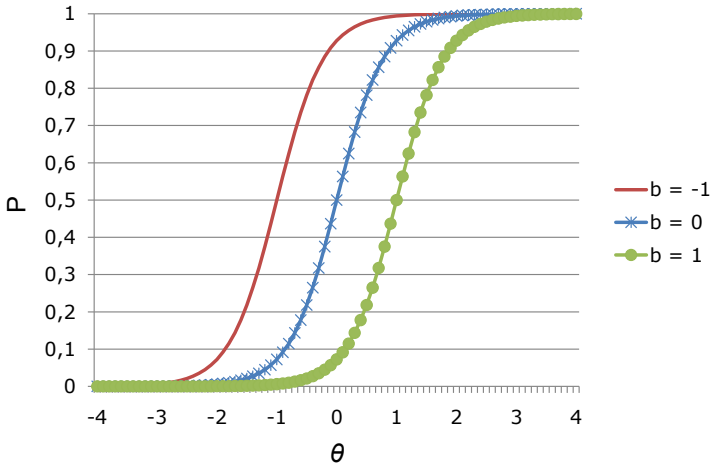



Figura 3.2: Parámetro $b_i$ (dificultad del ítem).

- $c_i$ o índice de adivinanza o azar del ítem. Representa la probabilidad de que un alumno sin conocimiento responda correctamente al ítem. Este parámetro intenta modelar el efecto del azar en la respuesta del estudiante, pero, tal y como se menciona en (Hambleton et al., 1991), los valores que toma suelen ser más pequeños que la probabilidad real de acertar si se responde aleatoriamente. Gráficamente, un valor mayor del parámetro influye en una probabilidad mayor de la curva cuando $\theta$ tiende a cero (figura 3.3).

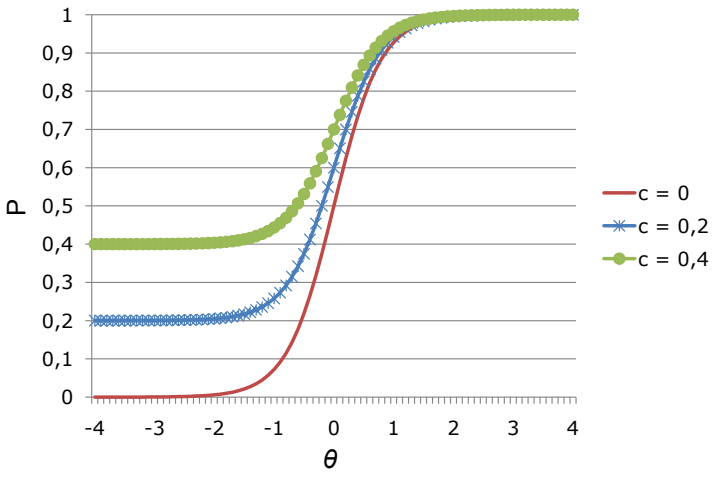



Figura 3.3: Parámetro $c_i$ (adivinanza del ítem).

Los **modelos normales** fueron los primeros usados y modelaban la respuesta a un ítem bajo el supuesto de que los errores estaban distribuidos de forma normal. Estos modelos se introdujeron en la Psicometría a partir de la rama de la Psicofísica, donde ya contaban con una extensa utilización. Dependiendo del número de parámetros, se distingue entre: modelo normal de 1 parámetro, o los denominados modelos ojivales (varias dimensiones) normales de dos o tres parámetros (en inglés *(2/3)-parameters normal ogive models*). En general, estos modelos establecen que, para un estudiante $j$ y un ítem $i$, la respuesta observada se expresa en función de las variables latentes reflejadas mediante la siguiente ecuación, obtenida de (Rao y Sinharay, 2007):

$$Y_{ij} = c_i + (1 - c_i)(a_i(\theta_j - b_i) + \epsilon_{ij}) \tag{3.5}$$

En esta ecuación, el error de medida $\epsilon_{ij}$ sigue una distribución normal de media 0 y varianza 1. Utilizando esta característica, la probabilidad de responder correctamente un ítem ($P_i(\theta)$) se expresa en función de la distribución normal, $\Phi(y_i)$ aplicando la ecuación 3.6 (Reckase, 2009):

$$P_i(\theta) = c_i + (1 - c_i)\Phi(y_i) = c_i + (1 - c_i)\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{a_i(\theta - b_i)} e^{-\frac{z^2}{2}}dz \tag{3.6}$$

Como se observa en la ecuación, ésta queda definida a partir de los tres parámetros típicos. Igualmente, está la función normal de dos parámetros que sería un caso particular de la fórmula anterior, considerando $c_i = 0$; y la función normal de un parámetro, que sería la misma que la anterior pero con los parámetros $a_i = 1; c_i = 0$.

Los **modelos logísticos** tienen su origen en una serie de informes técnicos de Birnbaum (1957a,b,c) en los que se introducía la distribución logística para aproximar la curva producida por una distribución normal. Esta idea estaba motivada por el trabajo de Haley (1952) en el que se demostraba que la curva producida por la función logística no difería en términos absolutos más de 0,01 para todos los valores de habilidad, $\theta$, si se utilizaba una constante multiplicativa $D$. El uso de esta distribución simplifica el cálculo computacional requerido en la estimación de los parámetros y, además, proporciona una función explícita para los ítems y los parámetros de habilidad que mantiene la

interpretación sobre los parámetros de los modelos normales. Es por estas ventajas por lo que este tipo de modelos son los más populares y usados hoy en día (Muñiz, 2010).

Teniendo en cuenta el número de parámetros se puede diferenciar entre modelo logístico de un parámetro; de dos parámetros; y de tres parámetros, todos definidos por Birnbaum (1968). Normalmente, para referirse a uno u otro modelo se utilizan los términos 3PL, 2PL, o 1PL, dependiendo del número de parámetros, terminología que proviene del inglés *n-Parameter Logistic*. De forma general, la CCI logística de un ítem $i$ se expresa mediante la fórmula de la ecuación 3.7.

$$P_i(\theta) = P_i(u|\theta) = c_i + (1 - c_i)\frac{1}{1 + e^{-Da_i(\theta - b_i)}} \tag{3.7}$$

Aunque la ecuación se corresponde a la función 3PL, ésta puede ser particularizada para la función 2PL y 1PL considerando que: en el modelo 2PL la adivinanza ($c_i$) toma el valor 0; y que el modelo 1PL es equivalente al 2PL pero con el índice de discriminación fijado al valor constante 1. En la ecuación, $D_i$ es la constante numérica que minimiza la diferencia máxima entre la función normal y la logística. El valor que debe tomar esta constante para que ambos modelos sean lo más parecido posible es 1,702 (Camilli, 1994).

Muchos autores denominan al modelo 1PL modelo de Rasch (1960). Sin embargo, hay que destacar que, si bien es cierto que, en la práctica ambos modelos son equivalentes, son diferentes por varios motivos (van der Linden y Hambleton, 1996). Cuando los modelos logísticos se desarrollaron en América, George Rasch ya había introducido su modelo en Dinamarca de manera independiente y bajo una motivación diferente. La diferencia principal entre el modelo de Rasch y su equivalente logístico radica en que el primero forma una escala común entre la persona y el ítem, definiendo parámetros para ambos, mientras que los modelos logísticos establecen parámetros para la población (deben seguir una distribución normal).

### Modelos no paramétricos

Los modelos no paramétricos, en inglés (*non-parametric models*) son una familia de modelos que se usan normalmente para tests de aptitud o personalidad y definen la curva de los ítems sin parámetros, lo que supone utilizar el conjunto de valores que representan la probabilidad para cada valor del rasgo latente. Como consecuencia, los modelos no paramétricos se corresponden únicamente a datos estadísticos obtenidos a partir de la calibración de las curvas con datos reales. Estos modelos se basan en un conjunto mínimo de supuestos que deben cumplirse para obtener mediciones válidas de las personas y los ítems (Sijtsma y Molenaar, 2002). De esta forma, además de los dos supuestos básicos de independencia local y unidimensionalidad, ya explicados, se suman dos nuevos supuestos sobre las propiedades de las CCI:

- *Monotonicidad de las CCI*: Los valores de la curva deben definir una curva monótona creciente en $\theta$.

- *No intersección de las CCI*: No debe existir intersección alguna entre los valores de las diferentes curvas asociadas a cualquier par de ítems. Esta característica establece una relación de orden total sobre las curvas, siendo posible ordenar los ítems correspondientes en función de los valores de la curva.

Este tipo de modelos surgen para intentar dar mayor flexibilidad a los ajustes que no se consiguen con los modelos paramétricos, los cuales suelen fallar a menudo (Ramsay, 1991; Douglas, 1997; Douglas y Cohen, 2001). Es por ello que estos modelos buscan ser de utilidad allí donde los paramétricos no llegan. Por ejemplo, si a priori se sabe la dimensión del rasgo latente, se puede decidir entre usar un modelo paramétrico unidimensional o multidimensional, mientras que los no paramétricos exploran los datos para adaptarse a la dimensión (Sijtsma y Molenaar, 2002). Junker y Sijtsma (2001); Molenaar (2001) mencionan tres características de utilidad que deberían guiar los avances de los modelos no paramétricos: 1) proporcionar una mejor comprensión de los elementos fundamentales y comunes en los modelos de la TRI; 2) ofrecer mayor flexibilidad en entornos en los que los modelos paramétricos tienen un ajuste pobre; y 3) proporcionar una serie de mecanismos fáciles de usar con un número de personas y de ítems mucho menor que son usados en los tests a gran escala.

Estudios como el de Junker y Ellis (1997) contribuyen a entender mejor qué son realmente las variables latentes en modelos psicométricos. En relación con la segunda característica deseable, mencionada en el párrafo anterior, Holland y Rosenbaum (1986) muestran que es posible usar cumplir el supuesto de unidimensionalidad sin usar un modelo paramétrico, lo cual puede ser útil si no hay un buen ajuste y esto puede deberse a la forma del modelo paramétrico en lugar de al incumplimiento de los supuestos de la TRI; aunque, como cita Junker (2011), la aplicación de los modelos no paramétricos se hace prohibitivamente lenta conforme el tamaño del test incrementa.

Dentro de la TRI hay dos tipos de modelos no paramétricos que pueden destacarse (Sijtsma y Molenaar, 2002): 1) El *Modelo de Homogeneidad Monótona* (en inglés, *Monotone Homogeneity Model*), que es un modelo para ordenar personas según un cierto criterio medido en un test. Este modelo sólo utiliza los supuestos básicos de la TRI y el de monotonicidad, dejando fuera el de no intersección de las CCI. El modelo se basa en que la ordenación que se obtendrá va a coincidir con la obtenida si se considera la puntuación verdadera de las personas. 2) El *Modelo de Monotonicidad Doble* (en inglés, *Double Monotonicity Model*), el cual es un caso particular del anterior en el que sí se considera el supuesto de no intersección de las CCI. Estos dos modelos se corresponden a ítems dicotómicos, aunque también tienen su equivalente extensión para modelos politómicos descrita en (Sijtsma y Molenaar, 2002).

#### 3.2.1.4. Taxonomía de Thissen y Steinberg

Adicionalmente a los criterios anteriores para clasificar los modelos de la TRI, cabe destacar una de las taxonomías de clasificación más conocidas: la propuesta por Thissen y Steinberg (1986). En ésta se utiliza como criterio de clasificación principal las restricciones de los parámetros asociados a cada modelo. El problema que presenta esta taxonomía es que sólo contempla una clasificación de modelos unidimensionales y paramétricos. Por ello es extendida posteriormente en (Hemker et al., 1997) para incluir modelos paramétricos, donde las relaciones entre los diferentes modelos son descritas mediante un diagrama de Venn que se basa en relaciones de orden estocástico. Algunos de los modelos que se mencionarán no han sido explicados anteriormente, por lo que, si se desea conocer más información de los mismos, se puede consultar (Thissen y Steinberg, 1986). La taxonomía completa estaría formada por los siguientes grupos de modelos:

- *Modelos binarios* (en inglés *Binary models*): Son los modelos que corresponden

a los mencionados anteriormente como dicotómicos ya que la respuesta sólo es clasificada como correcta o incorrecta. Los modelos binarios son los primeros que surgen y sirven como base para definir el resto de modelos más complejos. Dentro de este apartado se encuentran los ya mencionados *modelos normales ogivales*; los *modelos logísticos* de Rasch (1960); Birnbaum (1968); y los *modelos basados en esplines* (no mencionados).

- *Modelos diferenciales* (en inglés *Difference models*): Su nombre viene de usar la diferencia de probabilidades asociada a la CCI para calcular la probabilidad de que un ítem tenga una puntuación concreta. En este grupo entrarían: el mencionado *modelo general de respuesta graduada* de Samejima (2010), definido a partir de los modelos binarios normales y logísticos; y el *modelo de escala de clasificación de Muraki* (no mencionado anteriormente).

- *Modelos de división por el total* (en inglés *Divide-by-total models*): Éstos son modelos politómicos en los que las curvas características de cada respuesta se representan mediante un numerador, asociado a la respuesta, y un denominador que es igual a la suma total de los numeradores de todas las respuestas posibles (de ahí el nombre de los modelos). Dentro de esta categoría se encuentran el *modelo de respuesta nominal* (Thissen et al., 2010); el *modelo de crédito parcial* (Masters, 2010); el de *crédito parcial generalizado* (Muraki, 1982); y uno no mencionado anteriormente, el *modelo de escala de clasificación* (en inglés *Rating Scale Model*).

- *Modelos con parte izquierda añadida* (en inglés *Lef-side added models*): En esta categoría entran los modelos que introducen el factor de adivinanza, considerando la probabilidad de que un alumno sin conocimiento que responda al azar, lo haga correctamente. Dentro de estos modelos se encuentra el modelo 3PL de Birnbaum (1968).

- Modelos con parte izquierda añadida y división por el total (en inglés *Lef-side added and divided-by-total*): Esta categoría hace referencia a modelos que combinan las técnicas asociadas a las dos categorías. Los modelos que se encuentran aquí son: *Modelo de opción múltiple*, una versión de Samejima y otra de Thissen y Steinberg; y el denominado *modelo 6*.

- Modelos no paramétricos: Éstos son introducidos en la extensión propuesta por Hemker et al. (1997) y son extensiones de dos de los incluidos en la taxonomía original. Concretamente, son el *modelo no paramétrico de crédito parcial*, y el *modelo no paramétrico de respuesta graduada*. Ambos modelos imponen los dos supuestos básicos de la TRI junto con el de monotonicidad de las CCI asociado a los modelos no paramétricos, pero difieren en la definición de la CCI de acuerdo a como se realiza en la categoría en la que se agrupa la correspondiente versión no paramétrica.

Además de los modelos mencionados en esta sección, que son los más importantes, existen gran cantidad de modelos que no se han mencionado. Para un mayor detalle acerca de nuevos modelos y otros que todavía no están todavía tan arraigados como los mencionados, se recomienda consultar (Rao y Sinharay, 2007; DeMars, 2010; Nering y Ostini, 2010).

### 3.2.2. Fiabilidad de la TRI

Una de las mayores contribuciones de la TRI es la extensión del concepto de fiabilidad. En los modelos clásicos, la fiabilidad se refiere a la precisión de la medida, la cual es medida usando un único índice que se define de diversas formas, como se pudo ver en el apartado 3.1.1. Mientras que la TCT considera una fiabilidad uniforme para todos los individuos, la TRI establece que esta precisión no es uniforme en el rango de puntuaciones del test. Por ejemplo, la puntuación en los extremos del rango que puede tomar el rasgo latente suele contener mayor error que la puntuación más cercana al medio del rango.

La TRI introduce el concepto de información del ítem y del test para la medición de la fiabilidad, los cuales son conceptos clave para el cálculo del error estándar de medición, como se podrá ver a continuación. Para determinar la información asociada a un ítem se emplea su *Función de Información del Ítem* (FII) (Hambleton et al., 1991), la cual se presenta en la definición 3.2. Su equivalente para el test, la *Función de Información del Test* (FIT), se expone en la definición 3.3.

**Definición 3.2** (Función de Información de un Ítem)**.** *La FII es una función que refleja la cantidad de información que un ítem proporciona respecto de cada valor del rango en el que se localiza un rasgo latente. De forma general la FII, $I_i(\theta)$, asociada a un ítem i viene definida por la expresión de la ecuación 3.8. En esta ecuación $P_i'(\theta)$ es la derivada de la CCI y $Q_i(\theta)$ es la probabilidad responder incorrectamente, es decir, la probabilidad complementaria de la CCI, que equivale a $1 - P_i(\theta)$.*

$$I_i(\theta) = \frac{[P_i'(\theta)]^2}{P_i(\theta)Q_i(\theta)} \tag{3.8}$$

Normalmente, de acuerdo al modelo de la TRI utilizado, se utilizan expresiones que son simplificadas tras aplicar la fórmula asociada al modelo en la ecuación 3.8. Así pues, en el caso particular del modelo 3PL, tras realizar los cálculos, la FII se simplificaría a la ecuación siguiente (Hambleton et al., 1991):

$$I_i(\theta) = \frac{2{,}89a_i^2(1 - c_i)}{[c_i + \exp^{1{,}7a_i(\theta - b_i)}][1 + \exp^{-1{,}7a_i(\theta - b_i)}]^2} \tag{3.9}$$

La información que proporciona la FII de un ítem depende en gran medida de su factor de discriminación, produciendo un valor mayor de información si el ítem tiene mayor discriminación. La localización del punto de mayor información está altamente relacionada con la dificultad del ítem. Según Birnbaum (1968), y citado por Hambleton et al. (1991), este punto es el que equivale a la ecuación 3.10. Donde, si el factor de adivinanza es pequeño, este punto coincide con la dificultad $b_i$ del ítem. En general, si $c_i > 0$, el punto de valor máximo de la FII suele ser ligeramente mayor que su dificultad.

$$Max(I_i(\theta)) = b_i + \frac{1}{Da_i} ln[0{,}5(\sqrt{1 + 8c_i})] \tag{3.10}$$

**Definición 3.3** (Función de Información de un Test)**.** *Dado un test concreto, su FIT es una función que muestra la cantidad de información que éste proporciona en relación con los valores del rango que toma un rasgo latente. Esta función, $I(\theta)$, se calcula a partir de la FII de cada una de los n ítems que componen el test mediante la ecuación 3.11.*

$$I(\theta) = \sum_{i=1}^{n} I_i(\theta) \tag{3.11}$$

La cantidad de información en un test, en $\theta$ está inversamente relacionada con la precisión con la que el rasgo latente es estimado en ese punto (Hambleton et al., 1991). De esta forma, para estimar el error estándar de la medición, $EEM(\hat{\theta})$, se utiliza el estimador de la ecuación 3.12. El valor obtenido puede ser utilizado para el mismo fin que el EEM en la TCT.

$$EEM(\hat{\theta}) = \frac{1}{\sqrt{I(\theta)}} \tag{3.12}$$

### 3.2.3. Ventajas e inconvenientes de la TRI sobre la TCT

La TCT y la TRI, aunque tienen características similares y persiguen un objetivo común, difieren notablemente en la formulación de sus modelos. Las diferencias principales se han resumido en la tabla 3.1, la cual se ha adaptado a partir de los trabajos de Hambleton y Jones (1993); Muñiz (2010).

| Aspecto | TCT | TRI |
|---|---|---|
| Modelo | Lineal | No Lineal |
| Supuestos | Débiles (fáciles de cumplir) | Estrictos (difíciles de cumplir) |
| Elemento principal | Test | Ítem |
| Relación ítem-test | Sin especificar | CCI |
| Escala de puntuación | Entre 0 y la máxima del test | Entre $-\infty$ y $\infty$ |
| Invariancia de las mediciones | No | Si |
| Descripción de los ítems | Índices de discriminación y dificultad | Parámetros o valores (modelos paramétricos / no paramétricos) |
| Tamaño muestral | Entre 200-500 | Como norma, más de 500, aunque depende del modelo |

Tabla 3.1: Diferencias entre la TRI y la TCT.

La primera diferencia mencionada es la forma del modelo: el modelo lineal de la TCT es mucho más simple, siendo más fácil su manejo, su aplicación, y la estimación de los parámetros. No obstante, esto también se convierte en una forma más limitada de modelar el conocimiento. La principal ventaja de la TCT frente a la TRI es la facilidad con la que los datos pueden cumplir los supuestos requeridos para su aplicación. En la TRI es más difícil que el modelo se ajuste a los datos, siendo también requerida una población mayor para poder realizar las estimaciones. Aunque este tamaño varía dependiendo del modelo empleado, se suele requerir más de 500 individuos (Hambleton y Jones, 1993).

La principal ventaja de la TRI proviene del elemento hacia el que está enfocado la teoría y que supone una de las diferencias principales. Mientras que en la TCT se utiliza el test, en la TRI el elemento central es el ítem. El estudio de las propiedades individuales de los ítems, en lugar del test, hace que el modelo sea invariante, en el sentido de que es posible usar las CCI para estimar el conocimiento en poblaciones

diferentes de las que se obtienen las estimaciones de las CCI. Esta invariancia se encuentra también en la evaluación del individuo respecto de la dificultad que posea el test, la cual, en la TCT, está condicionada.

Otra de las ventajas de la TRI, es el uso de una escala común para situar el rasgo latente del estudiante y para describir las propiedades de los ítems mediante la CCI. Esta escala permite relacionar directamente los ítems y los niveles del rasgo latente de un individuo, lo cual es la base para proporcionar adaptación de los ítems al conocimiento del estudiante. Por último, en la TRI, la determinación de la fiabilidad no requiere de tests paralelos estrictos, siendo también independiente de la población donde se aplica. Además, el hecho de que la FIT esté definida de esta forma permite ser calculada a partir del subconjunto de ítems que participan en el test. Esto en la TCT no es posible, siendo necesario el conjunto completo de ítems.

## 3.3. Tests adaptativos informatizados

Un *Test Adaptativo Informatizado* (TAI) (en inglés *Computerized Adaptive Test*) es una prueba, construida para fines de evaluación psicológica o educativa, cuyos ítems se presentan y responden mediante un ordenador, siendo su característica fundamental que se va adaptando al nivel de competencia progresivo que va manifestando la persona (Olea, 2002). De esta forma, si un individuo responde correctamente un ítem, recibirá otro más complejo, mientras que una respuesta incorrecta hará que sea presentado un ítem más fácil.

Mediante la adaptación que se realiza a partir de la evaluación, se presenta una serie de ítems guiados por las necesidades de cada individuo. Como consecuencia, un TAI es más eficiente que los tests convencionales de ítems fijos, proporcionando una medición más precisa para la misma longitud del test, o tests más cortos para la misma precisión (Ponsoda, 2000). Esto es así porque los tests convencionales tienen el problema de que, al ir dirigidos a un grupo, deben cubrir el mayor rango de habilidad posible, lo que supone presentar muchos ítems que no son apropiados al nivel de los individuos dentro del grupo, sobre todo aquellos con un nivel de habilidad situada en los extremos (o muy alto o muy bajo) (Wainer, 2000).

A pesar de las ventajas inherentes al uso de TAC y de que los TAI proporcionan estimaciones más eficientes que los simples TAC, como también se demuestra en los trabajos de Conejo et al. (2000); van der Linden y Glas (2000), éstos también presentan inconvenientes. El primer problema que se plantea en los TAI proviene de la necesidad de tener un banco de ítems bien calibrado, lo que implica que éstos sean presentados con anterioridad a una muestra poblacional lo suficientemente amplia para que las estimaciones sean precisas (Wainer y Mislevy, 2000). Otro problema se encuentra en la seguridad del test, en el sentido de que los alumnos pueden memorizar las respuestas y compartirlas con futuros examinados, lo cual requiere de un banco de ítems lo suficientemente grande y de mecanismos de control de exposición, para evitar que haya ítems frecuentemente repetidos.

La idea subyacente a los TAI no es nueva: ya en los primeros tests de Binet et al. (1913) se clasificaban los ítems de acuerdo a la edad mental, se utilizaba un conocimiento de partida estimado por un examinador, y se adaptaba el siguiente ítem para estimular al individuo hasta que se pudiese determinar con certeza la edad correcta. Aunque los primeros desarrollos relacionados con los TAI surgen en los 70, no es hasta

los 90, con el uso de la tecnología para la realización de los TAC y con el desarrollo de la TRI, cuando este tipo de tests se hacen verdaderamente operativos (Ponsoda, 2000).

En la actualidad, el número de personas que son evaluados usando los TAI en estos exámenes crece cada año (Wainer, 2000) y tiene especial importancia y arraigo en Estados Unidos, donde este tipo de tests son utilizados como herramienta de evaluación en exámenes tan importantes como: el *Graduate Management Admissions Test* (GMAT) para la admisión a la universidad; el conocido *Test Of English as a Foreign Language* (TOEFL) como acreditación sobre lengua inglesa; el *Graduate Record Exam* (GRE) para el acceso a escuelas de postgrado; y el *the Armed Services Vocational Aptitude Battery* (ASVAB) para la admisión de personal en la armada estadounidense.

La aplicación de un TAI consiste en un proceso iterativo, cuyos pasos típicos son resumidos en la figura 3.4, adaptada a partir de (Barrada, 2012). En primer lugar, se utiliza un procedimiento de arranque que consiste en utilizar algún criterio para elegir el nivel del conocimiento de partida. A continuación, como todavía no se ha realizado ningún ítem, el criterio de parada no se puede comprobar, por lo que se selecciona un ítem mediante algún criterio que tenga en cuenta el nivel estimado actual del individuo. Acto seguido, se presenta el ítem y el individuo responde a éste. Seguidamente, se actualiza el nivel del conocimiento en base a la respuesta dada. Una vez realizados los cálculos asociados, se comprueba si el criterio de parada se cumple, finalizando el TAI en tal caso o volviendo a seleccionar un nuevo ítem, en caso contrario.

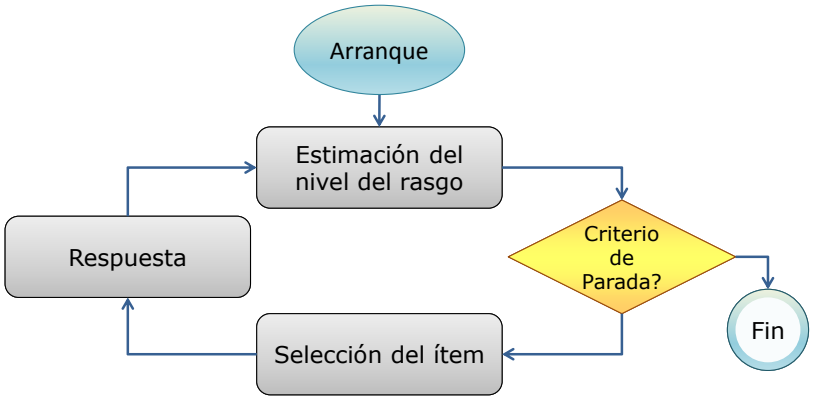



Figura 3.4: Flujo de ejecución de un TAI.

En este proceso es necesario disponer de varios elementos fundamentales para realizar la aplicación de los TAI (Guzmán, 2005):

- Un *Modelo de respuesta del ítem* que describe cómo el sujeto respondería al ítem según su nivel de conocimiento. Cuando se llevan a cabo mediciones del nivel de conocimiento cabe esperar que el resultado obtenido no dependa del instrumento utilizado, es decir, la medida sea invariante con respecto al tipo de test y al sujeto al que se le aplica el test.
- Un *Banco de Ítems* en donde se almacenan los ítems y las estimaciones correspondientes al modelo de respuesta asociado. Este elemento es uno de los más importantes en la creación de un TAI pues condiciona la calidad de la adaptación posterior. Un banco de ítems eficiente debe tener, entre otras muchas características deseables (Reckase, 2010): el suficiente número y variedad para cubrir las

distintas áreas a evaluar; cada ítem debería ser independiente; y, en conjunto, deberían cubrir el mayor rango posible de valores del rasgo latente.

- Un procedimiento para la *estimación del nivel inicial del conocimiento*, el cual puede reducir sensiblemente la longitud del test. Los tres criterios básicos para esta tarea son (Barrada, 2012): a) asignar el promedio poblacional; b) asignar un valor por debajo del promedio para aumentar la probabilidad de acierto e incrementar la motivación del individuo; o c) usar información sobre los examinados que permita predecir el rendimiento en el test, tales como puntuaciones en otro test o elemento educativo, o estimar un nivel de conocimiento inicial diferente por examinado (van der Linden, 1999).

- Un *criterio de selección de ítems* que permita determinara cuál de los existentes en el banco de ítems maximiza una función de valoración. Esta función de valoración tendrá en cuenta el conocimiento estimado del sujeto y unos objetivos particulares del test entre los que mencionan: generar una estimación precisa; garantizar el ajuste a las especificaciones de contenido del test, facilitar el mantenimiento del banco de ítems, o limitar la probabilidad y efectos de una filtración de ítems (Barrada, 2012).

- Un *criterio de finalización* del TAI que, de acuerdo a los objetivos del test, determine cuándo es conveniente finalizar. Para ello se pueden usar criterios como: un número fijo o máximo de ítems; un umbral mínimo sobre la precisión de la estimación; o la detección de una situación en la que cualquier ítem de los restantes no va a aumentar la precisión.

### 3.3.1. Calibración de los ítems en los TAI

Previamente a la aplicación de la TRI en un TAI es necesario obtener la curva CCI correspondiente a cada ítem. A este proceso de estimación se denomina *calibración*. La calibración consiste en estimar las características de los ítems a partir de una muestra poblacional. Es decir, los parámetros correspondientes a cada ítem (si el modelo es paramétrico), o los valores de la curva (en modelos no paramétricos). Esta estimación es similar al problema de ajuste de regresión, con la diferencia de que el modelo asociado no es lineal y que la variable usada para medir el conocimiento no es observable.

#### 3.3.1.1. Anclaje y equiparación

Puesto que es necesario utilizar datos de una muestra de examinados, para poder realizar el proceso de calibración es necesario realizar un proceso previo de recolección de datos. Debido al elevado número de ítems de los bancos, muchas veces es inviable presentarlos todos a un mismo grupo de estudiantes puesto que los resultados estarían influenciados por elementos como la fatiga. Es por ello que suelen dividirse los bancos en grupos de ítems y se presentan a diferentes grupos de individuos. Para garantizar que las estimaciones estén en la misma escala es necesario utilizar un proceso de nominado *proceso de anclaje y equiparación* que está compuesto de dos fases (Olea, 2002):

- Anclaje (en inglés *linking*): Típicamente, esta fase consiste en dividir el banco en varios tests que tienen en común un conjunto de ítems, el cual se denomina *test de anclaje* y a sus componentes, *ítems de anclaje*. El test de anclaje debe ser

representativo del banco completo y debería suponer sobre el 20 % del número de ítems que tienen los diferentes tests a equiparar. El proceso de anclaje debe planificarse antes de realizar la recolección inicial de los datos. Para ello, es necesario también considerar el tamaño de la muestra, el cual dependerá del tamaño de los ítems de cada bloque de tests y del modelo de la TRI a aplicar.

- Equiparación (en inglés *equating*): Consiste en poner en la misma escala las estimaciones obtenidas tras aplicar la calibración a los diferentes tests por separado. Puesto que los tests de anclaje comparten los ítems, las estimaciones de estos ítems de anclaje deben estar linealmente relacionadas. El problema se reduce a determinar los coeficientes de la relación lineal entre un par de tests de anclaje y aplicar la transformación a las estimaciones del resto de los ítems no anclados. Esto pondría un par de test en la misma es cala, repitiéndose el proceso con el resto de tests no equiparados.

#### 3.3.1.2. Métodos de calibración

Tras la obtención de los resultados presentados en los tests no adaptativos que contienen los conjuntos de ítems de anclaje, y previamente a la equiparación de las estimaciones, se realiza la calibración en sí de las CCI. Los métodos que se utilizan para esta tarea están basados en una función denominada *función de verosimilitud*, la cual se define a continuación.

**Definición 3.4** (Función de verosimilitud)**.** *La función de verosimilitud, $L(u|\theta)$, representa la distribución del conocimiento de un sujeto dado un patrón de respuestas que éste ha dado en un test ($P(\theta|u)$). Esta probabilidad se calcula, bajo el supuesto de independencia local, combinando las respuestas dadas por el estudiante y la probabilidad de dar esa respuesta, reflejada en la CCI, mediante la siguiente ecuación:*

$$L(u|\theta) = P(\theta|u) = \prod_{i=1}^{n} P_i(u_i = 1|\theta)^{u_i}(1 - P_i(u_i = 1|\theta))^{1-u_i} \tag{3.13}$$

*En la ecuación, $u = u_1, u_2, .., u_n$ representa el patrón de respuestas dada en los n ítems realizados; $u_i = 1$ indica que se ha respondido correctamente al ítem i realizado; y $P_i(u_i = 1|\theta)$ es la CCI del ítem.*

El proceso de calibración a partir de los datos obtenidos se puede llevar a cabo de diversas formas (Baker y Kim, 2004). A continuación se mencionan los cuatro métodos más conocidos y comúnmente utilizados:

- *Máxima Verosimilitud Conjunta* (en inglés *Joint Maximum Likelihood*): Este método, propuesto por Birnbaum (1968), busca los valores que hacen más probable la obtención de los datos empíricos a partir del modelo y se basa en estimar conjuntamente los parámetros de los ítems y el conocimiento de los sujetos. La estimación según este método es un proceso iterativo en dos fases en el que primeramente se fija un valor para el conocimiento de los sujetos y a partir de éste, usando la derivada de la función de verosimilitud anterior respecto de cada parámetro, se estiman los parámetros de los ítems. Con la estimación de los parámetros, se estima el nivel de conocimiento de los sujetos. De nuevo, se vuelven a repetir estas dos fases y el proceso continúa hasta que no hay una actualización

significativa en los parámetros. Para realizar la aproximación es necesario el uso de métodos numéricos. Algunos de los más conocidos en esta tarea (Jennrich y Sampson, 1976; Ehlers, 2002) son el método de Newton-Raphson; la *maximización de la esperanza*, más conocido por E-M (del inglés *expectation-maximization*); y el método de puntuación de Fisher.

Este método de calibración tiene la ventaja de que es computacionalmente simple, pero tiene varios problemas asociados. El primero es la inconsistencia de las estimaciones cuando el número de evaluados es muy grande, ya que éstas no convergerán a sus valores verdaderos. No sólo son inconsistentes, sino que también son sesgadas ya que se realizan conjuntamente con las estimaciones del conocimiento. No obstante, cuanto mayor es el tamaño del test, menor importancia toma el sesgo, pues los parámetros del sujeto pueden ser estimados con mayor precisión. Otro problema del método es que no puede ser aplicado cuando un alumno ha respondido correctamente o incorrectamente todos los ítems o cuando, para un ítem concreto, todos los alumnos lo han respondido correctamente o incorrectamente. Esto es así porque los cálculos en estos casos sitúan el valor del conocimiento y de la dificultad del ítem, en un límite de la función de verosimilitud.

- *Máxima Verosimilitud Condicional* (en inglés *Conditional Maximum Likelihood*): Propuesto por Andersen (1972), este método proporciona estimaciones consistentes y eficientes gracias a la factorización de los parámetros asociados al conocimiento fuera de las ecuaciones derivadas de la función de verosimilitud. Para poder realizar esta factorización son necesarios ciertos estadísticos que sólo están disponibles en el modelo 1PL. En este caso, el uso del número de respuestas correctas es un estadístico suficiente para estimar la dificultad de los ítems condicionado a este número de respuestas. De esta forma, la expresión normal de la verosimilitud ($L(u|\theta)$) se reemplaza por $L(u|r)$, donde $u$ es un vector con los patrones de respuestas de cada sujeto y $r$ es un vector con el número de respuestas correctas de cada sujeto. Puede ser demostrado que $L(u|r) = L(u|\theta)/L(r|\theta)$, que es independiente de $\theta$ porque los términos en el numerador y denominador se anulan entre sí.

  Aunque este método tiene la ventaja de que separa los valores del conocimiento y los parámetros de los ítems, tiene varias limitaciones. La primera, al igual que en el método anterior, es que no puede aplicarse cuando las respuestas son todas correctas o incorrectas. También, los alumnos que tengan el mismo número de respuestas correctas pero diferente patrón de respuestas tendrán la misma puntuación. Por último, presenta problemas a la hora de aplicarse a tests largos, patrones de respuestas sin evidencia, o ítems politómicos y sólo puede aplicarse a modelos 1PL.

- *Máxima Verosimilitud Marginal* (en inglés *Marginal Maximum Likelihood*): En lugar de usar la función de verosimilitud condicionada al conocimiento del estudiante, Bock y Lieberman (1970) propusieron utilizar una función de verosimilitud incondicional $L(u)$ que representa la probabilidad de observar el patrón $u$ en un sujeto de habilidad desconocida seleccionado aleatoriamente de una población. Si la habilidad de la población tiene una distribución descrita por una función de

densidad $g(\theta)$, entonces esta función viene representada por la siguiente ecuación:

$$L(u) = \int \sum_{-\infty}^{\infty} L(u|\theta)g(\theta)d\theta \tag{3.14}$$

Esta función depende sólo de los parámetros del ítem puesto que la habilidad $\theta$ ha sido integrada. Dado que esta integral suele requerir estimación mediante el proceso de cuadratura Gaussiana, tiene asociado un alto coste computacional. Es por ello que Bock y Aitkin (1981) reformularon este método introduciendo para la estimación el algoritmo E-M. Este algoritmo tiene dos etapas: en la primera, denominada "esperanza", se obtiene los valores esperados de las frecuencias en los puntos de cuadratura y se calculan las frecuencias de los sujetos que responden correctamente los ítems; en la segunda, denominada "maximización", estos valores esperados son la entrada de las ecuaciones de estimación para maximizar la verosimilitud. Las dos etapas se repiten hasta que las estimaciones convergen. Normalmente, se utiliza el método de Newton-Gauss para la resolución de las ecuaciones usadas en la maximización de la verosimilitud.

Este método presenta diversas ventajas sobre los anteriores. En primer lugar, puede ser aplicado eficientemente para tests de longitud considerable. Tampoco presenta la limitación de los otros métodos, permitiendo aplicarse si el alumno ha respondido todos los ítems correctamente o incorrectamente. No obstante, también tiene algunas limitaciones como el uso de una función de distribución, asumiéndose la normal si ésta no es conocida. El efecto de usar una normal en una población con otra distribución puede ser minimizado puesto que se puede estimar la distribución a partir de los datos. Otra limitación es que, aunque tiene estimaciones inconsistentes, mantiene el problema de las estimaciones sesgadas, lo que nos lleva al siguiente método.

- *Métodos bayesianos*: Estos métodos están basados en el teorema de Bayes, el cual especifica la relación entre la probabilidad condicional e incondicional de la ocurrencia de un evento. En base a este teorema, la función de verosimilitud se utiliza para estimar la probabilidad de $\theta$ condicionado a un patrón de respuestas $u$ como sigue:

$$P(\theta|u) = \frac{L(u|\theta)P(\theta)}{L(u)} \tag{3.15}$$

Esta probabilidad $P(\theta|u)$ se puede considerar la distribución de las estimaciones de la habilidad de acuerdo a las respuestas de los ítems, es decir, la función de densidad de la distribución a posteriori de la habilidad. $P(\theta)$ es la distribución de la habilidad a priori, o lo que es lo mismo, la función de densidad $g(\theta)$. $L(u)$ es la verosimilitud del patrón de respuestas, independiente de $\theta$ (Olea, 2002). Puesto que el denominador es un valor constante, la distribución a posteriori es proporcional al producto entre $L(u|\theta)$ y $P(\theta)$ y puede aproximarse mediante la ecuación 3.16. Si la distribución $g(\theta)$ fuera uniforme, el estimador bayesiano coincidiría con el de la máxima verosimilitud.

$$P(\theta|u) \propto L(u|\theta)P(\theta) \tag{3.16}$$

El procedimiento de estimación utiliza esta relación y la información de la distribución a priori, la cual es informativa si la varianza de la distribución es pequeña.

Si la varianza es grande, haría que la distribución a priori tuviera un menor impacto en la estimación de los parámetros de habilidad. En cambio, si la distribución es informativa, las estimaciones se agruparían sobre la media de la distribución, evitando valores de habilidad que diverjan considerablemente de esta. Esta característica ayuda a resolver el problema de las estimaciones sesgadas en los métodos anteriores y permite la estimación de patrones de respuesta donde todas o ninguna de las respuestas son correctas.

Este método establece dos estimadores del valor de $\theta$ en la probabilidad a posteriori. El primero es la moda, en cuyo caso, el método se denomina MAP (estimación bayesiana del *máximo a posteriori*). El segundo estimador es la media de la distribución y el método se denomina EAP (estimación bayesiana de la *esperanza a posteriori*).

Aunque estos métodos son desarrollados para modelos dicotómicos, también son extensibles a modelos de la TRI politómicos, con la diferencia de que, en lugar de usar la CCI, se utiliza la curva característica de las respuestas seleccionadas por el alumno. De esta forma, la ecuación 3.13 sobre la función de verosimilitud se puede reescribir como (Kleinbaum y Klein, 2010):

$$L(u|\theta) = P(\theta|u) = \prod_{i=1}^{n} \prod_{j=1}^{m} P_i(u_i = j|\theta)^{y_{ij}} \tag{3.17}$$

donde $m$ es el número máximo de categorías de los ítems; $P_i(u_i = j|\theta)$ es la curva característica del ítem $i$ asociada a la $j$-ésima categoría; y el valor $y_{ij}$ es 1, si la respuesta al ítem $i$ es igual a la categoría $j$, o 0 en otro caso.

Cuando la calibración se realiza sobre modelos no paramétricos el modelo es totalmente diferente, pues no hay parámetros. Esto implica que en la estimación de la curva, definida por los valores de probabilidad, tengan que usarse técnicas de regresión no paramétrica. Aunque existen muchos métodos, cada uno basado en técnicas diferentes, uno de los más conocidos, por su facilidad de uso y conveniencia computacional, es el método de *suavizado del núcleo* (en inglés *Kernel Smoothing*). Propuesto por primera vez por Ramsay (1991), este método utiliza técnicas de suavizado del núcleo cuyo objetivo es hacer que la estimación de una función tenga una curva suave en la que el ruido es atenuado mediante un parámetro de suavizado. El suavizado del núcleo se basa en promediar los valores alrededor de un punto y usar la media asociada como valor representativo para producir la curva de regresión. Estos modelos pueden ser aplicados para estimar tanto modelos dicotómicos, como politómicos. Puesto que no se han usado modelos no paramétricos en el trabajo de esta tesis, no se entra en mayor detalle, sugiriendo al lector interesado en ampliar información sobre éste y otros métodos de estimación no paramétrica consultar (Eubank, 1999; Molenaar, 2001; Härdle, 2004).

### 3.3.2. Uso de la TRI en la ejecución de los TAI

Como se ha mencionado al principio de esta sección, el desarrollo de la TRI y los CAT es fundamental para que los modelos teóricos desarrollados para TAI comiencen a ser aplicados. En esta aplicación, la TRI se torna como una base bien fundamentada sobre la que asentar las decisiones involucradas en la adaptación. Para ello, el rasgo latente estimado mediante la TRI es el conocimiento. Éste, unido con una representación de las características de los ítems, mediante la CCI, puede ser usado para la

toma de decisiones adaptativas. Otra aportación de la TRI a los TAI es el uso de una escala común que permite comparar ítems y tests, tanto si están diseñados en momentos distintos, como si se han calibrado en poblaciones diferentes (Wainer y Mislevy, 2000), siempre bajo el supuesto de que las CCI están bien calibradas. A continuación, se explica cuál es la aportación concreta de la TRI a cada fase de aplicación de un TAI tomando como base las etapas mencionadas en la figura 3.4.

#### 3.3.2.1. Estimación inicial del conocimiento

Cuando el alumno todavía no ha respondido a ningún ítem es importante tener alguna estimación inicial del conocimiento que permita definir cuál es el punto de partida del TAI. Esto cobra más importancia en tests más cortos, ya que, de acuerdo a van der Linden y Pashley (2010), en tests más largos (entre más de 20/30), el TAI puede recuperarse de una mala estimación inicial y generar todavía una estimación precisa del conocimiento.

En esta fase, la estimación de las CCI permite usar como punto de partida un ítem de dificultad media, si no se dispone de otra información acerca del conocimiento que el alumno pudiera tener inicialmente. En caso de que se dispusiera de este valor, la elección del primer ítem a mostrar sería como en el funcionamiento normal del mecanismo de selección.

#### 3.3.2.2. Estimación general del conocimiento

Una vez el alumno ha respondido a algún ítem, se puede utilizar la respuesta proporcionada junto con la correspondiente CCI, previamente calibrada, para estimar el conocimiento del alumno. Los métodos que se utilizan para esta estimación están íntimamente relacionados con los explicados en la sección 3.3.1 y son dos principalmente:

- El método de la *máxima verosimilitud* (MV) (en inglés *Maximum Likelihood*), es uno de los más utilizados y se basa en la función de verosimilitud para realizar la evaluación. A partir de ésta, puesto que es una función de densidad, el valor de $\theta$ donde se encuentra el máximo de la misma representa el punto donde es más probable esté el conocimiento del alumno. La determinación de este máximo se puede obtener aplicando la primera derivada de la función de densidad. Normalmente la expresión se puede modificar aplicando logaritmos con el fin de transformar las potencias en productos y los productos en sumas, reduciendo el coste computacional y obteniendo el mismo resultado. No obstante, para aproximar este valor se requiere aplicar alguno de los métodos de cálculo numérico mencionados como Newton-Raphson, E-M, o el método de puntuación de Fisher.

- Como ya se ha puntualizó en la sección 3.3.1, el problema que tienen los métodos basados en la función de verosimilitud es que no pueden aplicarse cuando el alumno ha respondido el mismo patrón de respuestas en todos los ítems. Para solucionar esto, se aplica alguno de los dos métodos bayesianos MAP o EAP explicados en esa misma sección. Aunque estos métodos resuelven el problema del de máxima verosimilitud, tienen otros inconvenientes, como la influencia de la distribución de conocimiento a priori en la evaluación, la cual afecta haciendo que dos alumnos con el mismo patrón de respuestas tengan estimaciones del conocimiento diferentes, si tienen una distribución inicial diferente.

Un caso particular de la evaluación es cuando ésta es final y se va a mostrar al sujeto. Entonces, la escala de $\theta$ suele transformarse a otra más entendible. Los dos mecanismos de transformación más populares son (van der Linden, 2006): el método *equipercentil*, que consiste en equiparar aquellas puntuaciones cuyos percentiles son iguales; y el uso de la *Curva Característica del Test* (CCT), la cual relaciona la escala de $\theta$ con la puntuación verdadera del test. Esta última está se determina mediante las CCI de los ítems en el test y se define como (Baker, 2001):

**Definición 3.5** (Curva Característica de un Test)**.** *La CCT es una función monótona creciente que establece la puntuación verdadera en base al total de ítems del test para cada valor de* $(\theta)$*. A partir de un test* $t$*, su curva* $P_t(\theta)$ *se calcula sumando las CCI asociadas a los ítems involucrados en el test mediante la ecuación 3.18. Si se representan en un gráfico los valores de la función, el eje de abscisas se corresponde a* $\theta$ *y se sitúa en la escala* $(-\infty, \infty)$*; mientras que el eje de ordenadas va entre 0 y el número máximo* $n$ *de ítems del test, lo cual representa la puntuación verdadera.*

$$P_t(\theta) = \sum_{i=1}^{n} P_i(\theta) \tag{3.18}$$

#### 3.3.2.3. Selección del siguiente ítem

Para la determinación del ítem más apropiado se busca maximizar una función de valoración aplicada a cada ítem. De esta forma, el ítem con un valor máximo en esta función es el más adecuado. La definición de la función de valoración puede hacerse de acuerdo a unas necesidades determinadas en el TAI. Los tres métodos más importantes en la determinación del siguiente ítem son los siguientes (van der Linden y Hambleton, 1996):

- *Criterio de máxima información*: Consiste en seleccionar aquel ítem que maximiza la función de información de la definición 3.2. Para ello, se utiliza la estimación actual del conocimiento en la fórmula de la FII asociada al modelo de la TRI que se esté usando. Esta fórmula se aplica sobre los ítems que no han sido presentados seleccionándose el que tenga mayor valor.

- *Método bayesiano de la máxima precisión esperada*: Este método fue propuesto por Owen (1975) y consiste en minimizar la esperanza de la varianza de la distribución del conocimiento a posteriori expresada en la ecuación 3.19. De esta forma se maximiza la precisión de la estimación.

$$E(\sigma^2(\theta|u_n, u_{n+1})) = \sum_{i=0}^{1} \sigma^2(\theta|u_n, u_{n+1}) \int P(u_{n+1} = i|\theta) g(\theta) d\theta \tag{3.19}$$

  Aquí, $E$ es la esperanza matemática; $\sigma^2$ la varianza; $u_n$ el patrón de respuestas que el alumno y ha dado; $u_{n+1}$ la respuesta que daría el alumno en el ítem al que se aplica la función de valoración; el índice $i$ denota sólo dos valores, es decir, se está aplicando a modelos dicotómicos; $P(u_{n+1} = i|\theta)$ es la CCI; y $g(\theta)$ es la distribución inicial del conocimiento.

- *Método basado en el nivel de dificultad*: Este método, propuesto también por Owen (1975), consiste en dar mayor valoración a los ítems con una dificultad más

cercana a la del conocimiento actual del alumno. De esta forma, la función de valoración consistiría en minimizar la diferencia entre el parámetro $b_i$ del ítem candidato y el valor actual de $\theta$. Owen demuestra que este tipo de selección es equivalente al bayesiano si todos los ítems tienen la misma dificultad. Aunque en este método se reducen los cálculos necesarios, la decisión del siguiente ítem se realiza sin tener en cuenta otra información distinta del parámetro de dificultad.

Los dos primeros métodos tienen el problema de que tienden a presentar los ítems más discriminativos, lo cual influye en la seguridad del test al existir mayor probabilidad de que los alumnos compartan información sobre éstos ítems. En (Barrada, 2012) se hace una buena recopilación de los diferentes métodos de control de la seguridad, entre los que se pueden destacar: control de la tasa máxima de exposición; uso de la tasa de solapamiento (proporción de ítems que un par de examinados comparten); utilización de bancos rotatorios (diversos bancos de ítems que se van cambiando); métodos estratificados, que buscan reducir la tasa de exposición de los ítems más populares; métodos estocásticos, y un largo etcétera. Además, si el test se realiza sobre varios temas, no se controla la proporción de cada uno, lo cual puede ser un inconveniente dependiendo del tema y de los objetivos del test. En estos casos se suele usar un método de balanceo como el propuesto por Guzmán (2005), que equilibra el contenido mediante un método no heurístico basado en la dispersión de la distribución del conocimiento, la cual se determina con la varianza.

Los mencionados en esta sección son sólo los métodos clásicos y más utilizados. Aunque la explicación se ha realizado sobre ítems dicotómicos, para ítems politómicos existen sus métodos equivalentes (Choi y Swartz, 2009), junto con muchas otras variantes basadas en la entropía, en la FII o la información de las opciones de respuesta (Guzmán, 2005). En la actualidad hay cada vez más métodos que, basándose de los anteriores, añaden o modifican elementos que sustentan la decisión del siguiente ítem. Por ejemplo, en (van der Linden y Pashley, 2010) se mencionan criterios de selección más modernos como: el de *maximización de la función de información global*; el criterio de *información de la verosimilitud ponderada*; el criterio *completamente bayesiano*; o el criterio de *selección bayesiana con información colateral*.

Otro tipo de tests adaptativos son los denominados *tests en la sombra*, propuestos por van der Linden (2010), los cuales seleccionan ítems basándose en un criterio que tiene en cuenta el efecto del ítem a un nivel más amplio. Para ello, por cada ítem disponible se construye un test adaptativo de acuerdo a la estimación del conocimiento y respetando las posibles restricciones del test como el máximo número de ítems. El ítem a seleccionar es aquel en el que su test "en la sombra" maximice el criterio de selección asociado. Como se puede observar, existen muchos criterios que pueden establecerse. El lector puede ampliar información consultando el trabajo de van der Linden y Pashley (2010) o el de Barrada (2012), donde se hace se hace una extensa recopilación de métodos de selección, clasificados por diversos objetivos del TAI.

#### 3.3.2.4. Criterio de finalización

En la determinación de cuándo el TAI debe terminar se pueden distinguir varios criterios básicos que son aplicables independientemente de la TRI como: el establecimiento de un número fijo de ítems, un número máximo o un límite temporal. El problema de estos métodos es que la evaluación tendrá una precisión diferente en cada estudiante.

La aportación principal de la TRI al criterio de finalización está asociada con el

cumplimiento de un mínimo de precisión, la cual se determina mediante la función de información, tal y como se explicó en el apartado 3.2.2. También influye en el criterio de parada en el que los ítems que quedan por mostrar no van a mejorar la precisión.

En la práctica, los criterios que tienen en cuenta la precisión se combinan con aquellos que limitan la duración o el número de elementos del test para hacer que éste no sea demasiado largo. Estos criterios son igualmente aplicables para modelos politómicos, usando para ello la función de información correspondiente.

## 3.4. Otras formas de evaluación

Hasta ahora los mecanismos de evaluación vistos son los que se suelen aplicar para realizar la evaluación formal, enmarcándose en entornos de tests y evaluación de conocimiento mediante ítems simples. Uno de los objetivos principales de esta tesis es la extensión de la evaluación a EIRP manteniendo la formalidad y objetividad de la evaluación. Dado que normalmente el resultado de la evaluación se suele utilizar para la toma de decisiones que pueden influir sobremanera en las vidas de las personas, es importante cuidar la objetividad y el uso de elementos bien fundamentados. Esto nos lleva a estudiar si existen elementos que permitan aplicar los mecanismos anteriores a EIRP o si existen otros mecanismos formales.

Hay que destacar que en el campo de los STI existen infinidad de sistemas, cada uno con su propio enfoque particular sobre cómo realizar el diagnóstico y cómo modelar al alumno. No obstante, el objetivo de esta tesis no es analizar sistemas particulares, sino utilizar metodologías, marcos de trabajo, o paradigmas que puedan aplicarse de forma genérica a múltiples dominios. En este sentido, este apartado trata los campos que están relacionados con esta tarea y se revisa uno de los enfoques genéricos más conocidos que tratan la evaluación del alumno en tareas complejas.

### 3.4.1. Otros campos de estudio relacionados

Uno de los campos en donde se ha encontrado similitud con la evaluación en tareas complejas es el de la *puntuación automática* (en inglés *automated scoring*). Sin embargo, la mayoría de trabajos encontrados en la literatura están más cerca de la Psicología aplicada que del uso en EIRP, ya que se centran normalmente a tareas de evaluación oral y de escritura de ensayos. Es por ello que los enfoques existentes plantean pautas que guíen a un humano a la hora de evaluar estas tareas. No obstante, también hay aplicaciones más cercanas a los STI en este campo, como se recoge en (Williamson et al., 2006). Algunas de ellas son *Architectural Registration Examination* (ARE), inspirado en las reglas razonamiento de los sistemas expertos; *NetPASS*, que utiliza redes bayesianas; o *IMMEX* que usa redes neuronales. Todos estos enfoques pueden ser consultados en el mencionado trabajo. La característica común a todos ellos es que siguen siendo soluciones específicas en dominios concretos, cada una con una forma concreta y un propósito diferente. Aunque la metodología específica puede ser extendida a otros dominios, no existen una serie de pautas establecidas para la construcción, desarrollo y aplicación de nuevos sistemas, como sucede con los paradigmas de construcción de EIRP tales como el MBR o los tutores cognitivos.

Otra de las áreas donde la evaluación se aplica en tareas complejas es la denominada como *diagnóstico cognitivo* (en inglés *cognitive diagnosis*). El proceso de diagnóstico cognitivo consiste en inferir el estado cognitivo de una persona a partir de

su estado (Ohlsson, 1986). Los modelos de este campo pueden ser analizados desde dos perspectivas: en una más cercana a la Psicología donde destaca la utilización de CAT para realizar la evaluación (Huebner, 2010). La relación entre cada ítem de un test y las habilidades cognitivas o atributos se representa en una matriz denominada *Q-matriz* (Tatsuoka, 2009), la cual es utilizada para clasificar al alumno en diversas clases latentes. Respecto a este campo no se entrará en mucho más detalle pues su cercanía al ámbito de la Psicología se escapa de los objetivos de esta tesis. La otra perspectiva, más cercana a la IA y de mayor interés para este trabajo, nos lleva a los dos paradigmas principales descritos en el capítulo anterior: el MBR y los tutores cognitivos.

En relación con el diagnóstico del alumno en STI, de forma general, puesto que el objetivo es que los alumnos aprendan, se centran en modelar el aprendizaje de una tarea o concepto, por lo que la evaluación queda en un segundo plano. Además de los dos paradigmas explicados en el capítulo anterior, otros enfoques relacionados con el diagnóstico en EIRP se recogen en el trabajo de Desmarais y Baker (2012). Aquí, se mencionan enfoques de diagnóstico probabilísticos mediante cadenas de Markov o redes bayesianas, entre otros.

### 3.4.2. Diseño basado en evidencias

La limitación de los tests para la evaluación de tareas complejas es algo de lo que ya es consciente la comunidad científica desde hace tiempo. En la tarea de resolver este problema y extender la evaluación a tareas más complejas, uno de las propuestas más conocidas es el marco de trabajo del *Diseño Basado en la Evidencia* (DBE) (en inglés *Evidence-Centered Design*) (Mislevy et al., 2003a,b; Mislevy y Riconscente, 2006). Este marco de trabajo es fruto de una serie de investigaciones iniciadas en 1997 en la institución Educational Testing Service, cuyo fin es establecer una serie de principios para el diseño, la producción y la administración de evaluación educativa en base a evidencias del conocimiento.

En lugar de definir una metodología de evaluación concreta, el DBE ofrece un marco de trabajo genérico donde se pueden utilizar los modelos de evaluación vistos anteriormente, como la TCT o la TRI. El objetivo es establecer unas pautas que permitan realizar la evaluación en tareas que pueden ser muy dispares, como tests, TAI, resolución de problemas complejos, y, especialmente, entornos de simulación (Mislevy, 2011). La base del DBE es el uso de la tecnología como medio para obtener información a partir de interacciones complejas y usar esa información como parte de un *razonamiento evidenciario* (En inglés *evidentiary reasoning*) (Mislevy et al., 2003b). A diferencia de como sucede en el razonamiento deductivo, donde se conocen unos hechos y éstos se utilizan para generar axiomas y realizar deducciones, en este tipo de razonamiento no se tiene certeza sobre los hechos, utilizándose las evidencias y la evaluación para razonar y descubrir reglas que expliquen la evidencia.

#### 3.4.2.1. Fases del DBE

El marco de trabajo establece una serie de pautas no sólo en cuanto a las componentes que se deberían utilizar en un sistema de evaluación, sino también en cuanto a las tareas de diseño de la evaluación. En este sentido, el DBE se componen de cuatro etapas básicas, que serán explicadas a continuación. Las dos últimas etapas, como podrá comprobarse, son más cercanas a la implementación y administración del sistema

de evaluación, ya que definen las componentes básicas que deberían definirse y cómo deberían ser usadas para proporcionar la evaluación (Mislevy et al., 2003a).

### Etapas previas al diseño de la evaluación

Antes de realizarse el diseño de la evaluación más cercano a la implementación de un sistema de evaluación, es necesario llevar a cabo dos etapas previas que, junto con las dos etapas básicas que serán explicadas a continuación, constituyen el marco de trabajo completo del DBE. Estas dos fases son las siguientes (Mislevy, 2011):

- *Análisis del dominio*: En esta etapa los diseñadores estudian el dominio sobre el que se va a realizar la evaluación desde diversas perspectivas para obtener la mayor cantidad de información posible. El objetivo es obtener información, patrones, estructuras y relaciones a partir de las cuales establecer teorías que permitan organizar la evaluación y definir los objetos involucrados. Para llevar a cabo este objetivo se deben considerar diversas fuentes de información como: ejemplos donde se realizan las tareas, características de las tareas, formas de representar las tareas, resultados generados con la tarea, estructuras del conocimiento y relaciones.

- *Modelado del dominio*: En esta fase, la información recopilada es procesada intentando buscar variables potenciales que puedan medirse y relaciones entre las tareas, evidencias posibles, y las variables potenciales. Esta etapa es una antesala de la tercera fase donde se prepara la definición de las componentes del sistema. Como parte de las tareas a estudiar se encuentran la definición de situaciones mediante las cuales se puedan involucrar al alumno en la evaluación; las formas en las que se pueden recopilar las evidencias; y qué evidencias son reflejo del conocimiento. Además, se estructura una argumentación sobre la información observable del estudiante que permita realizar inferencias y tomar decisiones en base al conocimiento del alumno.

### Marco de trabajo de evaluación conceptual

Tras las dos etapas anteriores, el DBE define una tercera etapa en la que se especifica de forma genérica las diversas componentes del sistema involucradas con la evaluación. Estas componentes son recogidas en el denominado *marco de trabajo de evaluación conceptual* (MTEC), cuyo nombre original en inglés es *Conceptual Assessment Framework* (Mislevy et al., 2003a). El MTEC se descompone en piezas más pequeñas, los modelos, que relacionan el proceso de evaluación y las actividades de un sistema de evaluación. Respecto al proceso de evaluación, los modelos proporcionan un marco de trabajo formal para especificar el conocimiento y las habilidades a medir, las condiciones a seguir para realizar las observaciones, y la naturaleza de la evidencia que será recopilada para realizar la inferencia. Respecto a las actividades, los modelos describen los requisitos para los procesos a realizar en el sistema en el que se administrarán las tareas.

Los diferentes modelos que componen el MTEC son resumidos en la figura 3.5, la cual ha sido adaptada a partir de (Mislevy, 2011). Éstos son enumerados y explicados a continuación:

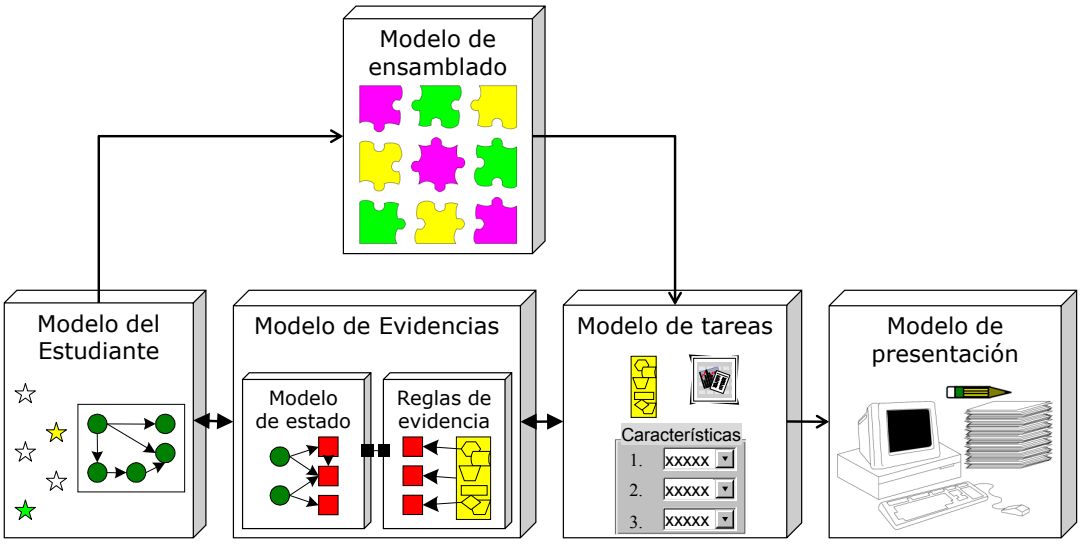



Figura 3.5: Marco de trabajo de evaluación conceptual del DBE.

- El *modelo del estudiante* contiene lo que se está midiendo, es decir, el conocimiento o habilidades sobre las que se quieren realizar alguna inferencia. Este modelo consiste en una o más variables asociadas a lo que se desea medir, las cuales acumularán la evidencia recopilada de las tareas. Por ejemplo, un modelo simple puede ser la proporción de tareas que es probable el estudiante responda correctamente, mientras que un modelo complejo puede contener diversas variables asociadas a diferentes habilidades que son necesarias para resolver una tarea.

- El *modelo de evidencias* especifica detalladamente cómo se deberían actualizar las variables del modelo del estudiante a partir de los *productos de trabajo* obtenidos de las tareas. Los productos de trabajo son la información producto de las acciones del alumno, como una respuesta a un ítem o la representación de una solución en un problema. El modelo de evidencias está compuesto de dos modelos más pequeños:

  - Reglas de evidencia: Determinan cómo obtener las evidencias observables a partir de los productos de trabajo. Por ejemplo, un modelo simple asociado a un test donde el producto de trabajo es la respuesta del estudiante, la evidencia observable es si el alumno ha respondido correctamente o no. Para determinar esto, las reglas de evidencia comprobarían la respuesta dada con la que es correcta. En un proceso más complejo en donde hay una tarea compleja con diversos pasos, la regla de evidencia puede ser comprobar si el alumno ha realizado cada paso.
  - Modelo de estado: Esta parte proporciona información sobre la conexión entre las variables del modelo del estudiante y las variables observables. Aquí es donde se suele usar los modelos psicométricos descritos a lo largo de este capítulo como la TCT o la TRI para, usando el resultado de las reglas de evidencia, actualizar el resultado en las variables del modelo del alumno. Es decir, este modelo establece la forma en que se actualiza el modelo del alumno. Hay que destacar que el DBE no impone ningún modelo concre-

to, siendo posible utilizar modelos probabilísticos como redes bayesianas o cualquier otro.

- El *modelo de tareas* establece cómo estructurar las situaciones de las que se pueden obtener las evidencias necesarias en el *modelo de evidencias*. Este modelo describe el material presentado al alumno y los *productos de trabajo* que son generados como respuesta. Además, contiene *variables de tareas* que describen propiedades y características de los *productos de trabajo* y del material presentado. Estas variables se asocian a información descriptiva de las tareas que es útil para seleccionar tareas, calibrar, diseñar o evaluar. Un modelo de tareas especifica el conjunto de tareas que pueden realizarse. Por ejemplo, un modelo de tareas simple describe los ítems de un test asociados a un tema, mientras que un modelo complejo puede ser un conjunto de problemas, una serie de variables que determinan qué obtener de ese problema y cómo para generar los *productos de trabajo*.

- El *modelo de ensamblado* describe cómo los modelos del estudiante, de evidencias, y de tareas interaccionan entres sí para producir la evaluación. En este modelo se especifica cómo de preciso debe ser la medición de las variables en el modelo del estudiante y cómo balancear la presentación de las tareas para que las evidencias reflejen adecuadamente la diversidad del dominio siendo evaluado. En un TAI, por ejemplo, la regla de selección del siguiente ítem sería parte de este modelo.

- El *modelo de presentación* determina la forma concreta de presentar el material al estudiante. En este sentido, hay muchos medios que permiten realizar esta tarea, no sólo a través del ordenador, sino también otros muchos medios como el papel, la transmisión oral, dispositivos móviles, etc.

#### Arquitectura de administración en cuatro procesos

La puesta en común de todas las componentes del MTEC es el *modelo de administración* (en inglés *delivery model*). Este modelo establece cómo los modelos del MTEC funcionan entre sí, tratando elementos que quedan fuera de los otros modelos como la plataforma de aplicación, el tiempo, y la seguridad. La descomposición del proceso de evaluación en las diferentes piezas explicadas permite combinarlas de formas diferentes, de acuerdo a las necesidades del sistema de evaluación.

Para realizar la última fase del diseño, relacionada con la administración del material modelado en el MTEC, el DBE propone un marco de trabajo genérico conocido como *arquitectura de administración de cuatro procesos* (AACP) (Almond et al., 2002), que en inglés es *Four Process Delivery Architecture*. Como su nombre indica, el AACP está compuesto de cuatro procesos, los cuales son mostrados en la figura 3.6, adaptada a partir de (Mislevy, 2011). En esta figura, la componente central, etiquetada con *Librería de tareas y evidencias* es una componente ficticia que proporciona los datos necesarios a cada proceso. Realmente, esta componente sería uno o varios elementos del MTEC necesarios para realizar el proceso. Los cuatro procesos y su relación con los elementos del MTEC son los siguientes:

- La *presentación* consiste básicamente en presentar cualquier material al estudiante y recopilar los productos de trabajo. En este caso, la librería central serían los modelos de presentación y de tareas.

- El *procesado de respuesta* es responsable de obtener las características clave de un producto de trabajo que representan las evidencias observables de una tarea particular. De este proceso se puede volver al estudiante, presentando el *refuerzo a nivel de tarea*, o puede irse al proceso de acumulación. En un test, el procesado de la respuesta sería determinar si esta es correcta. En este proceso las componentes del MTEC relacionadas son las reglas de evidencias con las que comprobar los productos de trabajo generados en el proceso anterior.

- La *acumulación de evidencias* consiste en actualizar el modelo del estudiante con las nuevas evidencias para producir la evaluación. Por ejemplo, en un test este proceso implicaría contar las respuestas correctas, si se está usando la TCT, o actualizar el nivel de $\theta$, si se usa la TRI. A partir de aquí se proporciona un *refuerzo resumen* al estudiante, o se continúa al siguiente proceso. En este proceso la librería de evidencias utiliza el modelo de estado para actualizar las variables del modelo del estudiante. También, el refuerzo de resumen utilizaría el modelo del alumno para mostrar la información.

- El proceso de *selección de actividad* se encarga de determinar cuál es la siguiente tarea y cuándo detener el proceso de evaluación. En este caso, un STI también tomaría decisiones como el cambio entre modos de evaluación o de instrucción. La componente del MTEC que se utiliza en este proceso es el modelo de ensamblado, que contiene las reglas para balancear la presentación de tareas y decidir cuál es la siguiente tarea más apropiada.

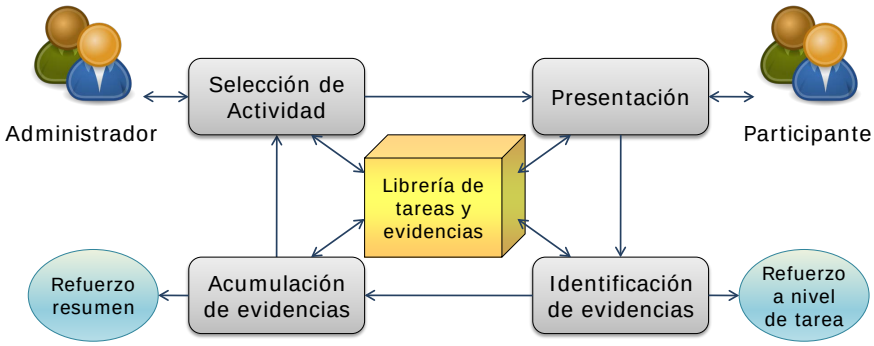



Figura 3.6: Arquitectura de administración en cuatro procesos.

#### 3.4.2.2. Ejemplos de uso

La genericidad del marco de trabajo permite ser aplicado con diferentes propósitos y usando diferentes combinaciones de elementos. Así pues, en este marco de trabajo se puede implementar desde un test, hasta tareas complejas de muy diversa índole. Para ilustrar la versatilidad del marco de trabajo y a modo de resumen de funcionamiento, se presentan dos ejemplos totalmente diferentes de uso en los que se visualiza el uso del marco de trabajo desde la perspectiva de la AACP:

- Uso del DBE para evaluar mediante tests: En este caso, el sistema comenzaría con la selección de un ítem aplicando alguno de los criterios de los TAI o bien, si

el test es fijo, usando el siguiente prefijado en la secuencia. Para ello se utiliza el modelo del estudiante, el cual puede ser un nivel de conocimiento $\theta$ establecido con la TRI, o el número de ítems correctos, según la TCT. Seguidamente, el proceso de presentación mostraría el ítem seleccionado al alumno y recogería la respuesta, la cual sería el producto de trabajo usado en el modelo de tareas. El procesado de la respuesta consistiría en aplicar las reglas de evidencia que simplemente comprobarían si la proporcionada por el estudiante se corresponde a la correcta para ese ítem. La acumulación de evidencias se haría bien con la TCT, mediante el conteo de las respuestas correctas, o bien mediante la TRI aplicando alguno de los métodos de evaluación del alumno para generar un nuevo valor de $\theta$. De nuevo, se volvería al proceso de selección donde se comprobaría un criterio de finalización antes de seleccionar el siguiente ítem. La comprobación de si se termina el test puede seguir alguno de los criterios de los TAI o simplemente la comprobación de una longitud fija en la TCT.

- El segundo ejemplo se muestra para un STI bajo el MBR: Siguiendo con la formulación clásica, el modelo del alumno del DBE se correspondería con el del MBR. Usando este modelo, la selección de un problema se haría utilizando los heurísticos que se aplican al modelo del alumno para determinar el problema sobre el que el alumno presenta mayores problemas. La presentación del problema se haría mediante un EIRP, generando como producto de trabajo la representación de la solución construida para el problema. El procesado de la respuesta se correspondería con la comprobación de las restricciones violadas y satisfechas. En este proceso existiría una correspondencia directa entre las reglas de evidencia y las restricciones. El refuerzo mostrado al alumno tras la comprobación de errores en el MBR se correspondería con el refuerzo a nivel de tarea del DBE. Para acumular las evidencias, bastaría con usar las reglas violadas y satisfechas para aumentar el conteo de las mismas en el modelo del alumno y actualizar el nivel heurístico del alumno. En este caso, el sistema no decide en ningún momento cuando finalizar, sino que la decisión la toma el alumno.

En la actualidad, el DBE ha sido aplicado a diversos sistemas, enmarcados principalmente en el ámbito de los entornos de simulación donde es complicado establecer una forma concreta de evaluar. Algunos ejemplos de uso en estos sistemas son recogidos en (Mislevy, 2011), donde se mencionan los desarrollos en el dominio de la ingeniería de redes de ordenador, mundos virtuales para investigaciones sobre ciencia, y resolución de problemas en higiene dental. Por mencionar algún otro ejemplo, el DBE puede aplicarse en el ámbito de la evaluación del lenguaje escrito y oral mediante tests (Mislevy et al., 2003b), o mediante redes bayesianas como parte de un sistema tutor (Almond et al., 2002). Otros ejemplos y más información sobre este marco de trabajo pueden encontrarse en la página Web de Robert Mislevy [1].

## 3.5. Conclusiones del capítulo

En este capítulo se han mostrado los métodos de evaluación que desde hace mucho tiempo se han venido desarrollando para determinar las características psicológicas y

[1] http://www.education.umd.edu/EDMS/mislevy/papers/

mentales de las personas. En esta tarea la rama de la Psicometría ha utilizado principalmente los tests como herramienta de evaluación. Dado que los test deben medir de la manera más objetiva posible y deben proporcionar una estimación del error de medida, surgen las teorías de test. Éstas son un conjunto de modelos matemáticos y estadísticos que buscan realizar la medición de las características o rasgos de las personas, a la vez que responden a la necesidad de proporcionar métodos de estimación de la precisión. En relación con esta tesis se considera como rasgo latente a medir el conocimiento y como sujeto del test el alumno.

Dos son las teorías más populares de evaluación: la TCT, la cual ha sido explicada brevemente, que se centra en estudiar las propiedades de los tests; y la TRI, explicada en mayor detalle por su relación con esta tesis, que se centra en estudiar las propiedades de los ítems. Como se ha visto, la TRI presenta numerosas ventajas sobre la TCT entre las que destaca la independencia de las estimaciones de la población de la que se obtienen. Esto hace que los tests puedan ser aplicados sobre diferentes poblaciones manteniéndose la validez de los resultados.

El elemento principal de la TRI es la curva que modela la probabilidad de responder correctamente a un ítem, la CCI. Las diferentes formas de representar esta curva dan lugar a los diferentes modelos la TRI. Así pues, entre las diferentes agrupaciones se puede destacar la que caracteriza los modelos por la función utilizada, distinguiéndose entre modelos paramétricos y no paramétricos. Los primeros son muy populares por su facilidad de manejo, ya que la curva se representa por una serie de parámetros, mientras que los segundos requieren enumerar los diferentes valores de la curva. Las propiedades de esta teoría de tests la convierten en una de las formas de evaluación más populares. No obstante también hay que tener en cuenta que su aplicación requiere de un estudio previo donde se precisa de un tamaño poblacional bastante amplio para garantizar la mayor precisión de la misma.

Una aplicación importante de la TRI son los TAI, los cuales son tests que se adaptan a las necesidades particulares del alumno, haciendo que el test presente ítems adecuados a su nivel de conocimiento y acorte la longitud para obtener una evaluación igualmente o más fiable que con tests normales y de longitud mayor. Los TAI establecen cómo orquestar el uso de la TRI en cada paso de la administración de un test, empezando por la calibración de los ítems. Cuando éstos están calibrados, se pueden utilizar para: determinar el conocimiento del alumno, en base a sus respuestas; para la selección adaptativa del siguiente ítem a mostrar; y para identificar cuándo el TAI puede terminar, como por ejemplo, comprobando que la medida tiene una precisión mínima.

Puesto que uno de los objetivos principales de esta tesis es tratar de extender la evaluación formal mediante tests a EIRP, se han revisado brevemente las características de los campos de investigación relacionados con este tipo de evaluación. En este sentido, se han identificado similitudes con la puntuación automática y el diagnóstico cognitivo. No obstante, las aplicaciones existentes son específicas del dominio donde se aplican y no definen una metodología genérica para aplicarse en otros dominios. En relación con esta genericidad, se ha explicado el marco de trabajo DBE, un trabajo que trata de extender los mecanismos de evaluación a tareas complejas, principal foco de interés de esta tesis.

El marco de trabajo DBE establece una serie de pautas y elementos genéricos sin especificar un dominio concreto de aplicación, una metodología de evaluación, ni restringe la forma de los elementos involucrados. Esto tiene la ventaja de que puede ser aplicado como guía para el desarrollo de cualquier sistema donde se puedan identificar

evidencias. No obstante, cada nuevo sistema donde se aplique requiere de un estudio diferente utilizando estructuras, elementos y métodos diferentes. En definitiva, supone diseñar el modelado del alumno y del dominio completamente desde cero. Es por ello que, sacrificando parte de esta genericidad, si se utiliza un paradigma como el MBR o los tutores cognitivos, la tarea de construcción de un nuevo sistema se reduciría a la construcción del modelo de dominio.

# Parte III

# Planteamiento

En esta parte se describe la principal aportación de la tesis. Primeramente, en el capítulo 4 se formaliza un modelo teórico para realizar la evaluación sumativa del alumno en EIRP. En el capítulo siguiente se explica la extensión realizada sobre la evaluación sumativa para proporcionar una evaluación formativa del alumno y su aplicación, tanto a sistemas de tests, como a EIRP.

# Capítulo 4

# Modelo de evaluación sumativa en dominios procedimentales

*La mayor sabiduría que existe*
*es conocerse a uno mismo*

Galileo Galilei (1564 - 1642)

**Resumen:** En este capítulo se formaliza un modelo teórico que utiliza dos de los paradigmas detallados en capítulos anteriores para realizar una evaluación sumativa del estudiante en EIRP.

Siguiendo con el objetivo de diseñar una metodología para el diagnóstico formal del alumno en EIRP, en los capítulos anteriores se han detallado los diferentes paradigmas que permiten el modelado del alumno en este tipo de entornos. Como se ha podido ver, hay dos paradigmas que destacan en la actualidad por tener una efectividad probada: los tutores cognitivos y el MBR. Determinar cuál es la mejor técnica para el modelado del estudiante no es una tarea sencilla. Cada una tiene sus ventajas e inconvenientes que no implican que una sea mejor que la otra. Diversos estudios existen en la literatura que comparan la idoneidad de las dos técnicas más populares de modelado. En una de las primeras comparativas existentes, realizada por Mitrovic et al. (2003), se menciona que los tutores cognitivos son una buena elección cuando los dominios contienen tareas de resolución de problemas bien definidas, es decir, cuando los pasos son claros y la construcción del conjunto de reglas de producción del dominio es fácil de elaborar. Sin embargo, el MBR requiere menos tiempo de desarrollo y esfuerzo para construir el modelo del dominio. Otras comparativas como la de Kodaganallur et al. (2005); Mitrovic y Ohlsson (2006); Kodaganallur et al. (2006), generaron controversia a la hora de establecer las limitaciones de una u otra técnica.

Al final del capítulo 2, en la sección 2.4, el autor de esta tesis realiza su propia comparativa en base a la revisión detallada sobre los dos paradigmas. Las debilidades encontradas en el MBR en la componente que sería equivalente a la TC de los tutores cognitivos hace que los sistemas MBR necesiten extenderse para mejorar la forma de modelar al alumno. Esta necesidad de mejora supone una de las razones principales que motiva el uso de este paradigma en lugar de los tutores cognitivos. Otro motivo que llevó a usar el MBR fue la identificación de una oportunidad clara de aplicar una metodología

formal de evaluación para cubrir esta necesidad. Ésta encajaba perfectamente con la abstracción sobre la que se asienta la base teórica de las restricciones, como se verá durante la sección 4.3.

Las razones anteriores unidas con la eficiencia que en ese momento destacaba en las investigaciones realizadas sobre el paradigma (Mitrovic et al., 2007); las ventajas que proporciona en la construcción de EIRP, las cuales se resumieron en el apartado 2.4; la teoría subyacente Ohlsson (1992, 1993, 1994), diseñada para modelar al alumno de una forma más simple que otras técnicas como la TM; así como los estudios que reflejaban las ventajas, en cuanto a tiempo requerido y facilidad de aplicación, en comparación con la otra gran técnica de modelado (Mitrovic et al., 2003; Mitrovic y Ohlsson, 2006), hicieron que el MBR fuera el paradigma elegido para la construcción de EIRP.

En relación con la metodología de evaluación formal utilizada, se eligió la ya explicada TRI. La elección de este mecanismo es más bien una consecuencia directa de otro de los objetivos perseguidos y que pretende cubrir una de las limitaciones de los mecanismos formales de evaluación principalmente aplicados en los sistemas de tests. En estos sistemas, la metodología de evaluación por excelencia es la TRI, por lo que cubrir esta limitación supone el uso de la mencionada metodología. Además, las características de la misma como técnica objetiva y bien fundamentada, explicadas en la sección 3.2 hacen de ella una herramienta deseable para realizar la evaluación de manera formal en EIRP. Aunque esta combinación se presenta como una forma de paliar la principal limitación de los tutores MBR, el beneficio también está presente en el sentido contrario. Como se podrá ver a lo largo de este documento, el uso de una técnica de modelado del alumno en EIRP permitirá que la TRI pueda ser aplicada en dominios procedimentales, proporcionando una forma de superar su mencionada limitación. No obstante, hay diversas consideraciones a tener en cuenta para garantizar la validez de la metodología, las cuales serán discutidos durante este capítulo.

El objetivo principal de este capítulo es formalizar un modelo teórico que permita aplicar la TRI en los EIRP para poder realizar la evaluación en dominios procedimentales. Para ello, en la sección 4.1 se describen de forma general las características del modelo. Seguidamente, se presenta una revisión breve sobre los pocos trabajos existentes que utilizan un enfoque similar. A continuación, se presenta la analogía encontrada entre los sistemas de tests y los tutores MBR que hace posible la utilización conjunta de los dos paradigmas mencionados. En la sección 4.4 se definen formalmente los elementos del modelo teórico conceptos necesarios para formalizar posteriormente la metodología de evaluación sumativa, la cual se presentará en la sección 4.5. Después, en la sección 4.6 se explica como el modelo puede ser generalizado a cualquier paradigma de EIRP. Por último, se presentan las conclusiones del capítulo.

## 4.1. Descripción general del modelo

El modelo de evaluación que se propone tiene varias características que fueron mencionadas como parte de los objetivos de esta tesis en la sección 1.2. A continuación se mencionan las características del modelo y se justifica su necesidad:

### Modelo de evaluación formal o sistemático en EIRP

Tal y como se mencionó en la sección 2.4, hasta ahora la evaluación formal se limita a tareas sencillas que pueden ser modeladas mediante ítems. El modelo trata de salvar

esta limitación extendiendo la evaluación a entornos procedimentales con un proceso de resolución complejo. Para ello se utiliza el paradigma del MBR como entorno en el que los alumnos pueden llevar a cabo la resolución de este tipo de tareas y problemas. Paralelamente, se trata de paliar otra de las limitaciones detectadas en el MBR que radica en el uso de heurísticos para estimar el conocimiento del alumno. De forma general, en el campo de los STI las diferentes técnicas de modelado carecen de mecanismos formales de evaluación. Si bien existen técnicas bien fundamentadas basadas en teoría probabilística, éstas se centran en modelar el aprendizaje sin modelar al alumno, lo cual es lógico dado que su objetivo es el mejorar el aprendizaje. Sin embargo, la inclusión de una metodología formal para realizar el diagnóstico incrementaría y daría mayor validez al modelo que el sistema guarda del alumno, con la consiguiente mejora de los mecanismos de instrucción que lo utilizan.

### Modelo de evaluación cuantitativo

Los modelos cualitativos, para realizar la evaluación, tienen en cuenta los fenómenos que ocurren en el entorno donde el proceso de evaluación es realizado con tal de proporcionar una visión completa del proceso. Así pues, se utilizan elementos como la labor del profesor y circunstancias que rodean la evaluación, lo cual añade subjetividad al juicio emitido. A diferencia, los modelos cuantitativos estudian elementos más objetivos. Aunque dentro de los modelos cuantitativos también existen métodos heurísticos, para cumplir con la característica anterior, éstos se evitarán, buscando métodos bien fundamentados que puedan ser estadísticamente confiables y generalizables. Esto implica que el modelo desarrollado pueda ser aplicado de forma general a cualquier población de estudiantes, dejando de lado la ocurrencia de fenómenos subjetivos e intentando maximizar la objetividad.

### Modelo de evaluación genérico

Otra de las características mencionadas como parte de los objetivos iniciales es que el modelo desarrollado sea lo más genérico posible. Para ello, a partir de una implementación concreta inicial, que en este caso se realiza utilizando el MBR, se intentará generalizar el modelo, en la medida de lo posible, en varios niveles:

- El nivel más bajo de generalidad se corresponde con el modelo de la TRI utilizado para modelar los elementos probabilísticos básicos de la metodología de evaluación. En este sentido, si bien el modelo utilizado tiene implicaciones prácticas a la hora de implementarse, se buscará que el modelo teórico sea independiente de estas características. Este nivel será tratado en la sección 4.5.

- En un nivel intermedio, y probablemente el más importante, el modelo debería ser independiente del dominio en el que se aplique. Esto quiere decir, que el modelo pueda evaluar al alumno de manera genérica en cualquier dominio, siempre considerando que éste sea de carácter procedimental. Con la utilización de un paradigma para la creación de EIRP como el MBR, esta abstracción ya se cumple, pues viene implícita con la técnica de modelado.

- Como nivel superior, sería ideal que el modelo pudiese ser aplicado independientemente del paradigma de construcción de EIRP utilizado. De esta forma, se busca

que pueda ser aplicado no sólo al MBR, sino que cualquier otro paradigma pudiese encajar en un marco de trabajo genérico. La forma de generalizar el modelo en este nivel y las diferentes consideraciones a tener en cuenta será explicada en la sección 4.6.

### Modelo de evaluación sumativa

En la sección 1.1.1 se explicaron los tipos de evaluación y se dio una de las posibles clasificaciones que distinguía tres tipos: la diagnóstica, la formativa y la sumativa. De cara a aplicar este proceso combinado de evaluación, lo lógico es usar estos tres tipos en el orden en el que se han mencionado. Primeramente se realizaría una evaluación sumativa inicial para determinar el conocimiento de partida y saber cuál es la situación inicial. Posteriormente se realizaría un periodo de evaluación formativa en la que se va informando al alumno sobre sus carencias de forma que éste pueda corregirlas. Al final del periodo formativo, se llevaría a cabo una evaluación sumativa en la que se emite el juicio sobre su conocimiento. Sin embargo, a la hora de diseñar un modelo que tenga en cuenta estos tipos, antes de diseñar una evaluación formativa es necesario disponer de un mecanismo sumativo.

En la comunidad científica asociada con el campo de la IA en la educación, en el que se enmarca este trabajo, la afirmación anterior genera controversia y polémica. Esto es así puesto que al mencionar la evaluación formativa y la sumativa, por asociación, se suele pensar en el proceso de aplicación en el que primero se aplica la evaluación formativa y, posteriormente, la sumativa. Así pues, decir que para construir una evaluación formativa, primero hay que disponer de una sumativa, resulta antinatural. No obstante, la base de esta afirmación está en lo que contiene una evaluación formativa. De acuerdo con Scriven (1967); Ramaprasad (1983); Sadler (1989); Black y Wiliam (1998), ésta contiene un juicio y un refuerzo que permita guiar en el proceso de aprendizaje. Taras (2005), además, precisa que la evaluación formativa es en sí una evaluación sumativa más un refuerzo. Partiendo de la apreciación de que se está diseñando la metodología en lugar de aplicarla, se justifica el orden de la frase anterior.

De esta forma, de cara a implementar un modelo de evaluación formativo, primero es necesario disponer de una evaluación sumativa. Ésta será posteriormente extendida con un refuerzo y con las acciones adecuadas que permitan guiar el proceso de aprendizaje para cumplir con los objetivos de aprendizaje de la evaluación sumativa. El modelo teórico que se presenta en este capítulo se centra en el diseño de este proceso de evaluación sumativa, el cual será extendido en el siguiente capítulo con las características necesarias para proporcionar una evaluación formativa.

Puesto que en el ámbito de la IA en la educación en donde se enmarca el trabajo de esta tesis, se suelen utilizar los términos *evaluación sumativa* y *evaluación formativa* para referirse a evaluaciones sobre sistemas educativos. Esto no debe confundirse con el uso que se está haciendo de estos términos en este documento, el cual recae en la evaluación educativa y se aplica a los alumnos.

## 4.2. Trabajos similares

En los dos capítulos anteriores se ha detallado el trabajo relacionado en las dos áreas que forman los pilares básicos de esta tesis, pero esta revisión del estado del arte se hace de manera individual sobre cada campo. En la evaluación de tareas complejas mediante

la TRI ya se ha mencionado el marco de trabajo del DBE, explicado en la sección 3.4.2. Este marco de trabajo es muy genérico y no establece una forma concreta de realizar el modelado del alumno y del dominio, que sí queda definido con el uso de alguno de los paradigmas para la construcción de EIRP explicados. A la hora de utilizar un enfoque combinado más específico, como el que utiliza el modelo de esta tesis, existen muy pocos trabajos que persigan un objetivo similar. Esta sección trata de recopilar los trabajos más destacados cuyo objetivo y características son similares a la sinergia utilizada en esta tesis.

Hasta la fecha, no hay constancia en la literatura del uso de metodologías como la TRI para la evaluación en sistemas basados en el MBR. En el apartado 2.3.3.2 se explicaba brevemente el uso de técnicas bien fundamentadas como las redes bayesianas para modelar el alumno. Sin embargo, este modelo tiene el inconveniente de la escalabilidad, al ser aplicable en sistemas con un número de restricciones muy reducido. Además, el objetivo de utilizar este enfoque para favorecer el proceso de aprendizaje, hace que no se disponga de un mecanismo de evaluación sumativa. Esto es así puesto que solamente se modelan probabilidades de haber aprendido las restricciones. Por estos motivos, éste sería el primer modelo que puede llevarse a cabo sin problemas de escalabilidad, que pueda implementarse para realizar el diagnóstico del alumno y la instrucción en estos sistemas de manera formal.

En otros paradigmas como los tutores cognitivos sí que existen algunos trabajos basados en la TRI para mejorar las capacidades instructivas de los tutores. Por ejemplo, Pardos y Heffernan (2011) usan el parámetro dificultad de un ítem para extender un modelo bayesiano sobre el aprendizaje del alumno. Sin embargo el trabajo presentado no profundiza en la forma de determinar la dificultad, ni justifican los valores que toman como fuente de evidencia, lo que, unido con su aparición relativamente reciente, hace pensar que probablemente la validez formal del mismo pueda estar todavía por probar, probablemente debido a la falta de madurez.

Otro de los trabajos sobre tutores cognitivos que están inspirados en la TRI es el de Pavlik et al. (2009b), el cual ya ha sido mencionado en el apartado 2.2.2. Este trabajo usa el análisis del factor de rendimiento para proporcionar adaptación. Sin embargo, el modelo presentado solamente está relacionado con la TRI por el modelo logístico desarrollado que se utiliza para dirigir la TC, el cual es una adaptación de uno de los modelos de la TRI. Sin embargo, en los trabajos existentes no se menciona tampoco cómo realizar la calibración ni se tienen en cuenta los supuestos bajo los cuales es aplicable la TRI.

## 4.3. Aplicabilidad de la TRI en tutores MBR

La aplicación de los fundamentos de la TRI a tutores MBR es posible gracias a una analogía identificada entre dos entornos educativos, a priori, diferentes. Los primeros son los sistemas de tests, donde el objetivo primordial es evaluar los conocimientos del alumno. Los segundos, son los tutores MBR, cuyo propósito es la mejora del proceso de enseñanza y que da mucha menos importancia a la evaluación. Esta analogía se encuentra en el núcleo de cada entorno educativo y concretamente en los elementos clave sobre los que se basa cada uno: el ítem en los sistemas de tests, y la restricción en el MBR (ver figura 4.1). Las similitudes que definen una analogía se encuentran en varias características:

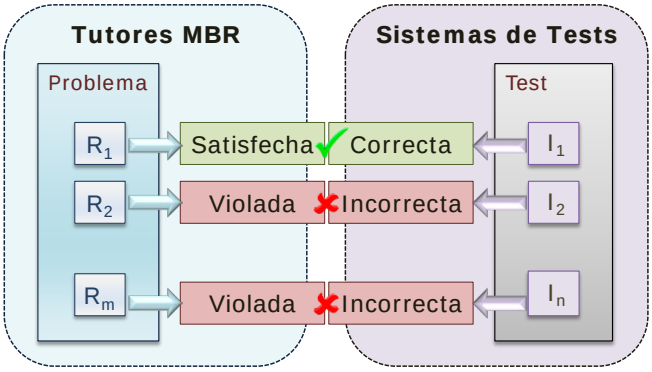



Figura 4.1: Similitud existente entre los tutores MBR y los sistemas de tests.

- Finalidad: Ambos elementos, aunque usando una fuente diferente, sirven como instrumento para proporcionar evidencia del conocimiento. Los ítems usan como fuente un estilo directo para recabar el conocimiento que consiste en la formulación de una pregunta interrogativa sobre el concepto a evaluar. Las restricciones, si bien no tienen como objetivo final la evaluación formal, se usan también como reflejo del conocimiento del alumno usando como fuente los principios violados durante un proceso complejo de resolución de problemas.

- Naturaleza y forma: Ambos modelan de forma declarativa un concepto o componente del conocimiento. Por un lado, las restricciones se implementan en forma de regla, lo cual deja patente su naturaleza declarativa. Los ítems, por otro lado, también se representan de forma declarativa como un enunciado y una serie de hechos asociados a las respuestas. Esto no implica que ambos sean reflejo de conocimiento declarativo, pues tanto uno como el otro pueden reflejar conocimiento procedimental. Un ejemplo claro en el MBR son las *restricciones camino* explicadas en la sección 2.3.2. En la TRI, se puede evaluar también conocimiento procedimental mediante ítems simples que requieran un proceso sencillo de aplicación del conocimiento procedimental.

- Valores resultado: Los dos elementos pueden tomar dos valores que representan el rendimiento correcto o incorrecto del alumno, los cuales pueden ser utilizados como una fuente de evidencia para estimar el nivel de conocimiento. En general, un ítem puede ser evaluado como correcto o incorrecto. En el MBR, una restricción puede ser satisfecha, lo que se asimila a un ítem contestado correctamente; o violada, al igual que un ítem incorrectamente contestado. Aunque los ítems pueden tomar muchos más estados que los dos mencionados si se utilizasen modelos de crédito parcial (Guzmán et al., 2007), la equivalencia encontrada implica el tratamiento de los ítems como modelos dicotómicos en los que sólo dos resultados pueden considerarse.

- Estructura: A nivel estructural y de forma más general también existe equivalencia en la forma en la que los elementos específicos son agrupados para determinar el conocimiento de un tema concreto. En los sistemas de tests los ítems son agrupados cubriendo un conjunto de conceptos, normalmente sobre un tema. De forma similar, en los tutores MBR las restricciones son agrupadas en problemas

que cubren los conceptos asociados para la resolución del problema.

La similitud encontrada entre los ítems de un test y las restricciones de un problema ha permitido elaborar el modelo que supone la principal aportación de esta tesis. Este modelo consiste en extrapolar los mecanismos de evaluación que tienen su origen en los sistemas de tests, a los sistemas tutores MBR, utilizando para ello las restricciones como si fueran ítems dentro de un test. De esta forma, el modelo diseñado pasa por extender la arquitectura típica del MBR con ciertos elementos y funcionalidades típicas de los sistemas de tests que se explican en la siguiente sección.

Previamente a la introducción del modelo es necesario revisar si las tres características deseables de la TRI, explicadas en la sección 3.2, se siguen cumpliendo en este tipo de sistemas.

- En cuanto a la independencia local, ésta se cumple debido a la naturaleza propia de las restricciones. Concretamente, por reflejar principios del dominio que deben ser independientes entre sí. Esto garantiza que la probabilidad conjunta de ocurrencia de varias restricciones se pueda calcular como el producto de las probabilidades individuales de cada restricción.

- En relación con la unidimensionalidad, el factor que influye en la construcción de la solución es el conocimiento del alumno. Como se mencionó en la sección 1.1.2 del capítulo de introducción, en este proceso habrá dos tipos de conocimiento involucrados, el conocimiento declarativo y el procedimental. No obstante, dado que se puede considerar un conocimiento más general (de Jong y Ferguson-Hessler, 1996), y que este principio puede relajarse dado que en la respuesta puede existir otras características asociadas al estado del alumno y su entorno, supondremos que en este sentido no hay problema para aplicarse.

- Otra propiedad deseable, es la de invariancia. Puesto que esta propiedad está relacionada con el ajuste del modelo a los datos, no se puede determinar a priori teóricamente, sino que hay que probarla mediante experimentación. Esta propiedad no es tan importante si se cumplen los supuestos de independencia local y unidimensionalidad, pues, en ese caso, aunque el ajuste del modelo a los datos no sea bueno, todavía se pueden realizar inferencias sobre el conocimiento del alumno (Lee, 2007; Junker, 2011). En la sección 7.4 se resumen las pruebas empíricas realizadas para estudiar esta propiedad.

## 4.4. Definiciones formales

Como ya se ha explicado en la sección 3.3, la TRI es una teoría fundamental para realizar la evaluación y la adaptación en los TAI. Cualquier modelo de la TRI tiene como característica central un modelo de respuesta que se usará como fundamento teórico y que será la base para la inferencia del conocimiento y el proceso de adaptación del alumno. Esta sección presenta las definiciones que forman parte de la formalización teórica de un modelo de respuesta para EIRP. El objetivo es representar los elementos formales que normalmente se usan sobre los ítems de los tests, en las restricciones de los sistemas MBR. Como punto de partida se define formalmente el modelo de dominio a partir del conjunto de restricciones y del de problemas.

**Nota**: Por mantener una notación sin pérdida de generalidad se obvia la utilización explícita del dominio en la formalización del modelo, asumiendo que todas las definiciones se aplican a un dominio educativo concreto. Esta consideración no hace que el modelo presentado deje de ser válido para sistemas multidominio. El hecho de considerar diferentes dominios implicaría simplemente la extensión de las definiciones con la identificación del dominio concreto sobre el que se aplicaría.

**Definición 4.1** (Conjunto de restricciones de un dominio)**.** *Dado un dominio d sobre el que se define un sistema MBR, el conjunto de las n restricciones que conforman su modelo de dominio se define como* $\tau = \{r_1, r_2, .., r_n\}$.

**Definición 4.2** (Conjunto de problemas de un dominio)**.** *El conjunto de los problemas de un dominio d asociado a un sistema MBR, se define como* $\phi = \{p_1, p_2, .., p_m\}$*, donde m es el número de problemas del modelo del dominio.*

Para comenzar, veamos en qué consiste el modelo de respuesta en sistemas de tests, para posteriormente definir el de los tutores MBR. Dado un ítem cualquiera $i$, éste, además de un enunciado, contendrá una serie de opciones de respuesta. Cada opción de respuesta vendrá representada por $o_{ij}$, siendo $j$ un valor que identifica unívocamente la respuesta dentro de ese ítem $i$. El conjunto de todas las respuestas de $i$ tendrá una cardinalidad $m_i$ y viene representado por $O_i = \{o_{i0}, o_{i1}, o_{i2}, ..., o_{im_i}\}$. Como se observa, este conjunto, además de las respuestas del ítem contiene la opción $o_{i0}$ asociada a dejar el ítem correspondiente sin contestar (respuesta en blanco). Usando este conjunto de respuestas, la formalización del modelo de respuesta se asienta en la definición de una *función de respuesta seleccionada* que identifica si una respuesta $o_j$ ha sido seleccionada para el ítem $i$ y, en base a ésta, *una función de evaluación de la respuesta* que determina si las respuestas proporcionadas forman un patrón correcto para la respuesta (Guzmán, 2005).

De forma similar, una restricción tiene asociada una respuesta, con la diferencia de que ésta es una solución compleja construida por el alumno y que los valores que toma no se corresponden a ninguna opción. Consideremos una restricción $r$. Ésta no tendrá enunciado, pues no se plantea directamente como sucede con los ítems, sino que es implícita al proceso de resolución de un problema. Además, la respuesta de una restricción no se compara con las diferentes opciones, como pasa en los ítems. Por el contrario, la solución construida se usa para determinar la violación o satisfacción de las restricciones, que resultarán en una respuesta incorrecta o correcta, respectivamente.

Otra diferencia entre las restricciones y los ítems es que las primeras no poseen una opción de respuesta en blanco. Siempre que una restricción sea presentada tendrá una respuesta correcta o incorrecta. Sin embargo, el hecho de que una restricción sea presentada, el cual se corresponde a la relevancia de una restricción, se debe tener en cuenta en la formulación del modelo. El efecto que tiene la condición de relevancia en esta formulación es el de una función que se define a continuación y que tiene dos vertientes, dependiendo de si se aplica a una solución particular, o de forma más general, a un problema. Previamente a definir estas dos funciones, es necesario definir lo que es un hecho, una solución, los conjuntos de soluciones posibles y el conjunto de soluciones dadas a un problema.

**Definición 4.3** (Hecho de una solución)**.** *En el ámbito que nos ocupa, un hecho h es un elemento que contiene información concreta representando partes de las soluciones*

*que se pueden construir en un EIRP. Por tanto, éste representa principalmente la evidencia recopilada de la interacción con el alumno, pero también representa piezas de información de una solución ideal proporcionada por un profesor. Dependiendo del dominio donde se utilice, un hecho puede representar diferentes tipos de información, como valores numéricos, texto, coordenadas en una imagen, etc.*

**Definición 4.4** (Solución particular de un problema)**.** *El término solución en esta formalización no se refiere al elemento que hace que el problema sea resuelto, sino que se utiliza para referirse a una respuesta particular que el alumno da sobre un problema, la cual puede ser correcta o incorrecta. Formalmente, una solución i asociada a un problema p, tal que $p \in \phi$, es un conjunto de m hechos que representan un estado concreto sobre la solución y que se define como $s_{pi} = \{h_1, h_2, .., h_m\}$. Este conjunto es dependiente del dominio y tiene los valores que cada elemento de la solución toma. Por ejemplo, si el dominio en cuestión es el de las fracciones, el conjunto de hechos vendrá definido típicamente por dos hechos con valores numéricos, uno para el numerador y otro para el denominador. Lógicamente, se asume que los problemas del dominio son resolubles mediante algún estado solución.*

**Definición 4.5** (Espacio de soluciones de un problema)**.** *A partir de un problema p, tal que $p \in \phi$, puede haber una o más soluciones particulares. Al conjunto de todas estas soluciones posibles lo llamaremos espacio de soluciones del problema y vendrá representado por $S_p$. Este conjunto no puede definirse por extensión, pues podría llegar a ser infinito. Esto es así porque resulta de las diferentes combinaciones que se pueden hacer con los diferentes hechos que pueden componer la solución. La posibilidad de que un hecho esté reflejando un elemento tan simple como un número, hace que haya infinitas soluciones posibles en el conjunto que se está definiendo.*

**Definición 4.6** (Conjunto de soluciones de un problema)**.** *Dado un problema p del dominio, el conjunto de las n soluciones dadas a ese problema en un momento concreto vendrá representado por $\sigma_p = \{s_{p0}, s_{p1}, s_{p2}, .., s_{pn}\}$ tal que $s_{pi} \in S_p$. Cada solución $s_{pi}$ que cumple que $i \in [1, n]$ es una solución dada por algún estudiante a ese problema. Además, la solución $s_{p0}$ es la que proporciona el profesor como solución ideal (ver apartado 2.3.3.1 para más detalle). Este conjunto irá creciendo conforme los alumnos realicen intentos sobre el problema en el sistema MBR.*

**Definición 4.7** (Espacio de soluciones del dominio)**.** *Al conjunto de todas las soluciones que pueden darse para todos los problemas de un sistema MBR será denominado espacio de soluciones posibles del dominio. Este conjunto vendrá definido por $\alpha = \bigcup_{i=1}^{m} S_i$. En esta expresión se unen los espacios de soluciones de cada uno de los m problemas del dominio.*

**Definición 4.8** (Función de relevancia de una restricción para una solución)**.** *La relevancia de una restricción r sobre una solución concreta $s_{pi}$ asociada a un problema p del dominio, se denota como $\mu : \tau \times \phi \times \alpha \rightarrow \{0, 1\}$. Esta función determina si para una solución dada por un alumno, la restricción r es relevante. En términos del MBR, y teniendo en cuenta la definición 2.2 de la condición de relevancia de una restricción,*

*la restricción será relevante para la solución si todos los hechos que representan la solución hacen que la condición de relevancia $\varrho_r$ sea cierta. El rango de valores que puede tomar se puede definir mediante la siguiente función por partes:*

$$\mu(r,p,s_{pi}) = \begin{cases} 1 & Si\ \forall h_i \in s_{pi}, \varrho_r = \top \\ 0 & En\ otro\ caso \end{cases} \tag{4.1}$$

**Definición 4.9** (Función de relevancia de una restricción para un problema)**.** *La relevancia de una restricción r se define de forma general para un problema p del conjunto de los problemas del dominio mediante $\rho : \tau \times \phi \rightarrow \{0,1\}$. Para determinar el resultado de la función se busca entre los elementos del conjunto de soluciones dadas al problema aplicando para cada uno de ellos la función de relevancia de la restricción para una solución definida anteriormente.*

$$\rho(r,p) = \begin{cases} 1 & Si\ \exists s_{pi} \in \sigma_p\ tal\ que\ \mu(r,p,s_{pi}) = 1 \\ 0 & En\ otro\ caso \end{cases} \tag{4.2}$$

En base a las definiciones anteriores se puede definir la función de evaluación de una restricción que es la base del modelo de respuesta que permitirá aplicar la TRI en los sistemas MBR. Mientras que esta función de evaluación en el ítem tiene en cuenta el patrón de respuestas dadas por el alumno, en la restricción, se hará en base a la solución proporcionada. La evaluación de una restricción realmente consiste en determinar si ésta ha sido violada o satisfecha. Para ello, se debe comprobar la condición de satisfacción formalizada en la definición 2.3 con los hechos que representan la solución dada (definición 4.3). Teniendo en cuenta la formalización de las restricciones del apartado 2.3.2, la función de evaluación sólo se puede aplicar si la condición de relevancia de la restricción ha sido satisfecha previamente. Con estas consideraciones, la función se define como sigue:

**Definición 4.10** (Función de evaluación de una restricción)**.** *Dada una restricción r; un problema p; y una solución $s_{pi}$ tal que $s_{pi} \in \sigma_p \wedge \mu(r,p,s_{pi}) = 1$, es decir, la solución pertenece al problema p y es relevante en éste; la función de satisfacción de la restricción respecto de la solución dada se denota por $\delta : \tau \times \phi \times \alpha \rightarrow \{0,1\}$. Esta función comprueba si alguno de los hechos que componen la solución $s_{pi}$ viola la condición de satisfacción de la restricción ($\varsigma_r$). En caso de que todos los hechos satisfagan esta condición, la restricción es satisfecha y el resultado de la función es 1; en caso contrario es 0. La definición de la función por partes sería la siguiente:*

$$\delta(r,p,s_{pi}) = \begin{cases} 1 & Si\ \forall h_i \in s_{pi}, \varsigma_r = \top \\ 0 & En\ otro\ caso \end{cases} \tag{4.3}$$

De una forma similar a como se define la CCI, las restricciones dispondrán de una *Curva Característica de la Restricción* (CCR). La forma de calcular esta curva dependerá del modelo utilizado para ello, el cual podrá ser cualquiera de los mencionados en el apartado 3.2.1. Dado que las respuestas son siempre dos, los modelos dicotómicos se presentan a priori como los más apropiados para esta tarea. El modelo teórico que se propone no está ligado a un modelo de la TRI concreto, ya que, como se mencionaba en las características del modelo, se busca la genericidad centrándose en los elementos que son independientes y que permiten establecer una metodología de evaluación.

Por ejemplo, si se utilizase el modelo paramétrico 3PL para modelar la CCR, la función que determinaría su forma es exactamente la misma que la dada por la ecuación 3.7. La expresión utilizaría los tres parámetros $a_r$, $b_r$ y $c_r$, con la salvedad de que la notación de la función $P_r(\theta)$ haría referencia a una restricción $r$ en lugar de a un ítem.

**Definición 4.11** (Curva Característica de la Restricción)**.** *La CCR es una función monotónica creciente, que representa la probabilidad de que un alumno con rasgo latente estimado $\theta$ satisfaga la restricción, siendo ésta relevante. Dada una restricción $r$, su curva característica viene representada por una función de probabilidad $P_r(\theta) = P(r|\theta)$. Esta función, al ser de probabilidad vendrá definida en el dominio de los reales y tendrá como rango el intervalo* $[0, 1]$.

En los sistemas de tests se suele modelar también una curva característica para cada una de las opciones que puede tener un ítem. El conjunto de las opciones forma un espacio probabilístico en el que los sucesos posibles son el conjunto de patrones de respuesta. En el caso de las restricciones la probabilidad de cada una de las opciones es equivalente al de un ítem cuya respuesta es verdadera o falsa. De esta forma, la probabilidad de que la restricción tenga como respuesta el valor $S$, de satisfacción, se corresponde con la CCR. Dado que el espacio probabilístico de las opciones de una restricción sólo contiene dos sucesos, la suma de ambos será una distribución uniforme que toma el valor 1. Por tanto, la probabilidad de violar la restricción o lo que es lo mismo, de que la respuesta de la restricción sea el valor $V$, se puede modelar como la probabilidad complementaria de la CCR y da lugar a la definición de la *Curva Característica Complementaria de la Restricción* (CCCR).

**Definición 4.12** (Curva Característica Complementaria de la Restricción)**.** *La probabilidad complementaria de la curva CCR es una función monotónica decreciente, que representa la probabilidad de que un alumno con rasgo latente estimado $\theta$ viole la restricción, siendo ésta relevante. Dada una restricción $r$, esta probabilidad resultante sería su CCCR, la cual viene representada por la función $Q_r : (-\infty, \infty) \to [0, 1]$, definida como $Q_r(\theta) = Q(r|\theta) = 1 - P(r|\theta)$. La forma de esta función de probabilidad es similar a la de la figura 4.2, donde, cuanto mayor es el rasgo latente del alumno ($\theta$), menor es la probabilidad de violarse la restricción*

En los sistemas MBR la opción asociada a la violación de la restricción es la base del paradigma. De hecho, la teoría sobre la que se asienta se conoce como *aprendizaje a partir de los errores*. Es por ello, que la probabilidad de la violación es utilizada por los autores que dieron lugar al paradigma como herramienta básica de sus sistemas. No obstante, por mantener una formulación similar con la TRI, en la que la evidencia positiva es usada para modelar los ítems, se utiliza la satisfacción de las restricciones para el modelo de respuesta. Como se ha mencionado anteriormente, y en caso de necesitarse, la probabilidad de violar una restricción se calcula directamente como se indica en la CCCR.

## 4.5. Evaluación sumativa en MBR mediante la TRI

Como ya se ha mencionado anteriormente en la sección 4.1, uno de los objetivos del modelo teórico planteado en este capítulo es el de disponer de una metodología

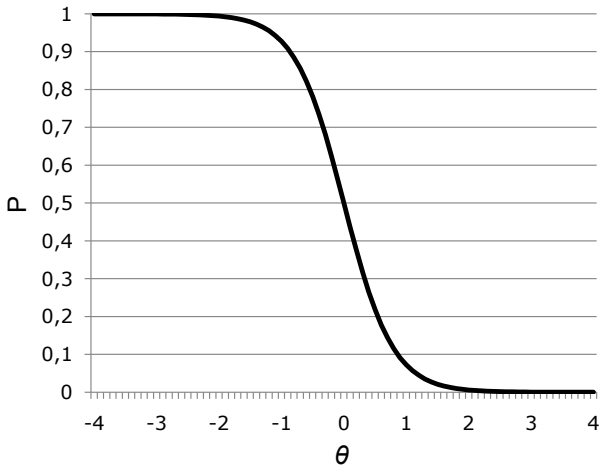



Figura 4.2: Curva característica asociada a la violación de una restricción.

de evaluación sumativa, previamente a la elaboración de la metodología de evaluación formativa. La base de este modelo requiere, al igual que sucede en sistemas de test, de un proceso inicial en el que se deben estimar las probabilidades de violar cada restricción. Posteriormente, una vez obtenidas estas probabilidades en las CCR de cada restricción, se pueden aplicar sobre las evidencias de un alumno, usando el modelo de respuesta anteriormente definido, con el fin de estimar su nivel de conocimiento.

### 4.5.1. Calibración de las restricciones

El proceso de estimación de los parámetros es una característica fuertemente ligada al modelo de la TRI que se use. Primeramente, el modelo elegido determina la representación de la CCR, bien mediante un conjunto de parámetros o mediante los ya calculados de valores de probabilidad de la CCR. En este sentido, si se está usando un modelo paramétrico, el cálculo de los parámetros se puede realizar mediante alguno de los métodos de la máxima verosimilitud o bayesianos mencionados en la sección 3.3.1. En cambio, si se está usando algún método no paramétrico, el cálculo de la CCR se puede realizar mediante el suavizado del núcleo o la variante propuesta por Guzmán (2005), explicados en la misma sección.

Puesto que se está intentando elaborar un modelo genérico y dado que el modelo concreto elegido condiciona tanto la representación de la CCR como el proceso de calibración, esta parte queda fuera del modelo de evaluación que se está formulando. Con esto, el proceso de evaluación es independiente de si se usa un modelo de la TRI u otro, centrándose en modelar qué evidencia es útil para aplicar un método de calibración, lo cual se explica en este apartado; y en utilizar la CCR resultante en la fase posterior de evaluación, tratado en el apartado siguiente.

Como se explicó en la sección 3.3.1, la calibración en sistemas de tests se hace primeramente sin considerar adaptación, seleccionando el siguiente ítem de los disponibles en un banco, con el fin de obtener información suficiente sobre cada uno. De igual manera, para calibrar las restricciones también hay que obtener información sobre el resultado de los alumnos en base a éstas. Sin embargo, la restricción que menos información posee no puede usarse individualmente como criterio para presentar el siguiente problema

a un alumno durante el periodo de recopilación, puesto que ésta no es la única que forma parte de un problema. Por este motivo, es necesario considerar las diferentes restricciones relevantes de un problema para determinar cuál es el más apropiado en cada momento de la recopilación de evidencias. Con esta consideración se proponen tres mecanismos de selección:

- El primero, es el tradicional de selección aleatoria, que elige al azar el problema siguiente sin aplicar otro criterio.
- El segundo método que se propone tiene como fin que el proceso de calibración sea lo más efectivo posible y que recopile el mayor número de evidencias posibles. Para ello, se elige el problema con mayor ratio de restricciones relevantes no presentadas anteriormente. Esto requiere aplicar la función $\rho(r, p)$ de la definición 4.9 para determinar las restricciones relevantes y mantener un conjunto con las que ya se han presentado.
- El tercero es una evolución del segundo y en lugar de mantener un conjunto usaría un contador por cada restricción relevante en un problema con las veces que ésta se ha presentado. Usando el contador se selecciona el problema cuya suma de los valores asociados a las restricciones relevantes sea menor, garantizando en cada momento que se tome el problema que haga que el número de evidencias se distribuya homogéneamente.

En los sistemas de test, una vez que se tiene información suficiente para cada ítem, éste se puede usar de forma directa con alguno de los algoritmos de calibración mencionados anteriormente. Sin embargo, en un EIRP como en el MBR, la evidencia recopilada no se puede usar de forma directa. Esto es así debido a una consideración importante que se deriva de los supuestos, mencionados en la sección 4.3, sobre los que la TRI se aplica. En particular, para que el supuesto de invariancia se mantenga, durante el proceso de calibración, así como en el de evaluación, el conocimiento del alumno se debe mantener constante y no debería haber aprendizaje. En los EIRP bajo el MBR, esta consideración entra en conflicto con la filosofía básica de un tutor MBR, la cual busca la mejora del aprendizaje mediante el uso de refuerzos sobre los errores detectados.

Los tutores MBR permiten a un alumno la realización de varios intentos consecutivos en relación con un problema. Entre cada intento se muestra un refuerzo para hacer que el alumno revise su conocimiento en relación a una restricción violada. Con la realización de un intento consecutivo, tras haberse mostrado el refuerzo, es muy probable que la solución satisfaga la restricción. Esto implica que entre las evidencias recopiladas para una restricción pueda haber aprendizaje debido al refuerzo. El tratamiento de la problemática anterior tiene dos soluciones posibles.

#### 4.5.1.1. Métodos de recolección de evidencias

**Eliminación completa del refuerzo**

Una opción que surge de forma casi inmediata consiste en eliminar la presentación del refuerzo por parte del sistema, evitando así que alumno modifique el conocimiento sobre las restricciones. De esta forma, todas las evidencias recopiladas pueden usarse en el proceso de calibración. Esta solución implica ejecutar el sistema en un modo especial de funcionamiento para recopilar las evidencias que consiste simplemente en captar las

soluciones de los mismos y registrar los errores internamente. Teóricamente, dado que el conocimiento es el mismo, el alumno sólo necesita hacer una vez un problema dado. Este proceso incluso puede realizarse en algunos dominios mediante un examen a papel en el que el alumno resuelve el problema y, posteriormente, la solución es introducida en el sistema MBR para comprobar la violación / satisfacción de las restricciones. Un ejemplo de esta opción se da en el estudio que se explica en la sección 7.7.

**Método de la primera vez relevante**

La segunda opción que se propone tiene como fin el modificar lo menos posible el uso normal del sistema que realicen los alumnos. Con ello se busca que la calibración no se vea sesgada conscientemente o inconscientemente por el alumno. Esta característica está en la línea del principio de incertidumbre de Heisenberg y es parecido a la experimentación a ciegas en la que el alumno no sabe si forma parte de un grupo de control o de experimentación para no influir en su comportamiento. Para ello, el sistema evita mostrar información sobre las restricciones que se han violado pero sí avisa al alumno de si la solución proporcionada contiene algún error o no. De acuerdo con las implementaciones del MBR explicadas en la sección 2.3, esto se corresponde con el nivel más bajo de refuerzo posible.

Dado que todavía existe un refuerzo, aunque este es mínimo el alumno podría todavía tener algún aprendizaje dado que se le permite revisar y modificar elementos de la solución. Esta característica es lo que se conoce como jugar con el sistema, en inglés *gaming the system* (Baker et al., 2010). Es por ello, que en esta opción es necesario algo más que garantice que no hay aprendizaje. Es el método que se ha denominado *de la primera vez relevante*, el cual, como su nombre indica, consiste en considerar la evidencia de una restricción en una serie de intentos sólo la primera vez que la restricción es relevante.

Un ejemplo de cuáles de las evidencias se utilizan de acuerdo al método de la *primera vez relevante* es simplificado en la figura 4.3. Aquí, se pueden ver tres intentos consecutivos $I_{ij}$, donde $i$ representa el problema realizado y $j$ el número del intento. Cada intento tiene una serie de restricciones relevantes $R_k$. Como se puede ver para la restricción $R_2$, ésta es relevante en todos los intentos. Sin embargo, las restricciones $R_3$ y $R_6$ aparecen nuevas en el segundo y tercer intento, lo cual significa que el alumno ha añadido nuevos elementos a la solución. El método que se propone toma sólo los valores asociados a las restricciones que están marcadas con un círculo rojo, dado que éstos se asocian a la primera vez que la restricción es relevante. Este método será explicado posteriormente en la sección 5.2.2 en comparación con algunas consideraciones a tener en cuenta cuando la metodología se va a aplicar en un entorno con un fin formativo, más que sumativo, y con un uso prolongado del sistema.

#### 4.5.1.2. Matriz de rendimiento

En cualquier caso, el modelo que se propone dará por supuesto que, previamente a realizar el proceso de calibración, de acuerdo a alguna de las dos formas anteriores, las evidencias serán recopiladas para cada estudiante respecto a cada restricción. Esta información cumplirá el supuesto de no aprendizaje durante su recopilación y podrá expresarse en lo que llamaremos la matriz de rendimiento, la cual será la entrada para el algoritmo específico que se utilice para la calibración. La formalización de la matriz

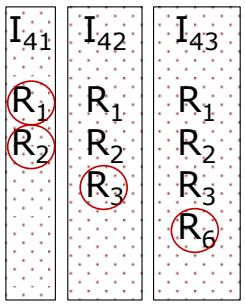



Figura 4.3: Selección de las restricciones la primera vez que son relevantes.

usa la *función de evidencia de una restricción* como base para esta formalización. Esta definición y otras sobre las que se apoyan son dadas en este apartado.

**Definición 4.13** (Conjunto de estudiantes del sistema)**.** *Formalmente, el conjunto de estudiantes que utilizan un sistema MBR vendrá dado por el conjunto* $E = \{e_1, e_2, .., e_m\}$.

**Definición 4.14** (Conjunto de intentos de un estudiante)**.** *Dado un estudiante e, el conjunto de intentos realizados por éste en un sistema MBR se corresponde a las soluciones que el alumno ha desarrollado en cada problema intentado. Supondremos que existe una forma de determinar directamente si una solución ha sido realizada por un estudiante. Este conjunto será representado por la ecuación siguiente, donde m es el número de problemas del dominio y n el número de soluciones de un problema particular. Por comodidad para las definiciones posteriores, y dado que esta característica no supone una gran problemática, supondremos que el conjunto* $I_e = s_{i1}, s_{i2}, .., s_{ik}$ *estará ordenado temporalmente. Es decir, para cualquier par de intentos* $s_{ia}, s_{ib}$ *que pertenecen a* $I_e$*, si* $a > b$ *entonces* $s_{ia}$ *es posterior en el tiempo a* $s_{ib}$.

$$I_e = \bigcup_{i=1}^{m} \bigcup_{j=1}^{n} s_{ij} \text{ tal que } s_{ij} \in \sigma_i \text{ y } s_{ij} \text{ ha sido realizada por } e \tag{4.4}$$

**Definición 4.15** (Función de relevancia general de una restricción)**.** *Dada una restricción r, y un estudiante e, la función de relevancia general determina si la restricción ha sido relevante en alguna solución para el estudiante. Es decir, si hay alguna evidencia de esa restricción para el alumno. La función* $R : E \times \tau \rightarrow \{0, 1\}$ *se define a partir de la función de relevancia de una restricción en una solución (definición 4.8), como sigue:*

$$R(e, r) = \begin{cases} 1 & Si\ \exists p, s_{pi} \text{ tal que } s_{pi} \in I_e \text{ y } \mu(r, p, s_{pi}) = 1 \\ 0 & En\ otro\ caso \end{cases} \tag{4.5}$$

**Definición 4.16** (Función de recolección de evidencias)**.** *A partir de una restricción r y un estudiante e, la función de recolección de evidencia determina mediante alguno de los dos métodos de recolección de evidencias (eliminación completa de refuerzo o primera vez relevante), cuál es la solución que hace a una restricción relevante por primera vez en el tiempo, junto con el problema asociado. Esta función* $\lambda : E \times \tau \rightarrow \alpha \times \phi$ *se*

*define por comprensión de acuerdo a la ecuación 4.6. Para poder aplicarla se supone que* $R(e,r) = 1$ *y que el conjunto de intentos* $I_e$ *esté ordenado por el tiempo de realización. Nótese que esta función es válida tanto para el método de la primera vez relevante, dada su definición, como para el método que elimina el refuerzo, ya que en ese caso no debe haber más de un intento por problema.*

$$\begin{aligned}\lambda(e,r) = \langle s_{pi}, p\rangle \ tal\ que\ \mu(r,p,s_{pi}) = 1 \wedge s_{pi} \in I_e \wedge \\ (\nexists s_{xj}\ tal\ que\ \mu(r,x,s_{xj}) = 1 \wedge j < i)\end{aligned} \tag{4.6}$$

**Definición 4.17** (Función de evidencia de una restricción en un estudiante)**.** *La función de evidencia de una restricción* $r$ *en un estudiante* $e$ *se denota por* $\psi : E \times \tau \rightarrow \{-1, 0, 1\}$*. Esta función proporciona, el resultado de la evidencia que será tenida en cuenta para ese estudiante / restricción, o el valor especial* $-1$ *para indicar la ausencia de evidencia. La definición se realiza en la ecuación 4.7 en base a la función de evaluación de una restricción (definición 4.10) y la función de recolección* $\lambda$*.*

$$\psi(e,r) = \begin{cases} -1 & Si\ R(e,r) = 0 \\ 1 & Si\ \lambda(e,r) = \langle s_{pi}, p\rangle \wedge \delta(r,p,s_{pi}) = 1 \\ 0 & En\ otro\ caso \end{cases} \tag{4.7}$$

**Definición 4.18** (Matriz de rendimiento)**.** *Esta matriz refleja el rendimiento de un alumno durante su uso del sistema en base a las restricciones del dominio. La matriz estará compuesta por* $n$ *filas y* $m$ *columnas, siendo* $m$ *el número de estudiantes y* $n$ *el cardinal del conjunto* $\tau$*. De esta forma, cada elemento* $a_{ij}$ *de la matriz representa el resultado del alumno* $i$ *para la restricción* $j$*, el cual será el resultado de invocar a la función* $\psi(i,j)$*.*

### 4.5.2. Evaluación

Una vez se tienen las CCR de cada restricción del alumno, se pueden utilizar sobre las evidencias recogidas en la matriz de rendimiento para estimar el nivel de conocimiento $\theta$ del alumno. Esta variable, al igual que las curvas de las restricciones, se distribuye dentro del intervalo $[-\infty, \infty]$. No obstante, como se explicó anteriormente, por ser despreciable la probabilidad para valores muy bajos o muy altos del conocimiento, $\theta$ también se suele expresar con algún valor dentro del intervalo $[-4, 4]$, o $[-3, 3]$. La estimación del nivel de conocimiento del estudiante se puede realizar mediante algún método como el de la máxima verosimilitud o el máximo a posteriori, explicados en la sección 3.3.2.2. En este caso, se utiliza una función de verosimilitud que está basada en la matriz de rendimiento y que difiere ligeramente sobre la utilizada en sistemas de tests, como se verá en este apartado.

En teoría, la determinación del conocimiento $\theta$ se puede explicar en dos fases, en una primera fase, se calcula una función de densidad del conocimiento combinando las evidencias recogidas del alumno y en una segunda fase se obtiene el valor más probable de $\theta$ en esa función. Esta función de densidad es la de verosimilitud conjunta y se representa por $P(\theta|R)$, donde $R$ es un patrón de respuestas asociado a los resultados que dan un conjunto de restricciones. Como ya se vio en el apartado 3.3.2.2, esta función es la inversa de la probabilidad condicionada representada en la CCR, puesto que representa la probabilidad del conocimiento dado un conjunto de evidencias. La

definición de la función se realiza tomando como base el supuesto de independencia local según el cual las respuestas asociadas a las restricciones son independientes, lo que permite calcular la expresión anterior como la combinación de la probabilidad de los sucesos independientes.

De acuerdo a lo anterior, la verosimilitud conjunta de un patrón $R$ de resultados de un conjunto de restricciones, es la combinación de las probabilidades de que ocurra cada resultado independiente. Dependiendo del resultado de cada evidencia individual, se usa la función de probabilidad definida por su CCR, si es una evidencia correcta, o la definida por su CCCR, si es incorrecta. La combinación que se realiza consiste en la multiplicación de las probabilidades individuales para un valor de $\theta$ concreto y se realiza sobre todo el rango de valores de $\theta$. Esto es similar a la función de verosimilitud conjunta de tests con ítems cuyas respuestas siguen una distribución de Bernoulli (sólo tienen dos resultados posibles).

La función de verosimilitud debe tener en cuenta la evidencia asociada al resultado de cada una de las restricciones del dominio, que se obtendrá a partir de la matriz de rendimiento. Al igual que en el proceso de calibración, el método de evaluación también requiere que durante el proceso de recolección de evidencias el conocimiento del alumno sea constante. Por este motivo la matriz de rendimiento del alumno se debe formar igual que se hace en la calibración mediante alguno de los métodos de recolección de evidencias que eviten el aprendizaje. Los elementos $a_{ij}$ de la matriz de rendimiento serán usados junto con la función de relevancia general de una restricción (definición 4.15) para formalizar la función de verosimilitud.

**Definición 4.19** (Función de verosimilitud del conocimiento)**.** *A nivel general, la densidad de la variable aleatoria $\theta$ del conocimiento de un estudiante e viene determinada por las evidencias recopiladas para cada intento realizado sobre el conjunto de todos los problemas del dominio (con la consideración de que resulten de aplicar alguno de los métodos de recolección para evitar el aprendizaje). Esta función de densidad es la función de verosimilitud $L(\theta)$ que se puede expresar como $L(\theta) = P_e(\theta|\phi, \tau)$ y se define mediante la ecuación 4.8. El cálculo de la misma combina la CCR o la CCCR de cada restricción $r$ de acuerdo al valor del elemento $a_{er}$ de la matriz de rendimiento. Nótese que para evitar usar el valor -1, indicativo de que no hay evidencia, se utiliza la función de relevancia general $R$.*

$$P_e(\theta|\phi, \tau) = \prod_{r=1}^{n} [P_r(\theta)^{a_{er}} (1 - P_r(\theta))^{1-a_{er}}]^{R(e,r)} \tag{4.8}$$

Como se mencionaba anteriormente, en teoría se puede explicar el cálculo del conocimiento en dos fases: calcular los valores usando la función de verosimilitud, y posteriormente obtener el máximo. En la práctica se aplica alguno de los métodos bayesianos o basados en la máxima verosimilitud, explicados en la sección 3.3.2.2, los cuales buscan el valor de la primera derivada que hace cero la función mediante algún método de cálculo numérico. Estos métodos utilizan estimadores que hacen que el algoritmo converja en las limitaciones de patrones de respuesta con todos los resultados iguales.

La definición 4.19 es válida para los métodos de la máxima verosimilitud pero también es posible utilizarse con los que requieren una distribución inicial. Para ello, se debe considerar una función de densidad de una distribución continua de población de $\theta$ definida por $g(\theta)$. Esta función se añade como parte del producto de la definición, tal y como se explicó en la sección 3.3.1.2. Además, la expresión de la función se puede

modificar aplicando logaritmos con el fin de transformar las potencias en productos y los productos en sumas, haciendo que el coste computacional se vea reducido obteniendo el mismo resultado.

### Ejemplo

Para ilustrar el proceso de cálculo del conocimiento, se muestra un breve y simplificado ejemplo en el que se considera un problema cualquiera $p$ en el que solamente cuatro restricciones son relevantes. Estas cuatro restricciones se representan en la figura 4.4 mediante sus CCR. Los valores de las curvas se han calculado usando el modelo 3PL y, por tanto, cada una tiene asociados 3 parámetros. Así pues, las curvas de la imagen tienen los parámetros siguientes $R_a$: ($a = 0{,}5$; $b = 0{,}1$; y $c = 0{,}2$); $R_b$: ($a = 1$; $b = 2$; $c = 0{,}01$); $R_c$: ($a = 0{,}8$; $b = -2$; $c = 0$); y $_Rd$: ($a = 0{,}7$; $b = 2$; $c = 0{,}3$). Como se puede observar, la restricción $R_a$ tiene una dificultad media; $R_b$ y $R_d$ tienen dificultad relativamente alta; y $R_c$ es de dificultad baja.

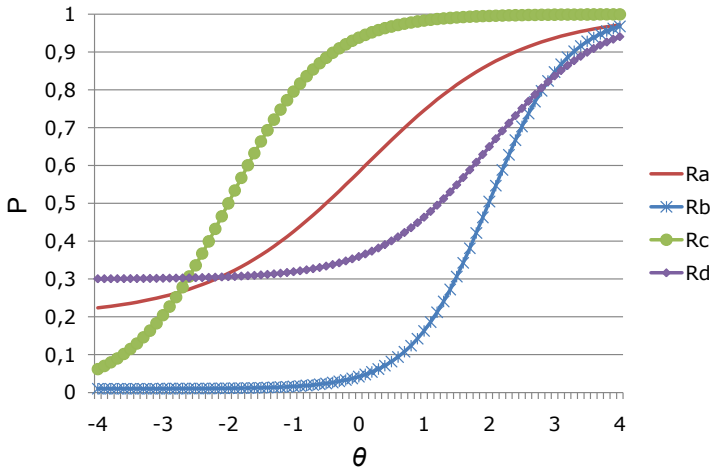



Figura 4.4: Conjunto de varias restricciones representadas por sus CCR.

Por simplicidad se considera el problema del ejemplo es el único del dominio y que hay cuatro alumnos que han lo han intentado. Cada solución tendrá un patrón de respuestas asociado a las cuatro restricciones. Este patrón lo representaremos por la secuencia de evidencias correctas o incorrectas que han dado como resultado a partir de la función de satisfacción sobre cada restricción. Así pues $++++$ es el patrón en el que todas las restricciones son satisfechas, y $+---$ el patrón con la primera restricción satisfecha y el resto violadas. Usando cada combinación de resultados, se aplica la función de verosimilitud sobre las restricciones del ejemplo, dando como resultado las cuatro funciones de densidad del conocimiento de la figura 4.5.

En la figura, la función de densidad del alumno A se corresponde a la resultante de usar el patrón de respuestas $++++$. Esta función tiene el máximo en el límite superior del intervalo $[-4, 4]$. De hecho, el máximo se sale del intervalo y se sitúa siempre en el máximo del intervalo considerado, pues tiende a infinito. La función de densidad del alumno B es el resultado del patrón $----$. Para esta función de densidad, el nivel más probable del conocimiento también se sale del intervalo, pero esta vez por el límite inferior. Estos dos casos, al encontrarse el máximo en el infinito pone de manifiesto la

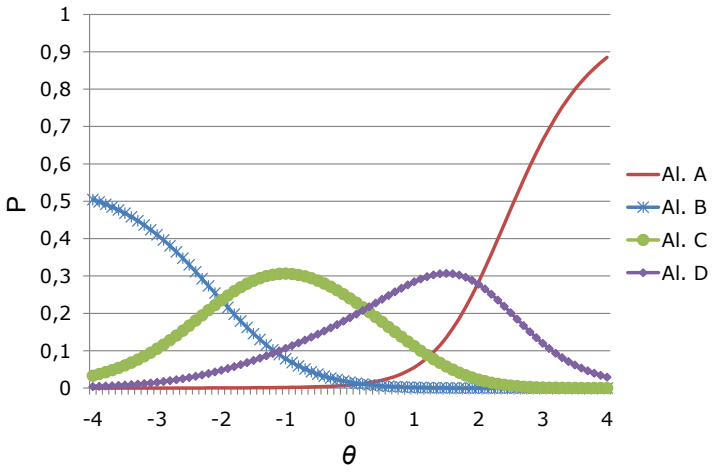



Figura 4.5: Distribuciones del conocimiento en base a diversas combinaciones de la evaluación de las restricciones.

limitación del método de Máxima Verosimilitud. El resultado de los alumnos C y D se corresponden con los patrones $--+-$ y $+-++$ respectivamente y en estos casos se observa claramente como el máximo se sitúa respectivamente en $\theta = -1$ (conocimiento medio / bajo del alumno), y $\theta = 1{,}5$ (conocimiento alto).

## 4.6. Generalización del modelo

El modelo de evaluación presentado en las secciones anteriores es genérico en el sentido de que es independiente del modelo TRI que se aplique y del dominio concreto. En esta sección se intenta dar un paso más allá intentando dar las pautas que permitirían generalizar el modelo anterior no sólo a sistemas MBR, sino a cualquier paradigma de EIRP que cumpla ciertas características.

La generalización del modelo está basada en el modelo teórico definido anteriormente y en el marco de trabajo del DBE explicado en la sección 3.4.2. Según este marco de trabajo se puede calcular el conocimiento de una variable latente usando otras variables observables que se corresponden a evidencias. Esto puede ser aplicado a cualquier tipo de tarea, desde los ítems de los tests, hasta tareas complejas como la resolución de problemas.

En esta sección se explica la parte del modelo genérico que permite realizar la evaluación sumativa del estudiante. La visión general del modelo se describe en la figura 4.6. Aquí se pueden ver dos componentes principales que coinciden con los dos paradigmas que se han combinado en el modelo teórico sobre MBR con la TRI. Por un lado está el EIRP sobre el que se aplicaría el modelo, el cual podría ser cualquiera que permita recopilar evidencias del conocimiento del alumno; y por otro la metodología de evaluación que se implementa con la TRI. A continuación se describe más detalladamente estas componentes y la base sobre la que se asientan.

Con el fin de poder realizar la evaluación es necesario que el paradigma usado para el EIRP cumpla con ciertas características. En primer lugar, éste debe mantener un modelo del estudiante en el que se vayan recopilando evidencias que sean un reflejo

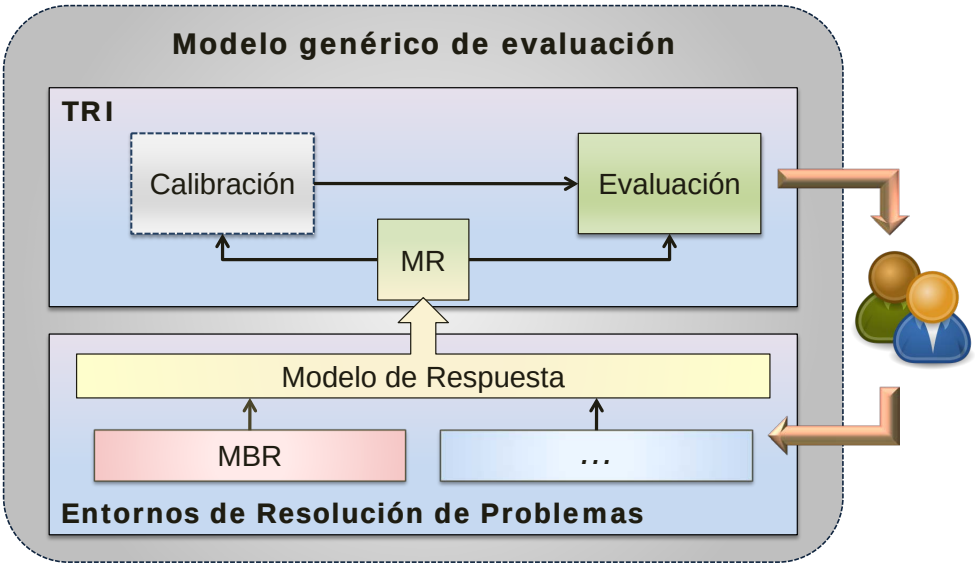



Figura 4.6: Componentes del modelo genérico de evaluación.

del conocimiento del estudiante. Las evidencias que podrían usarse pueden variar en su naturaleza. La forma más simple de evidencia es aquella que refleja el conocimiento de dos maneras: correcto e incorrecto. También podrían ser evidencias más complejas que reflejen el conocimiento mediante un conjunto discreto y finito de estados. En sistemas de test la fuente de evidencia es el ítem y en el MBR es la restricción. En segundo lugar, estas fuentes de evidencias deben cumplir con los supuestos básicos de la TRI para poder ser utilizadas en la inferencia del conocimiento. Así pues, según el principio de independencia local, los sucesos asociados a cada fuente de evidencia deben ser independientes entre sí, de forma que la probabilidad conjunta de dos evidencias pueda calcularse mediante el producto de las probabilidades. Respecto al principio de unidimensionalidad, éste depende del tipo de conocimiento necesario para resolver el problema, siendo recomendable su estudio empírico, al igual que el supuesto de invariancia.

Los estados que tendrían estas evidencias deben ser tenidos en cuenta en un modelo de respuesta similar al desarrollado en este capítulo. Para evidencias con más de dos estados de respuesta, se debería utilizar un modelo de respuesta politómico como el propuesto por Guzmán (2005). De acuerdo a este trabajo, el modelo de respuesta viene definido principalmente por unas curvas características de la respuesta, las cuales modelan la probabilidad de que un alumno seleccione un determinado patrón de respuestas. Dado que la forma más simple estaría dentro del modelo en el que se utilizan patrones de respuesta, hablaremos de forma general de patrones de respuesta para referirnos tanto a evidencias simples como complejas. Aplicando el modelo de respuesta genérico se obtendría el estado resultante de cada evidencia en una matriz de rendimiento del alumno (en la figura se etiqueta la componente como MR).

La matriz de rendimiento, al igual que en el modelo teórico MBR+TRI, recopila el resultado del conjunto total de evidencias que pueden recopilarse de la interacción con cada alumno. En este caso, cada elemento $a_{ij}$ tendrá $N$ estados posibles asociados a los distintos patrones de respuesta que pudieran darse. Los valores de estas restricciones

deben generarse a partir de algún método que garantizase que entre un periodo de recolección de evidencias no hay aprendizaje, con el objetivo de poder realizar una calibración y evaluación válidas.

A partir de los estados de las evidencias, reflejados en la matriz de rendimiento, se puede realizar la calibración de las curvas características. Tanto las curvas como el método de calibración dependerán del modelo de la TRI específico usado para modelar las curvas características. Por este motivo quedan fuera del modelo genérico. No obstante, la utilización de las curvas características es la base para realizar el proceso siguiente de evaluación.

Para realizar la estimación del conocimiento del alumno, se utilizarían las curvas características de los patrones de respuesta obtenidas en la etapa de calibración. Esto se combinaría con las evidencias de la matriz para dar lugar a una función de densidad del conocimiento del alumno, al igual que se presentaba para el modelo MBR+TRI. La expresión de la función de verosimilitud para este caso genérico puede ser expresada mediante la ecuación 4.9

$$P_e(\theta|\overrightarrow{pr_N}) = \prod_{i=1}^{n}[P_{i\overrightarrow{pr_i}}(\overrightarrow{pr_i}|\theta)]^{R(e,i)} \tag{4.9}$$

En esta ecuación $\overrightarrow{pr_i}$ es el patrón de respuestas asociado a la evidencia $i$; $n$ es el número de evidencias posibles; $\overrightarrow{pr_N} = \{pr_1, pr_2, .., pr_n\}$ son los n patrones de respuestas de la fila $e$ de la matriz de rendimiento; $R(e,i)$ es una función que indica si existe la evidencia $i$ para el estudiante $e$; y $P_{i\overrightarrow{pr_i}}(\overrightarrow{pr_i}|\theta)$ es la función de probabilidad de la evidencia $i$ que combina los diferentes patrones de respuesta de una forma u otra dependiendo de si los patrones de respuesta son dependientes o independientes. Siguiendo con la formulación de Guzmán (2005), la expresión en caso de ser independientes combina mediante el producto las curvas de los patrones de respuesta, dando lugar a una expresión similar a la de la función de verosimilitud dada en la definición 4.19, mientras que si fueran dependientes se hace sumando las curvas de los patrones.

Esta propuesta genérica, es un caso particular del DBE donde el modelo de presentación sería la interfaz del EIRP, la cual se asume es mediante el ordenador; y el modelo de evidencias y del estudiante se basa en el uso de la TRI sobre las evidencias observables, en la forma descrita anteriormente. En la sección 3.4.2.2 ya se mostró la equivalencia a grandes rasgos entre los modelos del DBE y el MBR. La generalización propuesta en esta sección se basa en el mismo patrón, pero usando la TRI en lugar de los heurísticos del MBR. En la sección 5.5 se especifica concretamente la correspondencia de los elementos del DBE con la metodología propuesta, ya con los elementos que permiten realizar la evaluación formativa.

## 4.7. Conclusiones del capítulo

En este capítulo se propone un modelo teórico de evaluación que permite realizar la evaluación sumativa de un alumno en EIRP. Debido al tipo de sistemas donde se pretende aplicar necesita de una técnica de modelado del alumno y de una metodología de evaluación. Como paradigma en EIRP se utiliza el MBR dado su eficiencia probada a lo largo de multitud de estudios y principalmente debido a la simplificación de uso frente a otras técnicas. En cuanto a la metodología de evaluación utiliza la TRI puesto que es bien fundamentada y posee diversas ventajas sobre otros enfoques.

El modelo intenta cumplir con los objetivos planteados en el capítulo 1 intentando ser un mecanismo sistemático y formal, característica que se obtiene de la naturaleza objetiva y bien fundamentada de la TRI. A la vez, busca ser un modelo cuantitativo en el que se dejen fuera de la evaluación elementos subjetivos propios de modelos cualitativos. También busca la genericidad a varios niveles: a) sin imponer un modelo específico de la TRI, dando las definiciones y pautas necesarias que lo hacen independiente del modelo TRI; b) aplicable a cualquier dominio, lo cual es inherente al paradigma del MBR; y c) en la medida de lo posible, dando unas pautas sobre la generalización del modelo sobre EIRP en general. Por último, el modelo desarrollado permite realizar una evaluación sumativa. Según lo explicado en la sección 4.1, este es el paso previo a diseñar una metodología de evaluación formativa o para el aprendizaje.

La aplicación de la TRI en los sistemas MBR es posible gracias a una analogía que se ha encontrado entre los ítems y las restricciones. Ambos son instrumentos de medida del conocimiento que, aunque en distintos entornos, reflejan evidencias correctas o incorrectas del conocimiento. Además, los entornos donde se aplican (los sistemas de tests para los ítems y los tutores MBR para las restricciones), mantienen una similitud estructural que favorece el uso de la metodología formal de evaluación en el paradigma de construcción de EIRP.

Para la definición formal del modelo de evaluación se parte de la definición del modelo de respuesta típico de los sistemas de tests (Guzmán, 2005) y se intenta extender con las particularidades de los sistemas MBR. Este modelo típico se asienta sobre una función de evaluación de los ítems que determina si éstos son correctos o no. De forma similar, se define una función de evaluación de las restricciones. Pero para ello, es necesario considerar previamente las diferencias que se introducen con sistemas MBR: los elementos principales del sistema que son las restricciones y los problemas del modelo de dominio, en lugar de los ítems y los tests; la respuesta, que viene representada por una solución a un problema a diferencia de las opciones en los ítems de los tests; la particularidad de que las restricciones pueden ser o no relevantes, de acuerdo al formalismo presentado en la sección 2.3.2; y la determinación de si una restricción es correcta, la cual se evalúa a partir de la condición de satisfacción en lugar de comprobar la correspondencia a un patrón de respuesta u otro. El formalismo del modelo de respuesta en sistemas MBR se completa con la definición de curvas que modelan la probabilidad de que una restricción sea una evidencia correcta (CCR) o incorrecta (CCCR).

Como parte de la metodología de evaluación es necesario realizar un proceso de estimación de las curvas de las restricciones. Esta etapa de calibración requiere una consideración especial relacionada con el supuesto de invariancia de la TRI, la cual supone que no hay aprendizaje durante el proceso de recolección de las evidencias. Esto entra en conflicto con la filosofía central del MBR que intenta hacer que el alumno aprenda mediante la presentación de un refuerzo. Para tratar esta situación se proponen dos métodos de recolección evidencias para realizar la calibración: bien eliminando el refuerzo directamente, o utilizando solamente la primera vez que las restricciones son relevantes, dejando cualquier efecto del aprendizaje mediante el refuerzo fuera del conjunto de evidencias. Ambos métodos tendrán como resultado un conjunto de evidencias que se reflejarán en una *matriz de rendimiento*, la cual ha sido definida formalmente a partir de una serie de funciones que se basan en el modelo de respuesta anterior. La matriz tendrá el valor resultado por cada estudiante (fila) / restricción (columna), y será utilizada por el método de calibración específico. Este método, así como el modelo

de la TRI utilizado para modelar la CCR, los cuales están directamente relacionados, queda fuera del modelo de evaluación con el objetivo de mantener la genericidad.

La parte final del modelo de evaluación consiste en la estimación del conocimiento del alumno mediante alguno de los métodos existentes, como los basados en la función de verosimilitud o los bayesianos, mencionados en la sección 3.3.2.2. Como base para estos métodos se ha formalizado la función de verosimilitud. Esta función es una función de densidad del conocimiento a partir de la cual se determina el valor más probable del conocimiento del alumno mediante algún método numérico de maximización. La función de verosimilitud se define a partir de la combinación de las funciones de probabilidad de las restricciones que proporcionan evidencias sobre el conocimiento. Como en cualquier modelo de evaluación de la TRI, la combinación consiste en usar la probabilidad de la evidencia correspondiente. Es decir, la CCR, si es una evidencia correcta, o la CCCR, si es incorrecta. La expresión de la función se realiza en base a la función de relevancia general de una restricción y al contenido de la matriz de rendimiento, ambas definidas en la sección 4.5.1.2. La matriz, al igual que en la etapa de calibración, requiere dejar fuera el aprendizaje que pueda haber en el sistema, por lo que se puede usar la misma definición de la matriz dada en la etapa anterior. Puesto que el método de cálculo a emplear es específico, se deja también fuera del modelo para hacerlo genérico. No obstante la función de verosimilitud sirve como base para ese método, sin apenas modificar su forma.

Por último, se intenta generalizar, en la medida de lo posible, la aplicación del modelo de evaluación sumativa a cualquier EIRP, independientemente del paradigma usado para modelar al alumno. La generalización está basada, como en el marco de trabajo DBE, en la evidencia que se puede recopilar del alumno y que serviría de instrumento para medir el conocimiento, siempre y cuando cumplan los supuestos de la TRI para aplicarse. Esta generalización se centra en definir una matriz de rendimiento que será proporcionada por el paradigma del modelado del alumno en cuestión. Esta matriz será usada en la calibración y en la evaluación. El primer proceso, al depender de un modelo de la TRI y de un método de calibración específicos, está fuera del modelo. La evaluación en cambio, se realizaría en base a las curvas características y a la matriz de rendimiento, también generando una función de verosimilitud que se utilizaría para estimar el conocimiento del alumno del mismo modo que se realiza para el MBR + TRI.

En muchos de los artículos y estudios existentes en el MBR se menciona que las restricciones representan conocimiento declarativo (Mitrovic, 2012), pero esto no quiere decir que la evaluación que se realiza, al utilizar restricciones, es una evaluación de este tipo de conocimiento. Aunque las restricciones se asocian generalmente a conocimiento declarativo, ni la solución del alumno es el resultado de aplicar únicamente conocimiento declarativo, puesto que el conocimiento procedimental también es necesario para resolver los problemas en entornos procedimentales; ni los principios del dominio están siempre asociados a conceptos, como se menciona en (Mitrovic y Ohlsson, 2006) y que también queda patente tras el uso de las *restricciones camino* mencionadas en el apartado 2.3.2, las cuales representan conocimiento procedimental.

De acuerdo a lo mencionado en el párrafo anterior, el resultado de aplicar las restricciones, bien satisfacción o violación, no es el reflejo exclusivo del conocimiento declarativo, ni del procedimental, sino que es el resultado de una mezcla de ambos. De hecho, todavía, la relación y la dependencia existente entre ambos tipos de conocimiento continúa siendo una cuestión sin resolver (Schneider y Stern, 2010). De esta forma,

las restricciones son un instrumento de medida del conocimiento en una dimensión general (de Jong y Ferguson-Hessler, 1996), sin poder distinguir hasta qué punto miden declarativo o procedimental. Además, tal y como mencionan Mitrovic y Ohlsson (2007), el aprendizaje, medido en términos de restricciones, sigue un patrón similar a si se hace en términos de reglas de producción, propias de los sistemas tutores cognitivos, más cercanos al conocimiento procedimental.

# Capítulo 5

# Aplicación del modelo para evaluación formativa

*El que enseña el camino al que va errado,*
*luz en su luz le enciende y a él le alumbra*
*habiéndola comunicado*

Marco Tulio Cicerón (106 a.C. - 43 a.C.)

**Resumen:** En este capítulo se describe cómo aplicar el modelo teórico de evaluación tanto a sistemas de tests como a sistemas MBR y cuáles son las características asociadas a la evaluación formativa.

Tras haber expuesto en el capítulo anterior el modelo teórico que permite realizar una evaluación sumativa, es necesario ver cómo extender este modelo con las características necesarias para proporcionar evaluación formativa que ayude en la mejora del aprendizaje. Como se comentaba en el apartado 4.1, la evaluación sumativa se aplicará durante todo el proceso, dentro del cual distinguiremos aquella que se realiza al final para determinar si el alumno cumple con cierto nivel u otro tipo de objetivos del aprendizaje. Nos referiremos a esta evaluación como *evaluación sumativa final* para distinguirla de la que se realiza como parte del proceso formativo.

Según Scriven (1967); Ramaprasad (1983); Sadler (1989); Black y Wiliam (1998); Taras (2005), además del juicio del alumno, que se corresponde con nuestra evaluación sumativa, debe haber un refuerzo que indique las carencias existentes entre el nivel actual del elemento objeto de evaluación y el objetivo de aprendizaje que será evaluado en la evaluación sumativa final. En base a esto, consideraremos, que aquel refuerzo que advierte al alumno sobre lo que debe mejorar para alcanzar el objetivo de aprendizaje es parte de la capacidad formativa del sistema. De forma muy general, cualquier uso del modelo de evaluación anterior con fines formativos, puede considerarse una forma de evaluación formativa en la que el objetivo de aprendizaje, de cara a una evaluación sumativa final, puede diferir ligeramente pero sigue el mismo propósito de hacer que el alumno mejore respecto a ese objetivo.

El MBR, como paradigma dentro de los STI, cuenta con mecanismos instructivos para hacer que el alumno aprenda y que, de forma indirecta suponen también parte del carácter formativo. En primer lugar está el refuerzo proporcionado por el sistema ante

los fallos del alumno mientras resuelve un problema. Este refuerzo no se presenta sobre la evaluación sumativa del capítulo anterior, sino que está asociada a una evaluación más específica que realiza el sistema sobre las soluciones para determinar su corrección. Aunque a un nivel muy bajo de genericidad, ya que se proporciona sobre las restricciones, esta forma de refuerzo es la componente más básica del MBR que proporciona información sobre los elementos que el alumno debe mejorar y por tanto se considera una componente de evaluación formativa. Los objetivos de aprendizaje que sigue el refuerzo en el MBR no están definidos a priori en una evaluación sumativa final, pero buscan el aprendizaje del alumno usando los errores concretos del mismo.

En su forma nativa, el MBR aplica una estrategia de adaptación a las necesidades del alumno que está basada en la estimación heurística del conocimiento del alumno. Esta adaptación es otra forma de evaluación formativa. Esto es así puesto que se utiliza un juicio emitido por el sistema sobre lo que el alumno sabe para guiar la formación. El objetivo de aprendizaje depende de la estrategia pedagógica particular utilizada por el sistema y puede perseguir el dominio de ciertos elementos de aprendizaje prefijados, como en la evaluación formativa descrita por los autores anteriormente mencionados. Otra alternativa que no tiene en cuenta objetivos concretos se basa en corregir los problemas más frecuentes del alumno o hacer que éste sepa el máximo posible del dominio.

En el modelo que se propone, se reemplaza la evaluación heurística por la evaluación sumativa del capítulo anterior. En base a esta estimación del conocimiento, primero, se proporciona la formación adecuada a las necesidades particulares, de cara a cubrir ciertos objetivos de aprendizaje de alguna estrategia instructiva. Los elementos que extienden la evaluación sumativa para guiar al alumno en las acciones que debe realizar para mejorar y llegar a esos objetivos antes de una evaluación sumativa final son parte de la evaluación formativa que se propone.

Este capítulo estudia los diferentes elementos que permiten utilizar la TRI en los sistemas MBR con fines formativos. Para ello, en la siguiente sección se explica la forma de aplicar el modelo a sistemas de test y las implicaciones de usar esta combinación para calibrar, evaluar y aplicar estrategias adaptativas para la formación. La sección 5.2 continúa con la aplicación del modelo a sistemas MBR, explicándose los cambios propuestos sobre la estructura básica de cualquier tutor MBR y un nuevo método para seguir la evolución del conocimiento del alumno. Seguidamente, en la sección 5.3, se expone la temática central de este capítulo mediante los diferentes componentes y mecanismos TRI que permiten la formación en sistemas MBR. Posteriormente, en la sección 5.4, se mencionan otras utilidades que proporcionan los mecanismos de la TRI en sistemas MBR. Finalmente, en la sección 5.5 se presentan las conclusiones del capítulo.

## 5.1. Modelo aplicado a sistemas de tests

De cara a estudiar la selección adaptativa de problemas como elemento base que proporciona capacidades formativas al MBR, se utilizará la aplicación práctica del modelo teórico más cercana al uso original de la TRI, la cual se encuentra en sistemas de tests. Con la incorporación del modelo anterior para este tipo de sistemas se pretende un doble objetivo: en primer lugar extender estos sistemas para cubrir su limitación principal a la hora de realizar la evaluación en entornos procedimentales; en segundo lugar, y de forma paralela, estudiar en su forma más natural cómo se puede realizar la

selección adaptativa. Por tanto, esta sección detalla las consideraciones que hay tener en cuenta para aplicar el modelo teórico formalizado en el capítulo anterior dentro de TAI y cómo el mecanismo de selección propio de los tests serviría para seleccionar problemas MBR.

El uso de la analogía presentada anteriormente en la figura 4.1 para aplicar el modelo de evaluación en este tipo de sistemas requiere de tener en cuenta una característica que surge de la equivalencia establecida entre el problema MBR y los tests. En sistemas de tests los mecanismos de adaptación se aplican seleccionando el siguiente ítem más apropiado para el alumno. De acuerdo con esta forma de adaptación, y en base a la analogía, la TRI permitiría seleccionar las restricciones más apropiadas, pero esto no tiene sentido en MBR, ya que las restricciones son inherentes a un problema y no se pueden separar. Es por ello que la adaptación debe hacerse seleccionando el siguiente problema más apropiado, que considerando la analogía en sistemas de tests corresponde a un test.

Con el fin de tratar este problema se propone como solución el rediseño de la analogía inicial mediante la introducción en los sistemas de tests de un nuevo tipo de ítems especial que permita establecer una nueva equivalencia con los problemas MBR. De esta forma los mecanismos de selección de ítems de la TRI pueden aplicarse a estos nuevas ítems y, por tanto, a los problemas. La consideración de especial se encuentra en que para mantener la analogía, este nuevo tipo de ítems requiere de tener sub-ítems que puedan corresponderse con las restricciones. Es por ello que se han denominado *ítems compuestos*, los cuales son similares a los testlets (Rosenbaum, 1988) en naturaleza pero, como veremos, presentan diferencias en la formulación asociada.

**Definición 5.1** (Ítem Compuesto)**.** *Un ítem compuesto, como su propio nombre indica es una agrupación de elementos asociados a diferentes evidencias. Éste puede ser visto como una caja negra donde diversas evidencias, recogidas de la interacción con el alumno, son agrupadas. Esta agrupación puede servir para realizar una evaluación conjunta, combinando las diversas componentes. En la práctica, un ítem compuesto puede ser modelado como un conjunto de ítems componente, como si de un testlet se tratara. En esta particularización, el ítem compuesto $i$ se representa por $C_i = \overrightarrow{U_i} = \{U_{i1}, U_{i2}, .., U_{in}\}$, donde cada $U_{ij}$ es un ítem componente de $i$.*

La forma de modelar este nuevo tipo de ítems, mediante varios componentes, permite usar como fuente de evidencia tanto ítems que se preguntan directamente al alumno como tareas complejas. Dependiendo de la naturaleza de la evidencia los ítems componentes pueden ser referidos de una manera u otra. Así pues, en el caso de que éstos estén asociados a preguntas tangibles y directas se usa el término ítem componente *real*; mientras que si están asociados a fuentes de evidencia que no son preguntas directas, provenientes de tareas complejas, se habla de ítem componente *virtual*. Por simplicidad y claridad se usará sólo el término "ítem componente", siendo identificable por el contexto a qué tipo corresponde.

A partir de los ítems compuestos se define la nueva analogía representada en la figura 5.1. Aquí, la equivalencia entre la restricción y el ítem se mantiene, dado que es la base para formular el modelo teórico anterior. En un nivel más general, un problema MBR se corresponde con un ítem compuesto, donde las restricciones relevantes al problema asumirían el rol de sub-ítems. La agrupación de problemas realizados en una sesión de trabajo de un tutor MBR es similar a los ítems realizados en un test.

El hecho de modelar componentes que, apareciendo en otros ítems compuestos, sean

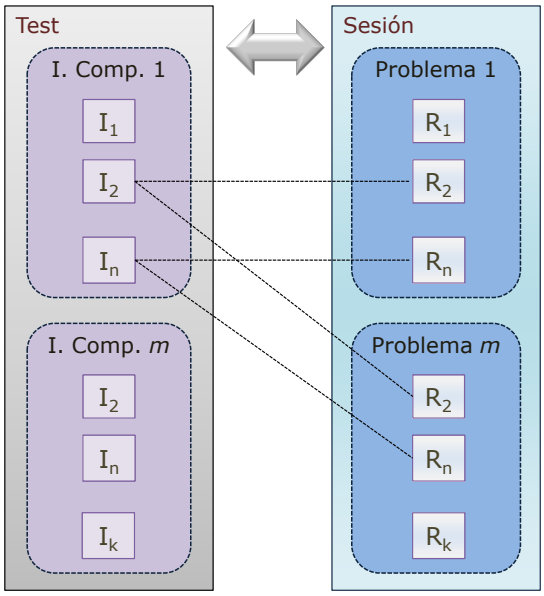



Figura 5.1: Equivalencia refinada de tutores MBR en sistemas de tests.

un mismo ítem influye en el modelo y, consecuentemente, en la forma de implementarse. Normalmente en los sistemas de tests existen ítems llamados *generativos* o *plantillas* (en inglés *items models*) (Bejar et al., 2002) que, aunque con valores diferentes, son isomórficos y representan el mismo ítem. Al conjunto de todos los ítems isomórficos entre sí se denomina familia de ítems. Existen tres modelos típicos para tratar con diversos niveles de isomorfismo (Sinharay y Johnson, 2005):

- *Modelo de parientes no relacionados* (en inglés *Unrelated Siblings Model*): Es aquel que considera cada ítem de la familia como totalmente independientes, tanto para la calibración como para la evaluación.
- *Modelo de parientes relacionados* (en inglés *Related Siblings Model*): Considera que pueda haber una relación entre cada ítem de la familia y calcula esta relación una vez calibrados los parámetros, la cual es usada también en la evaluación para atenuar una sobreestimación causada por la relación existente.
- *Modelo de parientes idénticos* (en inglés *Identical Siblings Model*): En este caso se considera cada elemento de la familia como si fuera el mismo. La calibración se hace sobre la familia en sí y la evaluación usa esa curva con cada evidencia.

En el modelo que estamos formulando es más apropiado considerar el modelo de parientes idénticos, puesto que estamos intentando modelar la misma evidencia pero en diversos ítems compuestos: un ítem componente es el mismo, independientemente de dónde esté contenido.

### 5.1.1. Calibración y evaluación

En cuanto a la calibración, ésta no se realiza a nivel de ítem compuesto sino a nivel de ítem componente, de forma independiente. Con esta consideración se puede definir

la *Curva Característica de un Ítem Compuesto* (CCIC). Esta curva es equivalente a la de un ítem de opción múltiple con respuestas independientes, con la diferencia de que los componentes no forman un espacio probabilístico. Aunque este nuevo tipo de ítems es básicamente un testlet, la expresión que determina su curva difiere a éstos últimos. En los testlets la curva se determina como la suma de las CCI porque la finalidad de esa curva es transformar la escala de $\theta$ a puntuación verdadera. En este caso, la curva se va a utilizar como si fuera la de un ítem normal para selección, por lo que debería tener como dominio, el de una función de probabilidad, es decir, el intervalo $[0, 1]$. Es por ello que la CCIC se define como sigue:

**Definición 5.2** (Curva Característica de un Ítem Compuesto)**.** *La CCIC representa la probabilidad de que un alumno con rasgo latente estimado $\theta$ responda correctamente un ítem compuesto $C_i = \overrightarrow{U_i}$. Esta curva, al igual que la CCI, es también monotónica creciente y se basa en la probabilidad de responder correctamente cada uno de los componentes internos. De forma genérica, las evidencias del ítem compuesto tendrán una curva característica que vendrá modelada por la CCI de ítems componentes. La función de probabilidad asociada a la CCIC, $P_i(\theta) = P(C_i|\theta) = P(\overrightarrow{U_i}|\theta)$, vendrá definida por la ecuación 5.1 utilizando la CCI de cada ítem componente ($U_{ij}$).*

$$P(\overrightarrow{U_i}|\theta) = \prod_{j=1}^{n} P_j(U_{ij}|\theta) \tag{5.1}$$

La curva de un ítem compuesto es un elemento de carácter simbólico, ya que no es necesario distinguirse como tal a la hora de usar las evidencias en el proceso de calibración. De la misma forma, para la evaluación tampoco se necesita utilizar el ítem compuesto, ya que las evidencias existentes se corresponden a los ítems componentes. Realmente, evaluar al alumno en un ítem compuesto es equivalente a evaluar al alumno en un test en el que los ítems involucrados son los componentes del ítem compuesto. La evaluación del alumno en varios ítems compuestos se reduce a utilizar sus componentes para la estimación.

### 5.1.2. Selección adaptativa

Como se mencionaba al principio de este capítulo, a la hora de aplicar la selección al modelo teórico de evaluación propuesto, habrá diversas estrategias posibles de selección. El objetivo será seleccionar el ítem compuesto que, según la estrategia seleccionada sea el más apropiado. Para los mecanismos básicos de selección mencionados en la sección 3.3.2.3 se explica cómo afecta el formalismo de los ítems compuestos en esta tarea. Puesto que esta parte es equivalente a la selección de problemas, en la sección 5.3.3 se continúa, dando nuevas estrategias.

#### 5.1.2.1. Criterio de máxima información

Si se utiliza este criterio habrá que seleccionar el ítem compuesto no seleccionado previamente que maximice la información proporcionada sobre el conocimiento del alumno $\theta$. En realidad, la información que proporciona un ítem compuesto es la proporcionada por los diversos componentes involucrados, siendo ésta equivalente a la que proporciona un test con los mismos ítems que los componentes. Es decir, la información de un ítem compuesto es la suma de la información proporcionada por cada componente. Formalmente, se expresa esta función mediante la siguiente definición.

**Definición 5.3** (Función de información de un ítem compuesto)**.** *Dado un nivel estimado del conocimiento $\theta$ para un estudiante concreto e, la función de información de un ítem compuesto $\overrightarrow{U_i}$ se calcula en base a cada uno de sus ítems componente. Para ello se aplica la función de información dada en la ecuación 3.8, que denotaremos mediante $I_{ij}$ para destacar su aplicación sobre un ítem componente $j$. En base a estos elementos, la función de información sobre el ítem compuesto, $I_i$, se define como sigue:*

$$\vec{I_i}(\theta) = \sum_{j=1}^{n} I_{ij}(\theta) \tag{5.2}$$

Para seleccionar uno de entre varios ítems compuestos se calculará la función de información de cada uno $\vec{I_i}(\theta)$ mediante la ecuación 5.2 y se seleccionará aquella que sea máxima para el valor $\theta$ de conocimiento estimado del alumno. Este cálculo se debe realizar cada vez que se vaya a seleccionar un ítem compuesto, pues la estimación del conocimiento del alumno puede variar y la aparición de ítems previos condicionará la información proporcionada por cada ítem.

#### 5.1.2.2. Método bayesiano de la máxima precisión esperada

En este método, que consiste en minimizar la esperanza de la varianza de la distribución del conocimiento del alumno a posteriori, hay que tener en cuenta la misma consideración que en el método anterior. Suponiendo que el conocimiento del alumno es $\theta$, la fórmula original de la función a evaluar puede reescribirse mediante la ecuación 5.3.

$$E_i(\sigma^2(\theta|\overrightarrow{U_i})) = \sum_{j=0}^{1} \sigma^2(\theta|\overrightarrow{U_i}) \int \widetilde{P}(\overrightarrow{U_i}|\theta)^j \widetilde{Q}(\overrightarrow{U_i}|\theta)^{1-j} g(\theta) d\theta \tag{5.3}$$

La expresión es similar a la original con la salvedad de que se particulariza para el ítem compuesto $\overrightarrow{U_i}$. En la fórmula, $g(\theta)$ es la distribución inicial del conocimiento; $\widetilde{P}(\overrightarrow{U_i}|\theta)$ y $\widetilde{Q}(\overrightarrow{U_i}|\theta)$ son la CCIC y la inversa de la curva, respectivamente; $E_i$ es la esperanza del ítem $i$; y $\sigma^2$ es la varianza.

#### 5.1.2.3. Selección basada en la dificultad

Este método se basa en presentar el ítem compuesto de entre aquellos que no se han seleccionado antes y que tienen un nivel de dificultad más próximo al nivel actual del estudiante, calculado mediante el método de evaluación sumativa del capítulo anterior. No obstante, dado que la CCIC es realmente una agrupación de curvas y que no hay un parámetro $b$ como tal de dificultad, su cálculo requiere de aplicar métodos numéricos para hallar su valor. A continuación se mencionan tres métodos para este cálculo.

- El primer método implica usar la definición de dificultad. De acuerdo a ésta, la dificultad es el valor del conocimiento que divide la curva en la misma probabilidad de acertar que de fallar. Esta probabilidad $p_{dif}$ se puede determinar usando el valor máximo y el mínimo sobre la función de probabilidad para determinar la mitad de este valor. La suma del valor medio y el mínimo de la función es la probabilidad buscada. Este cálculo se simplifica a aplicar la ecuación siguiente:

$$p_{dif} = \frac{Min(P(\overrightarrow{I_i}|\theta)) + Max(P(\overrightarrow{I_i}|\theta))}{2} \tag{5.4}$$

Aquí, $P(\overrightarrow{I_i}|\theta)$ es la CCIC, es decir, la multiplicación de cada CCI componente; $Max$ es la función máximo y $Min$ la función mínimo. Una vez calculado este valor, la dificultad del problema es el valor de $\theta$ que hace que la CCIC tome como valor de probabilidad $p_{dif}$. Es decir $P(\overrightarrow{I_i}|\theta) = p_{dif}$. La aproximación del valor máximo y mínimo de la función es muy sencilla pues bastaría con calcular el producto de las CCI en un valor de $\theta$ muy alto y muy bajo, respectivamente. Dado que el mínimo y el máximo se encuentran en el límite y que no se está buscando dónde se localiza el máximo o el mínimo, sino que se necesita su valor, el empleo de un $\theta$ muy alto o muy bajo da una buena aproximación de los valores buscados. Una vez determinado $p_{dif}$, es necesario aplicar algún método numérico como el de Newton-Raphson para determinar el punto donde la función toma este valor. Este cálculo no será muy costoso dado que el rango de valores donde se encuentra suele localizarse dentro del intervalo $[-4, 4]$.

- Otra forma de determinar la dificultad es usando la función de información del ítem compuesto dada por la definición 5.3. Por las propiedades de esta función, tal y como se mencionaba en la sección 3.2.2, su máximo se sitúa muy próximo al nivel de dificultad del ítem, siendo éste una buena aproximación, sólo si el parámetros $c_i$ es próximo a cero. Este prerrequisito se puede determinar viendo el valor del mínimo de la función en un valor muy bajo de $\theta$, que sería aproximadamente el valor de $c_i$. Esta forma de determinar la dificultad se reduce a hallar el máximo de su función de información, lo cual también requiere del uso de métodos numéricos.

- El último método implica usar la definición del parámetro de discriminación. Según ésta, el punto de inflexión de la curva es el que da el valor a $b_i$. Por lo que, para determinar la dificultad habría que usar algún método numérico para determinar el valor de $\theta$ donde se encuentra este punto.

En cualquier caso, el valor de la dificultad se puede determinar fácilmente si se discretizan los valores que $\theta$ puede tomar y se considera un intervalo cerrado para esta variable. Normalmente $[-4, 4]$ es el que se suele usar dado que fuera de éste, la probabilidad no suele variar respecto del límite más cercano del intervalo.

Además de las consideraciones anteriores sobre los ítems compuestos, la extensión de sistemas de tests para poder evaluar al alumno en entornos procedimentales requiere de combinar los mecanismos de la TRI comentados anteriormente con la presentación del problema. Es decir, cuando se selecciona un ítem compuesto para ser evaluado, bajo la suposición de que éste está asociado a un problema de un EIRP, es necesario algún tipo de comunicación o integración entre el sistema de tests y el EIRP que permita presentar al alumno el problema adecuado en la interfaz para la recolección de evidencias, y que tras finalizar, se informe al sistema de tests sobre el resultado. Esto es un aspecto más cercano a la implementación particular que se realice y dependerá de las características de los sistemas involucrados.

El uso de un sistema de tests para evaluar implica que durante el proceso de evaluación no debe haber aprendizaje o que hay algún método para obviarlo, como los mencionados en la sección 4.5.1.1. Dado que los mecanismos son los mismos en sistemas de tests y en MBR con la TRI, pasaremos a explicar estos últimos en la siguiente sección, donde puede resultar más intuitivo puesto que su funcionamiento es el principal motivo de tener que realizar estas consideraciones.

## 5.2. Aplicación del modelo a sistemas MBR

El resto de características asociadas a la evaluación formativa se estudiará sobre sistemas MBR que incorporan el modelo de evaluación sumativa. La forma más natural de aplicar el modelo teórico del capítulo anterior a estos sistemas consiste en, partiendo de los elementos originales de la técnica de modelado, extender con los elementos requeridos para que se puedan aplicar los mecanismos de la TRI. En esta sección se presenta esta extensión, basándose de la equivalencia entre ítems y restricciones, y las consideraciones a tener en cuenta para proporcionar evaluación formativa al estudiante.

### 5.2.1. Estructura MBR extendida

Las partes básicas que componen la estructura del MBR fueron explicadas en detalle en la sección 2.3.3. De éstas, las que están directamente relacionadas con la formación del alumno son el modelo del alumno y del dominio, como base para dirigir la instrucción; y el módulo pedagógico, que se encarga de realizar la estrategia instructiva del sistema en base a estos modelos. El resto de componentes recaen sobre estas partes pero no están involucradas directamente en la toma de decisiones o el funcionamiento interno. Con el fin de aplicar el modelo de evaluación MBR + TRI en estos sistemas para mejorar las capacidades formativas del mismo, es necesario extender estar partes. La figura 5.2 muestra los nuevos elementos asociados a la TRI que extienden la estructura básica.

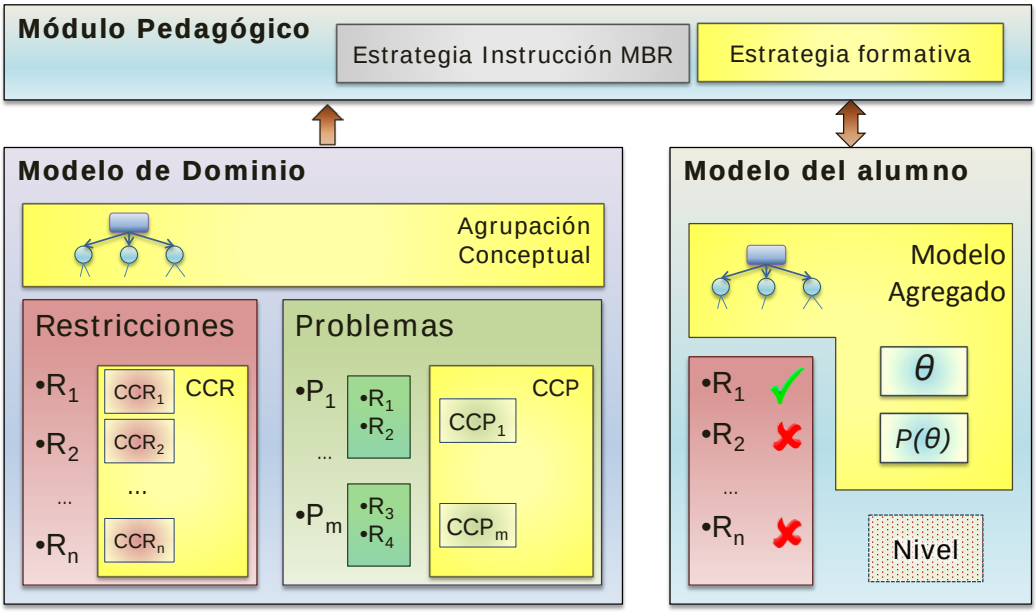



Figura 5.2: Estructura extendida del MBR con los elementos de la TRI.

El **modelo del dominio** es una pieza clave del paradigma MBR puesto que toda su formulación parte de él. Tal y como se explicó en el apartado 2.3.3.1, este modelo está compuesto por dos elementos importantes. Las restricciones que modelan los principios que ninguna solución puede violar, y los problemas que se pueden plantear al alumno. Como ya se ha mencionado, las restricciones son equivalentes a ítems cuya respuesta

sólo tiene dos resultados posibles: correcto o incorrecto. De igual forma que los ítems tienen una CCI las restricciones básicas deben ser extendidas con una estimación de su CCR, la cual vendrá definida por los parámetros apropiados, si se utilizan modelos paramétricos, o por los valores correspondientes, en otro caso.

Los problemas también son extendidos con una estimación sobre la probabilidad de resolverlo correctamente. Esta estimación viene representada por la *Curva Característica de un Problema* (CCP), la cual es un caso particular de la CCIC presentada en la sección 5.1.1. En el caso de la CCP las evidencias son los resultados de las restricciones y la agrupación en sí viene definida por el conjunto de restricciones relevantes en el problema. Esta curva se calcula como el producto de las CCR en las restricciones relevantes. Formalmente, se puede extender el modelo del capítulo anterior con la definición siguiente.

**Definición 5.4** (Curva Característica de un Problema)**.** *La CCP representa la probabilidad de que un alumno con rasgo latente estimado $\theta$ resuelva correctamente un problema $p$. Esta probabilidad es equivalente a una CCIC y, por tanto, es una curva monotónica creciente que se basa en la probabilidad de satisfacer cada una de restricciones relevantes en el problema. La función de probabilidad $P_p(\theta) = P(p|\theta)$ vendrá definida por la ecuación 5.5 utilizando la CCR de cada restricción ($P(i|\theta)$) y la función de relevancia de una restricción en un problema ($\rho(i,p)$).*

$$P(p|\theta) = \prod_{r=1}^{n} [P(r|\theta)]^{\rho(r,p)} \tag{5.5}$$

En el modelo del dominio, por encima de las restricciones y los problemas, se ha añadido una nueva componente que se utilizará para implementar las estrategias instructivas que forman parte de la evaluación formativa y que está relacionada con los objetivos del aprendizaje. Esta componente es una agrupación conceptual de las restricciones y/o los problemas en conceptos del dominio. En la sección 5.3 se explicará detalladamente tanto lo que representa esta agrupación como su uso para instruir al alumno.

El **modelo del alumno** tendrá como elemento adicional un modelo sobre la agrupación anterior. La parte más importante de éste es la estimación del conocimiento $\theta$, la cual reemplazará la estimación original basada en heurísticos (rectángulo punteado y con el texto *Nivel*). Este conocimiento vendrá determinado por la aplicación del mecanismo de evaluación sumativa explicado en la sección 4.5. Aunque esta estimación puede ser utilizada para adaptación en componentes concretas, la función de densidad sobre el conocimiento utilizada para estimar el valor numérico de $\theta$, representada en la figura mediante la componente $P(\theta)$, proporciona más información. Esta densidad se calcula a partir de la función de verosimilitud de la definición 4.19 y permite aplicar estrategias de selección que necesitan como entrada esta función. Ambas estimaciones del conocimiento se determinan para cada componente de la agrupación conceptual del modelo de dominio, la cual se explicará en mayor detalle en la sección 5.3.

La tercera componente extendida es el **módulo pedagógico** que se encarga de dirigir la instrucción. Esta componente añade una nueva lógica de funcionamiento para implementar las estrategias formativas que parten de la combinación MBR + TRI. Además, incorpora los elementos necesarios para implementar estas estrategias en base a las características extendidas de los modelos de usuario y de dominio. Las estrategias propuestas se explican en detalle en el apartado 5.3.

### 5.2.2. Traza del conocimiento en el MBR mediante la TRI

La extensión de las componentes MBR permite llevar a cabo el modelo de evaluación sumativa detallado en el capítulo anterior. Sin embargo, hay que tener en cuenta ciertas características a la hora de usar ese modelo en sistemas amplios. Como ya se comentó en el apartado 4.5, durante la calibración o la evaluación, no debería haber aprendizaje de una restricción determinada. Para evitar esto se emplean los métodos de recolección de evidencias de la sección 4.5.1.1. En este apartado se explica la idoneidad de los dos métodos en sistemas extensos con dos propósitos: para fines formativos, o para otros fines.

#### 5.2.2.1. Traza del conocimiento para evaluación sumativa o calibración

Cualquiera de los dos métodos anteriores de recolección de evidencias puede realizarse si no hay aprendizaje. Es decir, en una evaluación sumativa, o para realizar la calibración. No obstante, la aplicación de este método debe realizarse durante periodos cortos de tiempo, ya que, de otro modo, estaría descartando información importante asociada al conocimiento cambiante del estudiante. Consideremos, por ejemplo, SQL-Tutor. En este sistema MBR no hay ni ninguna restricción sobre el número de intentos por problema, ni ningún tipo de imposición en la secuencia de problemas a resolver y, por tanto, los estudiantes pueden realizar el número de sesiones que quieran, cuando quieran y resolver tantos problemas por sesión como gusten. Esto hace que una restricción pueda ser relevante varias veces, en diferentes momentos a lo largo del periodo de uso de cada estudiante, cada vez asociadas a diferentes estados del conocimiento. Esta situación hace que con el método de evaluación propuesto en el capítulo anterior no recoja la evolución del alumno en el conocimiento que se produce durante largos períodos de uso y, en general, en cualquier sistema tutor.

Para afrontar el problema introducido en el párrafo anterior, se propone una forma de aplicar el modelo de evaluación que permite realizar la traza del conocimiento del alumno mediante los mecanismos de la TRI en el MBR. Este nuevo método es equivalente a la técnica de la TC en los tutores cognitivos y consiste en tener en cuenta la evolución del conocimiento a la hora de construir la matriz de rendimiento del estudiante. Como parte de este método se redefine el concepto de "sesión" y "estudiante" que serán considerados como fuente de evidencia para formar la matriz.

Una sesión tradicional se suele referir al periodo de tiempo que trascurre desde que el estudiante entra en el sistema, lleva a cabo alguna actividad (o actividades) y luego cierra la sesión. Si la actividad de un estudiante en sesiones consecutivas se agrupan teniendo en cuenta períodos de actividad lo suficientemente cerca en el tiempo, se pueden tener ventanas de tiempo donde el conocimiento entre las sesiones sea constante o sin cambios significativos. Este periodo de tiempo define el concepto de *sesión de conocimiento constante* que será referido como CK-sesión (abreviatura del inglés *Constant Knowledge Session*).

El tiempo que separa dos sesiones tradicionales consecutivas en una CK-sesión no debe ser superior a cierto umbral, de lo contrario se considera que es lo suficientemente grande como para que el estudiante haya podido cambiar su conocimiento mediante alguna fuente externa. Esto se puede expresar formalmente de la siguiente manera: sea $a_{mi}$ el instante en el que la última acción $m$ sucedió en la sesión $i$-ésima ($S_i$); $a_{0(i+1)}$ el momento en que la primera acción tuvo lugar en la sesión $(i+1)$-ésima

($S_{i+1}$); y $T_{CK}$ un umbral fijo que representa un período de tiempo en el que se puede suponer que el conocimiento no va a cambiar (el nombre viene del inglés *Threshold of Constant Knowledge*). Si $(a_{0(i+1)} - a_{mi}) < T_{CK}$ entonces, $S_i$ y $S_{i+1}$ pertenecen a la misma CK-sesión. Para entender mejor esta formulación, se presenta un ejemplo en la sección 5.2.2.1, el cual se resume en la figura 5.3.

Mediante esta distinción, cada CK-sesión puede considerarse como una fuente de evidencia para ser usada para la construcción de la matriz de rendimiento del estudiante. Las CK-sesiones de un estudiante reflejan diferentes estados del conocimiento del mismo y, por tanto, deben ser tratadas como si fueran estudiantes diferentes a la hora de construir la matriz. Esto da lugar a un nuevo concepto que hemos denominado *estudiante virtual* y que representa un estudiante no existente cuyo conocimiento puede ser considerado constante durante el proceso de medida. Cada estudiante real tiene tantos estudiantes virtuales como CK-sesiones.

De acuerdo a lo anterior, el conjunto de CK-sesiones de un estudiante se convierte en un conjunto más amplio de estudiantes virtuales, cada uno con un estado de conocimiento diferente. Esto hace que, de cara a usar los mecanismos de la TRI, se evite el aprendizaje inter-sesiones (entre diferentes sesiones), pero todavía es necesario considerar el aprendizaje intra-sesión (dentro de una sesión) y minimizarlo mediante algunos de los métodos de recolección de evidencias presentados en el apartado 4.5.1.1.

#### Creación de la matriz de rendimiento

Para hacer que el lector comprenda mejor esta metodología y la forma de construir la matriz de rendimiento usando las CK-sesiones, se presenta un ejemplo a continuación. El ejemplo considera un conjunto muy reducido de ocho restricciones del dominio y un solo estudiante. La actividad que el estudiante ha realizado en diversas sesiones en un sistema MBR es esquematizada en la figura 5.3. En esta imagen se muestra una serie de intentos, cada uno viene representado por un rectángulo encabezado por la etiqueta $I_{ij}$, donde $j$ es el número de intento para el problema $i$. Cada intento tiene una lista de restricciones relevantes que puede ser diferente para dos intentos consecutivos de un mismo problema, ya que el estudiante podría haber añadido nuevos elementos en la solución presentada. Esto se representa por ejemplo en el intento $I_{42}$ el cual tiene relevante la restricción $R_3$ con respecto al intento $I_{41}$.

En este ejemplo, el estudiante ha realizado tres intentos sobre el problema 4; seguidamente, dos sobre el problema 2; después, dos intentos del problema 1; y, de nuevo, dos intentos más en el cuarto problema. El espacio horizontal entre cada par de intentos representa el tiempo que transcurre entre ellos. En este caso, se pueden observar tres espacios de tamaño significativo ($t_1$, $t_2$ y $t_3$) entre los cuatro problemas. La agrupación de evidencias en CK-sesiones se hace comprobando que los intentos no estén separados en el tiempo una cantidad mayor que el umbral $T_{CK}$. En el ejemplo, podemos ver que sólo $t_2$ es mayor que el valor umbral y, por tanto, hay dos CK-sesiones ($CKS_1$ y $CKS_2$), cada una representa una sesión única de un estudiante virtual ($VS_1$ o $VS_2$, según corresponda). En la figura 5.3 también está marcado con un círculo la primera vez que la restricción es relevante en la CK-sesión.

Nótese que podría ocurrir que el tiempo entre cada par de intentos consecutivos sobre un mismo problema ($a_{ij}$ y $a_{i(j+1)}$) fuera mayor que el tiempo entre intentos sobre problemas diferentes pero consecutivos ($a_{ij}$ y $a_{hk}$ tal que $h$ es realizado justo después de $i$). En este caso, a no ser que el estudiante hubiera cerrado la sesión por algún

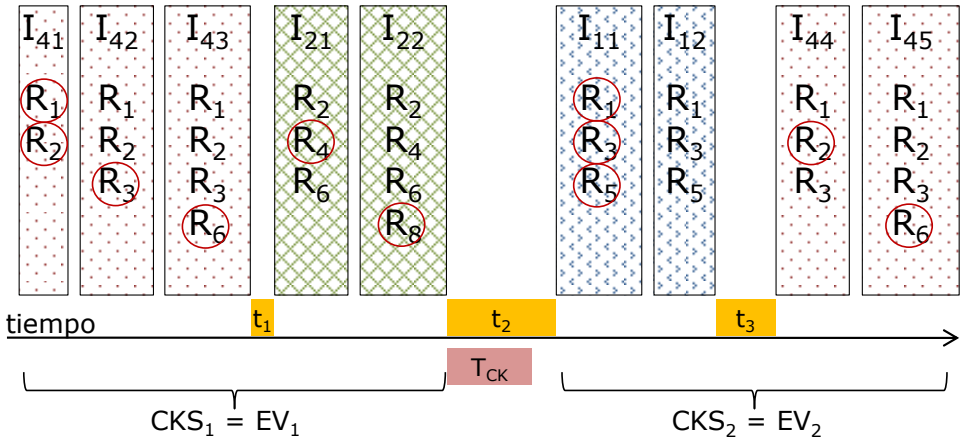



Figura 5.3: Agrupación de intentos en CK-sesiones y selección de restricciones dentro de cada CK-sesión.

motivo, serían considerados como parte de la misa sesión ya que durante ese tiempo el estudiante está resolviendo el problema. Esto quiere decir que solo las pausas entre diferentes problemas intentados se tienen en cuenta para realizar la diferenciación en CK-sesiones.

Siguiendo con el ejemplo, y suponiendo que se usa el método de la primera vez que una restricción es relevante para recolectar las evidencias, la matriz de rendimiento se forma tomando el resultado asociado a las restricciones marcadas con un círculo. La matriz resultante se muestra en la figura 5.4. Aquí, cada columna está asociada con una restricción $R_i$ y cada fila a un estudiante virtual. El elemento $a_{ij}$ de la matriz será el valor resultado de aplicar la función de evidencia de una restricción en un estudiante (definición 4.17). De esta forma, cada elemento tendrá un intento $I_{hk}$ a partir del cual toma el resultado de satisfacción o violación de la restricción. Este intento es el que se refleja en los elementos de la matriz, en lugar del resultado de aplicar la función, con el fin de ilustrar la forma de realizar la construcción. El valor del resultado es 1 para representar el valor de corrección (restricción satisfecha); 0 para representar incorrección (restricción violada), o -1 si no la restricción no ha sido relevante en la CK-sesión.

| | $R_1$ | $R_2$ | $R_3$ | $R_4$ | $R_5$ | $R_6$ | $R_7$ | $R_8$ |
|---|---|---|---|---|---|---|---|---|
| $CKS_1 = EV_1$ | $I_{41}$ | $I_{41}$ | $I_{42}$ | $I_{21}$ | x | $I_{43}$ | x | $I_{22}$ |
| $CKS_2 = EV_2$ | $I_{11}$ | $I_{44}$ | $I_{11}$ | x | $I_{11}$ | $I_{45}$ | x | x |
| ⋮ | | | | | | | | |
| $CKS_n = EV_n$ | $I_{ij}$ | $I_{ij}$ | x | x | $I_{ij}$ | x | $I_{ij}$ | $I_{ij}$ |

Figura 5.4: Matriz de rendimiento después de agrupar las CK-sesiones y seleccionar las restricciones.

Una vez la matriz de rendimiento está construida, ésta puede ser utilizada para apli-

car los mecanismos de la TRI, tal y como se comentó en la sección 4.5, para calibrar las restricciones o para la evaluación sumativa del alumno. En ambos casos no se debería realizar ninguna estrategia formativa y habría que minimizar el refuerzo presentado, siendo recomendable usar el método de eliminación total del refuerzo para que la calibración inicial sea mucho más objetiva y rigorosa. La mayor ventaja de este método de formación de la matriz es que permite realizar la calibración o la evaluación de un sistema donde ha tenido lugar aprendizaje. El resultado es un conjunto de estimaciones del conocimiento cambiantes del estudiante $C_s = \{\theta_1, \theta_2, .., \theta_m\}$ que se ha denominado *traza del conocimiento mediante la TRI en MBR*.

#### 5.2.2.2. Traza del conocimiento para evaluación formativa

Si se quiere utilizar el método anterior como parte de una evaluación formativa "en vivo" donde, además de realizar evaluación, se debe proporcionar refuerzo al estudiante para guiar la instrucción, entonces existe un problema importante. Aunque la distinción en CK-sesiones se puede realizar, el problema se localiza en los métodos de recolección de evidencia. Siguiendo con el ejemplo anterior, y centrándose en el los primeros tres intentos del problema 4, se ilustra la problemática en la figura 5.5. En esta figura se puede ver el conocimiento que se calcularía usando el método de recolección de la primera vez relevante, etiquetado con $\theta_0$, que sería el calculado dejando fuera el efecto del aprendizaje. No obstante, esto puede ser válido en los enfoques anteriores de calibración y evaluación sumativa donde el refuerzo está minimizado.

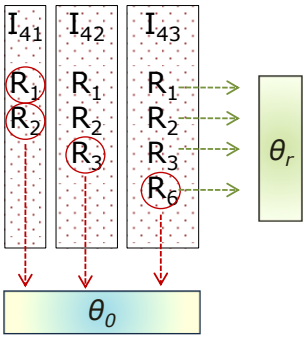



Figura 5.5: Problema del aprendizaje en la evaluación formativa.

En la evaluación formativa, el refuerzo se requiere como parte de la instrucción del alumno y este afecta al conocimiento del alumno modificándolo para aprender. Es por ello que, ante una situación como la de la figura, el valor de $\theta_0$ no sería indicativo del conocimiento del alumno tras recibir el refuerzo. Esto podría guiar erróneamente las decisiones instructivas del sistema al usarse una estimación del conocimiento que muy probablemente será incorrecta si el refuerzo ha hecho que el alumno aprenda. La estimación más adecuada del conocimiento en un enfoque formativo es aquella que contempla lo que el alumno ha aprendido, que en la figura está representado como $\theta_r$. Sin embargo este valor no se puede determinar con la TRI tradicional porque durante la medición hay aprendizaje, lo cual viola los supuestos de aplicación.

Para solucionar esto existen modelos de la TRI que modelan el aprendizaje en su formulación, como el propuesto por Lee et al. (2008). El estudio en profundidad y su utilización no se han incluido como parte del trabajo realizado en esta tesis, dejándose

propuestos para las líneas de investigación futuras. Estos modelos serían la base para usar las CK-sesiones con fines formativos.

Tanto en el uso formativo como sumativa de la evaluación, el proceso iría generando varias CK-sesiones que reflejarían la evolución del conocimiento del alumno. El proceso de evaluación, que se detallará en la sección 5.3.2.1 se haría cada vez que el alumno finaliza un problema determinado. Si el tiempo transcurrido entre la última CK-sesión del alumno y el inicio del problema es mayor que el umbral $T_{CK}$, entonces se crea una nueva CK-sesión con la evaluación realizada a la evidencia recopilada. En caso de que el tiempo sea menor que el umbral, entonces la evaluación de la evidencia se combina con la de la última CK-sesión y se vuelve a evaluar. Esta evaluación permite realizar un seguimiento de la evolución pero es necesario combinarse con alguna de las estrategias formativas mencionadas en la sección 5.3.

#### 5.2.2.3. Determinación del umbral de agrupación

El método propuesto separa las sesiones mediante un umbral de tiempo que determina de forma muy general si hay cambios en el conocimiento del alumno. Sin embargo, hay varias preguntas importantes que considerar en este método: ¿Cuál debe ser el umbral $T_{CK}$? ¿Hay otros métodos de agrupar las CK-sesiones? Como respuesta a estas preguntas se hace una breve discusión sobre los diversos mecanismos que se pueden seguir en la agrupación las CK-sesiones. De todas las formas que se mencionan sólo dos han sido estudiadas empíricamente dentro del trabajo realizado en esta tesis (ver sección 7.5). El resto se plantean como líneas futuras de cara a refinar la aplicación de la metodología formativa.

- TCK fijo: Es el más básico y simple y sobre el que se ha desarrollado un estudio empírico. Esta forma de uso implica que el umbral sea el mismo para todos los estudiantes en el sistema. El valor creemos es recomendable en esta metodología debe ser independiente del periodo de tiempo que va a durar la evaluación formativa y lo suficientemente pequeño para reflejar la evolución entre diferentes sesiones de uso. De esta forma el valor se situaría en minutos. Un estudio de los valores se ha realizado en la sección 7.5. No obstante, podrían fijarse valores mayores si se prevé que el alumno va a usar poco el sistema.

- Agrupación por problemas: Esta forma agrupa los intentos que se realizan sobre un problema concreto en una CK-sesión. Cada nuevo problema que se intenta tendrá una estimación del conocimiento diferente. El razonamiento de esta técnica está en que el conocimiento del alumno puede cambiar significativamente entre cada par de problemas. Esta metodología ha sido utilizada como parte del estudio empírico en la parte de evaluación experimental.

- Umbral por intervalos: Este método de formar las CK-sesiones tiene en cuenta el periodo de evaluación formativa y lo divide en intervalos de tiempo para los cuales es interesante disponer de una estimación del conocimiento. Estos intervalos pueden ser fijados equitativamente durante el periodo de evaluación formativa o también pueden ser irregulares con duraciones diferentes. La elección de un valor determinado del umbral debe realizarse teniendo en cuenta tanto la duración del periodo formativo como el uso que se prevé el alumno hará en el sistema. Por ejemplo, no sería recomendable usar intervalos de un mes en un periodo formativo

de cuatro meses si el alumno va a usar el sistema cada día porque se estaría descartando mucha información sobre su evolución. El caso completamente opuesto es fijar un intervalo de una semana si el alumno a lo mejor sólo usará el sistema una vez al mes. En esta línea, si se prevé el alumno usará más el sistema al final del periodo formativo, una vez que está más cerca la evaluación sumativa, se pueden fijar intervalos pequeños en ese periodo e intervalos mayores al principio.

- Agrupación por cambio significativo del conocimiento: En esta forma se agrupan las evidencias en una CK-sesión hasta la detección de un cambio destacado en el conocimiento del mismo. A partir de este evento, se crea una nueva CK-sesión hasta otro cambio. Este enfoque sería probablemente más apropiado para medir la velocidad de aprendizaje del alumno. La determinación de lo que es considerado como cambio significativo daría lugar a diferentes variantes que habría que estudiar.

- Agrupación por ratio de uso: Para evitar tener que considerar el uso de un estudiante en un sistema, puede fijase un periodo de evaluación formativa y el sistema iría agrupando en CK-sesiones dependiendo del uso del alumno, de manera dinámica. Una opción sería definir un umbral inicial que se adapte incrementándose o decrementándose de acuerdo a si el alumno ha realizado una proporción determinada de actividad en el sistema.

Estas agrupaciones pueden combinarse de forma conjunta durante el uso normal del sistema. Así, por ejemplo puede haber varios conjuntos de intervalos, con medidas del cambio significativo del conocimiento. Además, es posible distinguirse entre evaluaciones para el alumno y para un profesor tutor supervisor del aprendizaje, con la posibilidad de que cada uno fije sus propios intervalos o límites de acuerdo a sus intereses. Lógicamente, es necesario extender el estudio sobre cuál de los mecanismos es más efectivo en un entorno formativo real.

Un requisito indispensable de manera general en las formas de agrupación mencionadas es que el estudiante debe utilizar el sistema lo suficiente. De cara a generar una evaluación formativa adecuada sobre el conocimiento cambiante del alumno es necesario que éste utilice el sistema para que se pueda realizar una traza adecuada del conocimiento y actuar en consecuencia para paliar las deficiencias del aprendizaje.

## 5.3. Estrategias formativas mediante la TRI

Como se comentaba al principio de este capítulo, la evaluación formativa está compuesta por los elementos que extienden la evaluación sumativa para formar al alumno en base a los objetivos de aprendizaje que serán evaluados en la evaluación sumativa final. En el MBR los elementos relacionados con la formación del alumno son el refuerzo proporcionado para cada restricción y la estrategia de adaptación del contenido que es dirigida por el módulo pedagógico del sistema. Dado que el refuerzo también puede ser adaptado, de forma general, la estrategia de formación se corresponderá con la estrategia de adaptación del contenido del alumno. El reemplazo de las técnicas heurísticas por los mecanismos de inferencia de la TRI permite realizar una adaptación bien fundamentada y por tanto, el proceso formativo adopta esta misma característica.

La base que guía las estrategias formativas son los objetivos de aprendizaje, por lo que ambos elementos están íntimamente relacionados. Los objetivos son las metas que

el alumno debería cumplir y las estrategias son las acciones que el sistema realizará para hacer que el alumno oriente su formación a cumplir con esos objetivos. Veamos primeramente qué puede ser un objetivo de aprendizaje en un sistema MBR antes de explicar las estrategias formativas en la sección 5.3.3.

### 5.3.1. Objetivos de aprendizaje

Por norma general, los objetivos de aprendizaje que se tienen en cuenta para realizar una evaluación sumativa final están fijados y definidos previamente, incluso antes de realizar la evaluación formativa. De esta forma, el proceso formativo estará guiado por los objetivos que el alumno debería cumplir, ya sea un nivel determinado o una serie de requisitos más específicos. Inspirado en la filosofía de los tests referidos al criterio, donde se mide el grado de conocimiento acerca de un tema, estos objetivos de aprendizaje en el MBR pueden tomar diferentes formas.

La manera más simple de definir un objetivo de aprendizaje es establecer un valor general del conocimiento que el alumno debe cumplir. Si en la evaluación sumativa final, el alumno supera ese valor del conocimiento, habrá cumplido los objetivos. Esta forma es equiparable a la forma tradicional de evaluación de un examen en el que si el alumno supera cierto nivel, aprueba. Este objetivo está asociado al conjunto completo de elementos del modelo del dominio. El problema que tiene este objetivo para guiar la evaluación formativa es que no recoge información valiosa para realizar la instrucción. Concretamente, el problema radica en que el objetivo no impone qué conceptos son importantes, por lo que la instrucción podría alcanzar el objetivo con las evidencias de un subconjunto pequeño de conceptos y dejar el resto sin cubrir.

Para solventar este problema se propone una forma más completa de establecer objetivos de aprendizaje en base al modelo del dominio pero usando partes del mismo. De esta forma se puede dar más importancia en el aprendizaje a partes que serán consideradas posteriormente en la evaluación sumativa final. Para esta forma de establecer objetivos hay que considerar los elementos del modelo de dominio usados para determinar el conocimiento: las restricciones y los problemas. En base a éstos se puede definir una agrupación conceptual para fijar objetivos de aprendizaje más generales, de acuerdo a las necesidades de cada proceso formativo.

Este enfoque de agrupar las restricciones en conceptos ya ha sido considerado anteriormente en el MBR. Algunos de los estudios donde se menciona este enfoque son (Martin y Mitrovic, 2005, 2006; Martin et al., 2011). En (Suraweera y Mitrovic, 2004), además, los conceptos se organizan en una jerarquía que se usan como objetivos de aprendizaje para dirigir la estrategia instructiva. También, en la herramienta de autor ASPIRE, explicada en detalle en la sección 2.3.7.6, un paso requerido antes de generar las restricciones es la especificación de la taxonomía de conceptos involucrados en las restricciones. Aunque esta componente es algo que no forma parte de la estructura general del MBR, es prácticamente necesaria de cara a realizar una estrategia instructiva con objetivos múltiples.

En la figura 5.6 se presenta una agrupación genérica de dos tipos. En la parte superior se muestra una agrupación en conceptos sobre el conjunto de las restricciones. En esta agrupación, Cada nodo $C_i$ representa un concepto $i$ del dominio tal que $i \in [1, n]$. Las $m$ restricciones del dominio son los rectángulos con las etiquetas $R_1, R_2, ...R_m$.

Para asegurar que los mecanismos de la TRI son aplicables como parte de las estrategias de instrucción, las cuales se explicarán en la sección 5.3.3, la agrupación en

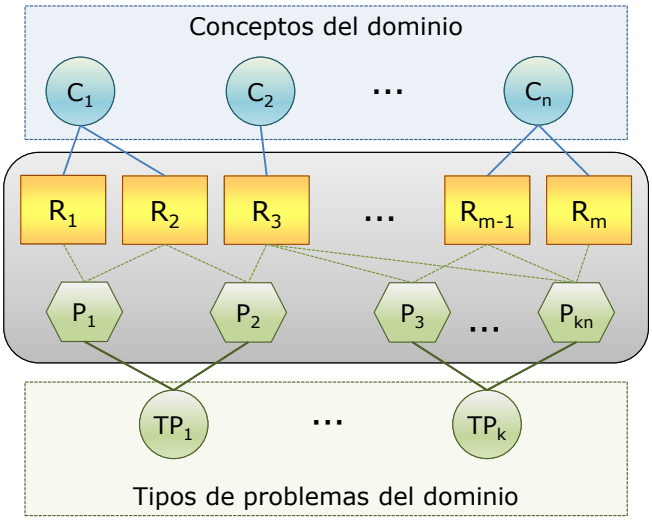



Figura 5.6: Agrupación de las restricciones en conceptos y tipos de problemas.

conceptos en el MBR debe cumplir unas características particulares:

- En primer lugar, un dominio tendrá un número determinado de conceptos $m$.
- Un concepto tendrá una o más restricciones y ningún otro sub-concepto.
- Una restricción sólo puede ser hija de un único concepto, puesto que los conceptos representan conjuntos independientes de principios del dominio.

En resumen, esta agrupación tiene forma de árbol donde los nodos pueden ser conceptos o restricciones y sólo hay un nivel de conceptos. Aunque podría definirse un estructura jerárquica con varios niveles de agrupaciones, al igual que se hace actualmente en algunos enfoques del MBR, evitaremos este tipo de estructura porque sería necesario utilizar modelos multidimensionales de la TRI. El trabajo de esta tesis se ha centrado en modelos unidimensionales principalmente, por lo que se deja propuesto este tipo de modelos y estructura más compleja para futuros trabajos.

En la parte inferior de la figura la agrupación se realiza en base a los problemas del dominio. Así pues, los problemas, etiquetados con $P_j$, $j \in [1, kn]$, son agrupados en tipos de problemas, etiquetados con $TP_i$, donde $i$ es un valor entre 1 y $k$). Realmente esta agrupación es una abstracción más sobre las restricciones, ya que, al fin y al cabo, un problema es la agrupación de las restricciones que son relevantes en él. No obstante, se utilizará la CCP para las estrategias instructivas que usan esta agrupación. Al igual que sucedía en la agrupación conceptual, podría haber diversos niveles de agrupaciones si se usaran modelos multidimensionales.

Aunque a priori es más intuitivo usar la agrupación de conceptos, la abstracción usando tipos de problema puede ser también un objetivo de aprendizaje útil en dominios en los que hay muchos conceptos y tipos de problemas y sólo algunos de los tipos de problemas son importantes de cara a la evaluación sumativa.

Usando esta agrupación, se pueden establecer objetivos de aprendizaje sobre unos u otros elementos, dependiendo de lo que se vaya a tener en cuenta en la evaluación sumativa final. El uso de esta estructura permite establecer no un objetivo solamente,

sino varios objetivos sobre diferentes conceptos o tipos de problemas, estableciendo un nivel de conocimiento mínimo que el alumno debería cumplir para cada uno y que serviría para guiar la estrategia formativa.

### 5.3.2. Evaluación de los objetivos de aprendizaje

Para cada objetivo de aprendizaje definido en la agrupación conceptual del modelo de dominio será necesario contar con un mecanismo para determinar el grado de cumplimiento del mismo para un alumno dado. El conjunto de estimaciones sobre el cumplimiento del alumno en cada nodo forma el modelo del alumno respecto de la agrupación.

La determinación del cumplimiento de un objetivo de aprendizaje consiste en valorar si el conocimiento del alumno respecto a las evidencias relacionadas con el concepto, cumple con un determinado valor impuesto previamente como objetivo de aprendizaje. Para ello, se propone un procedimiento de evaluación que actualice el conocimiento del alumno en cada nodo, a la vez que se obtienen evidencias.

El mecanismo propuesto está inspirado en el trabajo de Guzmán (2005), refinado posteriormente en (Guzmán et al., 2007). En éste se detalla cómo realizar la evaluación en sistemas de tests para modelos del alumno en varios niveles. Ésta puede llevarse a cabo mediante tres métodos: *evaluación agregada*, que evalúa solamente a los conceptos que forman parte del test; *evaluación completa*, que, además estos conceptos, evalúa todos aquellos nodos hijos hasta llegar al nodo que directamente contiene el ítem; y *evaluación completa con propagación hacia atrás* que es igual que la anterior pero, además, actualiza el conocimiento en los nodos padres hasta llegar a la raíz del árbol.

#### 5.3.2.1. Evaluación en base a conceptos

Para el caso de la agrupación de restricciones en conceptos, no existe un concepto que esté directamente siendo evaluado en el problema, sino varios. Esto hace que el proceso de evaluación se vea ligeramente modificado, actualizando cada concepto relacionado con las restricciones que han sido relevantes.

Antes de pasar a describir el algoritmo necesitamos formalizar la relación entre los conceptos y las restricciones. Para ello, supongamos que el conjunto de los conceptos del dominio es $\Upsilon$ y que la relación *contiene* entre conceptos y restricciones viene determinada por $\Upsilon = \{(x, y) : x \in \Upsilon \wedge y \in \tau \wedge (x \textit{ contiene la restricción y en su agrupación})\}$. En caso de usar modelos multidimensionales y varios niveles se debería usar una relación indirecta obtenida a partir del cierre transitivo de la relación anterior, tal y como se explica en (Guzmán et al., 2007).

Con esta formalización, el algoritmo para realizar la evaluación consiste simplemente en visitar cada concepto realizando la evaluación en aquellas restricciones que pertenecen a la relación $\Upsilon$ del concepto asociado. Esta evaluación se realiza después de que el estudiante $e$ haya resuelto un problema $p$. Los pasos del algoritmo se muestran en la tabla 5.1.

El algoritmo combina la evaluación sumativa con las consideraciones vistas en este capítulo: la traza del conocimiento y los objetivos de aprendizaje. Respecto de las CK-sesiones, en lugar de usar una matriz de rendimiento, ya que no se disponen de todos los datos de una vez, se van combinando cuando se van obteniendo (paso $2d$ del algoritmo). Esto es equivalente a aplicar la función de verosimilitud de la ecuación 4.19 sobre el

1. Inicialización:
   - Inicializar el conjunto de restricciones relevantes del problema $p$ realizado:
     $Relev_p = \bigcup_{j=1}^{J} r_j \; tal \; que \; \rho(r_j, p) = 1$, donde $J = |\tau|$.
2. Evaluación de conceptos.
   - Para cada concepto del dominio $C_i$ hacer:
     *a*) Si se cumple el criterio para crear una CK-sesión nueva (sección 5.2.2.3):
        1) Crear nueva CK-sesión para $C_i$.
        2) Inicializar distribución de conocimiento del alumno sobre el concepto $C_i$: $P_e(\theta|C_i) = 1$.
     *b*) Calcular el conjunto de restricciones que proporcionan evidencias a este concepto:
        $relev'_p = \bigcup_{k=1}^{n} r_k \; tal \; que \; r_k \in relev_p \wedge (C_i, r_k) \in \Upsilon$, donde $n = |relev_p|$.
     *c*) Si no hay restricciones que proporcionan evidencias ($relev'_p = \varnothing$):
        Seguir por el paso $2a$ en el siguiente concepto.
     *d*) Calcular distribución del conocimiento en el concepto $C_i$ combinando la que ya se tiene antes de añadir las nuevas evidencias ($P_e(\theta|C_i)$) con la obtenida mediante la aplicación de la definición 4.19 sobre $p$ y el conjunto de restricciones que proporcionan evidencia ($Relev'_p$):
        $P_e(\theta|C_i) = P_e(\theta|C_i)P_e(\theta|\{p\}, Relev'_p)$
     *e*) Calcular el nivel de $\theta$ sobre la distribución anterior y añadirlo a la CK-sesión actual.

Tabla 5.1: Algoritmo de evaluación de un alumno $e$ sobre la agrupación de conceptos tras realizar el problema $p$.

conjunto completo de evidencias de la CK-sesión.

Si se está realizando una evaluación sumativa, esta evaluación tomará las evidencias que no hayan sido tomadas antes en la CK-sesión (método de recolección de la primera vez relevante), mientras que si se usa la evaluación formativa, sería necesario usar un modelo que contemple el aprendizaje. En cada nodo del modelo del alumno se almacenará: la distribución del conocimiento de la CK-sesión actual; el nivel de conocimiento $\theta$ actual; y el conocimiento estimado de las Ck-sesiones anteriores, con el objetivo de poder hacer la traza de la evolución. Cabe mencionar que para la distribución de conocimiento del alumno se puede usar un conjunto discreto de valores sobre un intervalo determinado de $\theta$ para facilitar los cálculos y el espacio de almacenamiento requerido por el algoritmo.

#### 5.3.2.2. Evaluación en base a problemas

En el caso de agrupación de problemas el procedimiento es el mismo pero considerando las agrupaciones realizadas sobre los tipos de problemas. Por claridad, no se muestra otra vez el algoritmo, puesto que es prácticamente igual. En su lugar, se menciona la formalización necesaria y las diferencias sobre los pasos.

Para esta evaluación es necesario formalizar la relación entre los conceptos y los problemas. Para ello, supongamos que el conjunto de los tipos de problemas del árbol es $\Xi$ y que la relación *contiene* entre nodos y problemas viene determinada por $\Xi = \{(x, y) : x \in \Xi \wedge y \in \phi \wedge (x \textit{ contiene el problema } y \textit{ en su agrupación})\}$. La diferencia con el algoritmo de la tabla 5.1 es que, en lugar de usar la relación $\Upsilon$ se usa $\Xi$.

### 5.3.3. Adaptación formativa

Como se mostró en la sección 2.3.5, un elemento clave del MBR en la instrucción es la adaptación, la cual puede realizarse sobre el refuerzo o para seleccionar el siguiente problema. De este modo, hablar de estrategia instructiva implica hablar también de adaptación. En este apartado se revisará cuáles son las estrategias que, usando la TRI en el MBR, permiten dirigir la instrucción.

#### 5.3.3.1. Selección de problemas

Usando la analogía entre ítems compuestos y los problemas, es posible aplicar los mecanismos de adaptación mencionados en la sección 5.1.2 de la misma forma que se plantearon para ítems compuestos con la consideración de que la selección usa los valores de la última CK-sesión del estudiante. Para el caso particular del método de selección por máxima información, por equivalencia, existe una función de información sobre las restricciones y sobre los problemas. En el caso de las restricciones, esta función se denomina *Función de Información de la Restricción* (FIR) y tiene la misma forma que en los ítems simples de sistemas de tests enunciada en la definición 3.2 (sección 3.2.2). En el caso de un problema, es exactamente igual que la descrita para un ítem compuesto, en la definición 5.3, sumando la función de información de las restricciones dentro de la CK-sesión actual.

**Selección sin tener en cuenta los objetivos de aprendizaje**

En la formulación original del MBR para favorecer el aprendizaje existen diversas formas de selección, las cuales fueron explicadas en el apartado 2.3.5. De forma general, la estrategia instructiva consiste en buscar la restricción en la que el alumno presenta más violaciones, obtener los problemas en donde ésta es relevante y escoger el problema cuya dificultad difiere el mínimo posible en nivel con respecto al del alumno (Mitrovic, 2003a). El objetivo de seleccionar en base a la dificultad más parecida es que el problema sea el más cercano a la zona de desarrollo próximo de Vigotsky (1978).

A partir de este procedimiento de selección se observa una diferencia entre los objetivos de los mecanismos básicos de selección de la TRI, centrados en la evaluación, y los que promueven el aprendizaje. Concretamente, en los mecanismos básicos un ítem presentado anteriormente no se vuelve a considerar para la selección, mientras que en el MBR las restricciones que han sido relevantes se vuelven a tener en cuenta con el fin de determinar las que son más problemáticas. Aunque a nivel de problema, que es donde

se realiza la selección, esto no afecta al no considerase para selección los problemas ya presentados, sí que genera nuevas formas de selección basadas en los elementos más problemáticos:

- Inspirada en el mecanismo original del MBR, se define una técnica equivalente sobre los mecanismos de la TRI que hemos denominado *Selección por Restricción Problemática* (SRP). Para aplicarse esta técnica bastará con usar la última estimación del conocimiento del alumno $\theta$ para determinar la restricción más probable de violarse, lo cual equivale a la que tiene una probabilidad más baja de responderse correctamente. Este valor en cada restricción se obtiene aplicando la función CCR correspondiente para el valor $\theta$.

- Una extensión del mecanismo SRP es el uso de las restricciones que en conjunto son más problemáticas. De esta forma, usando las restricciones relevantes de un problema se puede determinar aquel que presenta más problemas para el alumno. Esta técnica se ha llamado la *Selección por Problema Problemático* (SPP), y consiste en determinar la probabilidad de fallar un problema usando su curva característica para el valor particular del conocimiento del alumno.

El mecanismo típico del MBR es buscar la restricción más problemática y seleccionar un problema con una dificultad cercana a la del estudiante. De esta forma, en nuestra metodología combinaría la SRP o SPP con la selección básica basada en la dificultad de un problema. Esta dificultad es su parámetro $b$ que, como vimos en la sección 5.1.2.3, se puede calcular de varias maneras. Estos mecanismos buscan hacer que el alumno aprenda pero, a diferencia del mecanismo original del MBR, reemplaza los heurísticos por el modelo probabilístico de la TRI.

Existen por tanto varios mecanismos básicos de selección que hemos mencionado: la SPP; los tres descritos en la sección 5.1.2; el mecanismo nuevo SRP; y la combinación de SRP con los tres anteriores. Aunque existen muchos otros criterios de selección, como los discutidos por Barrada (2012), o los ya explicados tests en la sombra (van der Linden, 2010), en esta propuesta nos centraremos sólo en los mecanismos básicos mencionados. Estos mecanismos de por sí no tienen en cuenta los objetivos de aprendizaje de la abstracción conceptual, pero son la base para los mecanismos que sí los tienen en cuenta y que se presentan a continuación.

#### Selección adaptativa por objetivos de aprendizaje concretos

En el MBR, hay antecedentes donde se utiliza una generalización como la propuesta anteriormente de agrupar los conceptos. Por ejemplo, Suraweera y Mitrovic (2004) usan una jerarquía donde se busca el concepto con mayor índice de violaciones para después seleccionar un problema asociado con una dificultad similar a la del alumno, o el más fácil de la lista ordenada si no hay un concepto que destaque sobre los otros. De forma similar se proponen varios mecanismos de selección que tienen en cuenta los objetivos de aprendizaje:

- Selección por tipo de problema: A partir de un tipo de problema $tp$, probablemente elegido por el alumno, el mecanismo implica crear un conjunto con todos los problemas que no se han presentado antes y que están relacionados con el tipo de problema directa o indirectamente en el árbol. Es decir, $Conjunto(t_p) =$

$p_i$ *tal que* $(t_p, p_i) \in \Xi$. Una vez obtenido este conjunto, se aplica alguno de los criterios básicos mencionados anteriormente.

- Selección por varios tipos de problemas: Este caso es igual que el anterior con la salvedad de que el conjunto de problemas a considerar para aplicarle alguno de los criterios básicos se forma como la unión de todos los problemas que están relacionados con los tipos a considerar. Es decir, si el conjunto de los tipos de problemas es $tipos = t_1, t_2, .., t_n$, entonces, $Conjunto(tipos) = \bigcup_{i=1}^{n} Conjunto(t_i)$, donde $Conjunto(t_i)$ de determina tal y como se expuso en el caso anterior.

- Selección en base a un concepto: De forma similar, a partir de un concepto $C_i$ se busca el conjunto de restricciones que están relacionadas directa o indirectamente con éste. Es decir $Conjunto(C_i) = r_j$ *tal que* $(C_i, r_j) \in \Upsilon$. Una vez obtenido este conjunto, se aplican los criterios básicos de selección sobre el total de problemas pero sólo usando las restricciones que están en el conjunto $Conjunto(C_i)$. Esto garantiza que la selección se realiza considerando sólo las restricciones relacionadas con el concepto seleccionado.

- Selección por conceptos múltiples: De forma similar a la generalización realizada para varios tipos de problemas, se puede generalizar también para varios tipos de restricciones. En este caso el conjunto de restricciones $conceptos = C_1, C_2, .., C_m$ se usa para formar el conjunto $Conjunto(conceptos) = \bigcup_{i=1}^{m} Conjunto(C_i)$, donde $Conjunto(C_i)$ se forma según se mencionó en el caso anterior.

En estos mecanismos se han mencionado algunos detalles como la elección de un concepto por parte del alumno, que serán extendidos en la sección 5.3.5, donde se explica cómo poner todas las componentes mencionadas en común para usar el sistema. Una diferencia que presenta el uso de estos métodos en la selección de problemas, respecto de la selección en TAI, es que no es requerido controlar la exposición de las restricciones dado que éstas son implícitas al problema y no se puede comprometer la seguridad de éstas. En su lugar, el control de exposición se debería realizar sobre los problemas.

#### Selección adaptativa general

Cuando no hay un objetivo concreto seleccionado, se pueden plantear diversas formas de elegir el siguiente problema mediante el uso de los mecanismos anteriores basados en objetivos concretos. Por ejemplo, la evaluación genérica, sin usar la agrupación de conceptos, es un caso particular en el que la selección se realiza sobre una única agrupación que considera el conjunto completo de restricciones o problemas. Si se desea proporcionar una selección adaptativa, en base a los objetivos, pero sin usar uno concreto, sino que se adapte dinámicamente a las necesidades del alumno, se proponen varias estrategias:

- Estrategia de selección por cumplimiento de objetivos: Dado el conjunto de objetivos de aprendizaje de un árbol, ya sea por tipos de problemas o por conceptos, se usa un conjunto de objetivos a considerar, que al comenzar el periodo formativo se inicializa al conjunto total. De los objetivos a considerar se busca aquel en el que el nivel del alumno difiere más del mínimo requerido. Una vez hallado este objetivo, se aplica la selección en base a los conceptos concretos explicada

anteriormente. Tras realizar el problema, se revisa el cambio en el conocimiento para eliminar del conjunto de los objetivos a considerar aquellos en los que el alumno ha superado el mínimo. También es posible que nuevas evidencias hagan que en un objetivo el alumno haya disminuido su conocimiento, por lo que debe ser introducido en el conjunto para ser considerado de nuevo.

- Estrategia de selección básica sobre el objetivo: Al igual que para un problema se puede determinar su curva como la combinación de las restricciones que son relevantes. Se puede definir una curva característica para los objetivos, usando la misma idea y combinando las restricciones. De esta forma, se pueden aplicar los criterios de selección básica pero a nivel del objetivo y luego aplicar los mecanismos de selección basados en objetivos concretos. Por ejemplo, se puede seleccionar el objetivo más problemático usando la curva característica del objetivo y el conocimiento del alumno y sobre este usar el método SPP.

#### 5.3.3.2. Refuerzo adaptativo

La segunda componente del MBR sobre la que se puede aplicar los mecanismos de la TRI para proporcionar adaptación es el refuerzo que se le muestra al estudiante. En la sección 2.3.4 se describía que la adaptación sobre este elemento tiene lugar cuando, tras detectar las restricciones violadas de una solución, se debe de elegir una restricción sobre la que proporcionar refuerzo. La forma más sencilla de elegir el refuerzo es usar el orden de ocurrencia o bien alguna ponderación. Sin embargo, en la revisión profunda realizada no se ha encontrado una descripción detallada de cómo se realiza esta adaptación en sistemas concretos.

Con el uso del mecanismo de evaluación sumativa, se tiene una estimación del conocimiento del alumno que permite realizar la selección del refuerzo usando una base bien fundamentada. Además, esta forma de adaptación tiene otra ventaja más sobre los enfoques que todavía se están desarrollando en el MBR: tal y como concluyen Martin et al. (2011), el refuerzo propuesto usando sólo las evidencias del modelo del alumno, sin tener en cuenta el de los demás alumnos es más ineficiente. En este sentido, la selección adaptativa proporcionada por la TRI no se centra sólo en la evidencia un alumno, sino que por detrás hay una población implicada en el proceso de calibración de las curvas, las cuales son la base para la selección mediante la TRI.

Las alternativas que se proponen para realizar el proceso de adaptación en el refuerzo son las siguientes:

- Selección de refuerzo en base a la necesidad: Cuando hay varias restricciones sobre las que hay que proporcionar un refuerzo se puede establecer un orden de presentación. Este orden se indica mediante la necesidad del mismo. Esto quiere decir lo siguiente: cuando una restricción es más probable de ser violada es más necesario proporcionar refuerzo sobre esa restricción que en otras. Usando este criterio, dado el conocimiento actual del alumno, se puede determinar la probabilidad de violarse la restricción de la misma forma que en la selección de los problemas se calculaba la restricción más problemática.

- Selección de refuerzo sobre objetivos: El enfoque anterior también puede aplicarse sobre la agrupación de conceptos. Existen trabajos previos en el MBR, tales como los de Martin y Mitrovic (2005, 2006) en los que el refuerzo se proporciona en base

a una jerarquía de conceptos y a diversos niveles de genericidad. Estos trabajos muestran que en algunos casos es más efectivo proporcionar refuerzo a distinto nivel de generalidad. De forma similar, se puede usar la jerarquía de conceptos para realizar una estrategia combinada en la que se ordena el refuerzo en dos niveles: primero por el concepto más problemático y, en otro nivel, la restricción más problemática.

- Uso de refuerzo asociado a objetivos: Inspirado en los trabajos mencionados, se puede definir un refuerzo sobre el concepto de la agrupación, en lugar de sobre la restricción. Aquel concepto que se identifica como el más problemático, tras la aplicación de los mecanismos de la TRI es el que proporciona el refuerzo a mostrar al alumno.

### 5.3.4. Modelo abierto del alumno

Un aspecto clave como parte del refuerzo que extiende la evaluación sumativa para dar lugar a una evaluación formativa es la apertura del modelo del alumno. Existen diversas investigaciones realizadas sobre sistemas MBR que dejan patente la mejora en el aprendizaje derivada del uso de modelos abiertos (Mitrovic y Martin, 2002; Hartley y Mitrovic, 2002; Thomson y Mitrovic, 2010; Mathews et al., 2012). El mostrar el conocimiento que el sistema tiene del alumno hace que éste pueda saber sus debilidades y hacia dónde debe dirigir su proceso de aprendizaje.

En esta apertura del modelo una componente esencial es la parte del modelo del alumno asociada a la agrupación conceptual que refleja el conocimiento en cada concepto o tipo de problema. En las investigaciones existentes en el MBR se recalca la necesidad de usar una abstracción de este tipo ya que las restricciones son demasiado específicas. Además, en dominios con muchas restricciones el mostrarlas todas es contraproducente de cara a dar una visión general del conocimiento del alumno y puede hacer que el alumno se trabe. A diferencia de los estudios realizados previamente en el MBR, el modelo abierto de en sistemas que combinan el MBR con la TRI usa el resultado de la evaluación sumativa en lugar de las estimaciones heurísticas.

Con el fin de que este modelo abierto del alumno destaque los resultados sobre aquellas agrupaciones de que poseen más importancia de cara a la evaluación sumativa final, los objetivos de aprendizaje fijados previamente se deben mostrar como parte de la visualización. Esta característica es exclusiva de la metodología propuesta en esta tesis, ya que en el MBR no existe actualmente el concepto de objetivo de aprendizaje con un uso posterior en una evaluación sumativa.

Para cada nodo de la agrupación se debe mostrar la evaluación sumativa en base a las evidencias de las restricciones asociadas a ese nodo. Esta evaluación sumativa puede ser sólo la última que el sistema ha realizado para el alumno, o el conjunto de CK-sesiones realizado mediante la traza del conocimiento. Aunque actualmente hemos evitado los modelos multidimensionales, si se usasen, la apertura del modelo podría mostrar la distribución del conocimiento a diversos niveles de conceptualización.

Un ejemplo simple de este modelo abierto se puede observar en la figura 5.7. Aquí se muestra la interfaz de Ingrid (Cruces et al., 2010; Conejo et al., 2011), un sistema para presentar modelos abiertos del alumno desarrollado en el grupo de investigación al que pertenece el tesitando. Esta figura es sólo una adaptación manual de una de las vistas de Ingrid para mostrar un ejemplo. En la agrupación mostrada se puede

observar un dominio *Demo* que tiene cuatro conceptos principales, de los cuales, sólo 3 han sido usados como objetivos de aprendizaje (icono en forma de llave). Cada concepto muestra, al situar el ratón sobre un concepto particular, un resumen sobre el nivel y el mínimo requerido como objetivo de aprendizaje. Los colores se usan para representar el conocimiento, el cual estará entre verde para alto nivel y rojo para poco nivel, o gris para representar la ausencia de evidencia.

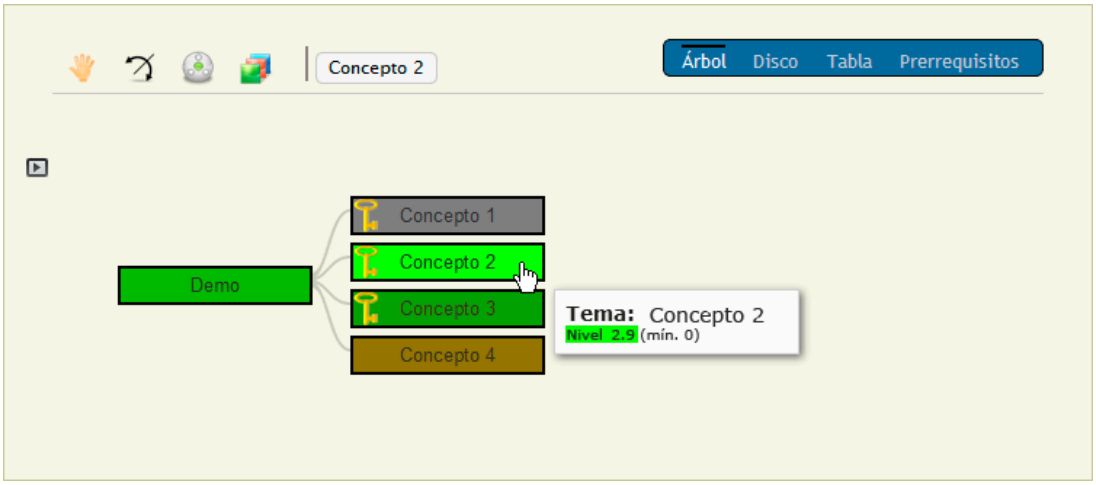



Figura 5.7: Ejemplo de modelo abierto del alumno sobre tutores MBR + TRI.

Aunque en la figura se muestra sólo la estimación del conocimiento actual y en la escala de puntuación nativa de la TRI, no la verdadera, sería necesario mostrar la traza del conocimiento mediante una gráfica que reflejara la evolución. La puntuación, para ser más entendible al usuario debería transformarse a puntuación verdadera tal y como se explicó en la sección 3.3.2.2.

### 5.3.5. Modos de funcionamiento

Como se ha podido ver en lo que va de capítulo, la metodología propuesta está compuesta de un mecanismo de evaluación sumativa, descrito y formalizado en el capítulo 4, y de los elementos que extienden el mecanismo anterior para proporcionar una evaluación formativa. Esta última, requiere del uso de modelos de la TRI que contemplen el aprendizaje, los cuales quedan propuestos para investigaciones futuras. Respecto del resto de elementos presentados, cada uno será más apropiado según el tipo de evaluación que se esté llevando a cabo por el sistema. Por este motivo, un tutor MBR que aplique esta metodología debe tener varios modos de funcionamiento que utilicen la combinación más apropiada.

De forma resumida, la metodología que se propone en esta tesis combina la evaluación sumativa y la formativa para hacer que el alumno aprenda de la siguiente forma: como paso previo a cualquier uso de la TRI es necesario calibrar las restricciones, tal y como se mencionaba en la sección 4.5.1. Otro paso previo a la aplicación de la metodología es la definición de los objetivos de aprendizaje que serán evaluados. Una vez se realizados los procedimientos anteriores, se realiza una etapa de evaluación formativa en la que se busca guiar al alumno para mejorar de cara a la evaluación final. Por último se llevaría a cabo la evaluación sumativa final para determinar el cumplimiento

de los objetivos.

Algunas de las combinaciones posibles entre los métodos existentes no son apropiadas para la evaluación con fines instructivos si su uso implica la limitación de las capacidades formativas del sistema. Igualmente, si la evaluación es meramente la emisión de un juicio, las características formativas no serán apropiadas. En base a estas consideraciones se definen dos formas de funcionamiento, cada uno con unas características determinadas que pretenden evitar la problemática expuesta.

#### 5.3.5.1. Uso del sistema sin aprendizaje

En este modo de funcionamiento el sistema usará la combinación de técnicas y elementos que eviten o minimicen el aprendizaje del alumno. Éste es útil en dos de las fases mencionadas en la aplicación de la metodología, cada una de ellas con sus particularidades:

- **Calibración de las restricciones**: Dada la importancia de esta fase, pues la evaluación posterior dependerá de las estimaciones de las restricciones, será necesario evitar cualquier aprendizaje. Para ello, las características de funcionamiento son las siguientes:
  - Recolección de evidencias: Se debe usar el método de eliminación completa de refuerzo, evitando cualquier influencia del aprendizaje en la evidencia.
  - Agrupación en CK-sesiones: En esta fase es la forma adecuada de agrupar en CK-sesiones es según el criterio de agrupación mediante problemas. Esta forma de agrupar ha sido la que empíricamente genera una calibración de mayor calidad. Más detalle sobre el experimento asociado se puede ver en la sección 7.5.
  - Adaptación: Cuando el sistema está recogiendo información para calibración la selección adaptativa de problemas será desactivada. En su lugar, se seleccionarán los problemas mediante alguno de los mecanismos de selección propuestos para esta tarea en la sección 4.5.1, ya sea, de forma aleatoria, mediante número de restricciones no presentadas o buscando homogeneidad en base a número de veces que una restricción ha sido relevante.
- **Evaluación sumativa final**: En esta fase la meta principal del sistema es la de determinar de la forma más objetiva posible el conocimiento del alumno. Esto es similar a la aplicación de los TAI para evaluar al alumno. Aquí también hay que evitar el aprendizaje al mínimo mediante las siguientes características:
  - Conocimiento inicial: Al igual que en los TAI es necesario un procedimiento de arranque sobre el conocimiento inicial. Dada la forma de la metodología propuesta, este proceso tiene lugar justo después del periodo formativo, por lo que se cuenta con una estimación a priori del conocimiento del alumno. Así, la estimación inicial es más fiable que los procedimientos típicos, los cuales pueden asignar un valor del conocimiento que diste del real.
  - Recolección de evidencias: Probablemente, en esta fase sería recomendable el uso del nivel más bajo de refuerzo posible avisando de si la solución es correcta. La base de esta hipótesis, todavía por probar, radica en que si el alumno ha cometido algún error que tenga como consecuencia la omisión

de partes de la solución con otras restricciones relevantes, con este refuerzo se podría hacer que el alumno añada estas partes, haciendo que el sistema evaluara con evidencias sobre un conjunto más amplio de restricciones. Este método requiere la técnica de la primera vez relevante para minimizar el efecto del aprendizaje.

- Agrupación en CK-sesiones: Esta forma de evaluación se realiza normalmente en una sesión corta, por lo que la agrupación en CK-sesiones puede realizarse mediante un intervalo que abarcara la duración de la sesión y garantizando que todas las evidencias de la evaluación recaen en la misma CK-sesión.
- Adaptación: En este caso la estrategia de selección de problemas más apropiada sería probablemente la que busca mejorar la precisión de la medición del conocimiento, tales como la selección de la máxima información o la selección bayesiana. Las estrategias que buscan mejorar las restricciones más problemáticas deben evitarse, ya que el objetivo no es proporcionar aprendizaje. El nivel de generalidad sobre el que aplicar la adaptación abre varias posibilidades. Así, se puede aplicar el mecanismo sobre problemas directamente, sin tener en cuenta los objetivos o, de forma más general, aplicar selección sobre un objetivo y después sobre las restricciones / problemas asociados. También sería recomendable usar algún mecanismo de control de exposición para evitar comprometer la seguridad de los problemas que son usados normalmente para la evaluación sumativa final. En este sentido, sería también necesario evitar mezclar problemas que normalmente se usan en aprendizaje con los usados para la evaluación sumativa final, lo cual puede requerir aumentar considerablemente el número de problemas.
- Mecanismo de parada: En este modo de uso, dado que se está midiendo formalmente el conocimiento como si de un TAI se tratara, esta opción debe ser tenida en cuenta, utilizándose algunas de las ya mencionadas en el apartado 3.3.2.4.

#### 5.3.5.2. Uso del sistema para aprender

El segundo modo se corresponde con la fase de **evaluación formativa** y del uso del sistema con fines instructivos. En este modo el sistema usa la evaluación sumativa en conjunción con los elementos mencionados que la completan para llevar a cabo la instrucción. Las características de esta forma de usar el sistema son las siguientes:

- Recolección de evidencias: Como ya se mencionó en la sección 5.2.2.2, los métodos de recolección de evidencias no son válidos en un uso conjunto de formación y evaluación al no tener en cuenta el aprendizaje. Por este motivo, en lugar de métodos de recolección se debe usar un modelo de la TRI que tenga en cuenta el aprendizaje producido por el refuerzo.

- Agrupación en CK-sesiones: Esta agrupación dependerá de los criterios que cubran necesidades de la evaluación formativa, no fijándose ninguno en particular.

- Adaptación: Para esta etapa se debería usar el mecanismo de selección de problemas y el refuerzo que tiene una mayor influencia positiva en el aprendizaje:

- Sobre los problemas, de las diversas técnicas propuestas no se puede garantizar la más efectiva al no haberse realizado todavía un estudio empírico. No obstante, la hipótesis inicial es que la selección de los elementos en los que el alumno presenta más dificultad es más apropiada. Así, se puede aplicar la selección general sobre el objetivo que presenta mayor dificultad y, posteriormente el problema asociado más problemático. También se puede usar una estrategia como la dirigida por objetivos para hacer cumplir los mismos.
- En el refuerzo también es necesario estudiar el más efectivo. En cualquier caso, cualquiera de los mecanismos propuestos sigue una base con mayor fundamento que las tradicionales del MBR, al proporcionarse sobre el conocimiento estimado mediante la TRI.

Como se ha mencionado, la eficiencia del nivel de generalidad y de las muchas estrategias posibles está todavía por estudiarse.

En este modo, con el fin de permitir que el alumno pueda también participar en su aprendizaje y, motivado por el uso de un modelo del alumno abierto, el sistema no debe forzar al alumno a realizar el problema que el sistema considera más apropiado. Por el contrario, se debe dar la opción al alumno para seleccionar entre el problema que el sistema considera mejor; un problema de un tipo determinado, que elige el sistema adaptativamente usando la selección en base a un tipo de problema concreto de la jerarquía; o realizar un problema asociado a un concepto particular, siendo también elegido por el sistema adaptativamente en base a la jerarquía de conceptos.

Aunque se han agrupado en dos categorías hay tres modos de funcionamiento reales que corresponden a la calibración, evaluación formativa y evaluación sumativa. Su utilización en un entorno real se realiza en el mismo orden en el que se han nombrado. La opción de fijar un modo u otro se debe realizar por un usuario con rol de profesor o un administrador del sistema. Teóricamente, el modo de funcionamiento de calibración sólo se usaría una vez (aunque podría requerirse re-calibrar las restricciones). Los otros dos modos dependerán de las características temporales en las que se organice el curso donde vaya a ser usado el sistema.

## 5.4. Utilidad de la TRI en el MBR: calidad de las restricciones

Como se explicó detalladamente en la sección 2.3, y como demuestran los resultados del experimento que será descrito en la sección 7.1, el MBR es un paradigma eficaz como herramienta formativa. El poder de su efectividad recae sobre en el conjunto de restricciones del modelo de dominio. Este conjunto es incluso más fácil de construir utilizando las herramientas de autor como ASPIRE (ver apartado 2.3.7.6), ya que no son requeridos conocimientos de programación. Lo que sí es necesario para modelar adecuadamente las restricciones es tener un amplio conocimiento del dominio en cuestión, pero esto es algo necesario con cualquier técnica de modelado del alumno que se use para construir un nuevo entorno de aprendizaje.

A pesar de la efectividad del sistema y de la facilidad de construcción del conjunto de restricciones, las restricciones podrían no reflejar debidamente el principio del dominio asociado, convirtiéndolas en instrumentos de medida no adecuados para evaluar.

Incluso la elaboración de las restricciones por expertos humanos podría dar lugar a esta situación, dada la imperfección de la naturaleza humana.

### 5.4.1. Fuentes de error en la elicitación de restricciones

Hay dos fases en los que la combinación de la TRI con el MBR puede presentar problemas y son fuentes de error que dan lugar a restricciones no apropiadas para evaluar. La primera fase se encuentra en el paso previo a la calibración de éstas: el proceso de codificación de las mismas. Incluso si la restricción no tiene otra fuente de error, si está mal codificada, es un instrumento de medida erróneo que genera estimaciones del conocimiento incorrectas o deformadas. Esta situación fue descubierta experimentalmente mientras se probaba la validez de mecanismos para detectar restricciones inapropiadas de acuerdo a la segunda fuente de error que se mencionará seguidamente. El estudio mencionado se explica detalladamente en la sección 7.7. Respecto a esta fuente de error, pueden cometerse dos tipos de errores:

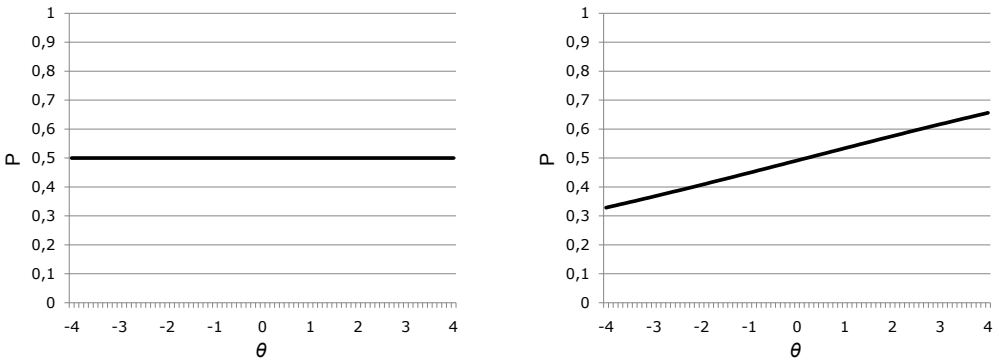
- El primer error, es evidente y se presenta cuando una restricción ha sido codificada incorrectamente. Esto puede darse por un mal uso del lenguaje asociado a la codificación o bien porque la restricción no esté realmente representando un principio correcto sobre el dominio. El primer caso es más frecuente que el segundo, pues implica el uso de un lenguaje de programación y, aunque se posea mucha experiencia en este terreno, es muy probable cometer errores como la construcción incorrecta de una expresión lógica o el uso incorrecto de un operador aritmético. Un fallo insignificante en la representación puede hacer que las condiciones de relevancia y / o satisfacción sean totalmente diferentes respecto de lo que realmente se estaba intentando modelar, resultando en una restricción incorrecta. El segundo caso es menos probable, pero es también posible. Es menos probable en el sentido de que está asociado directamente con el conocimiento del dominio que se está modelando, tarea que debería ser realizada por expertos del dominio. No obstante, si la persona que identifica la restricción posee un error conceptual, puede transmitirse igualmente a la codificación de la restricción.

- En caso de que el principio esté codificado correctamente, según la intención de la persona que realiza esta tarea, puede haber otros errores. Concretamente, estos errores están asociados con el nivel de generalidad de los principios codificados en las restricciones. Para entender esto se plantea el siguiente ejemplo: imaginemos restricciones sobre un dominio en el que la solución a los problemas del mismo puede ir representada mediante una tabla con una serie de filas y columnas, cada columna representa un elemento de la solución en el que las filas contienen valores sobre determinadas características. En este dominio, un principio podría requerir que la tabla tuviera un número determinado de columnas. No obstante, si cada columna tiene un significado semántico relevante para la solución, en lugar de una única restricción, probablemente, lo más adecuado sería tener varias restricciones, una por cada columna semánticamente importante, comprobando que éstas estuvieran presentes en la solución. De esta forma, una restricción puede modelarse de acuerdo a un principio más general o más específico. En algunos dominios, la decisión sobre qué nivel de generalidad más adecuado es fácil de determinar objetivamente, bien porque el dominio sea bien definido y simple, o porque los conceptos involucrados sean claros y concisos. En otros dominios, sin embargo,

pueden existir ambigüedades o controversia. Consecuentemente, la decisión sobre la especificidad de la restricción tiene un carácter más subjetivo y sujeto a la mentalidad de la persona que diseña la restricción. Esta situación es de la misma naturaleza que la estudiada por Martin y Mitrovic (2006) al agrupar o separar restricciones de acuerdo a la eficacia del refuerzo sobre las agrupaciones o divisiones de restricciones. De la misma forma, algunas restricciones muy generales serían un mejor instrumento de evaluación si éstas fueran divididas en otras más específicas, o, al contrario, si son muy específicas, su generalización, agrupando varias restricciones, podría ser mejor para evaluar.

La segunda fase que puede suponer una fuente de error es en la aplicación del modelo de evaluación sumativa para determinar el conocimiento. El motivo no está relacionado con la validez de los mecanismos de la TRI, sino con la cantidad de evidencia de la que se dispone, la cual afecta a las dos etapas fundamentales del proceso de evaluación.

- En la primera etapa, donde se calibran las CCR, si el algoritmo de calibración no recibe ninguna evidencia como entrada, teóricamente se obtendría una curva como la de la figura 5.8a, en la que la curva realmente sería una línea. Si el número de evidencias es muy reducido, la curva presentaría una apariencia similar a la de la figura 5.8b, la cual presenta una curva real con una forma más parecida a una línea.




(a) Curva generada a partir de ninguna evidencia.

(b) Curva generada a partir de pocas evidencias.

Figura 5.8: Muestras de diferentes curvas en base al número de evidencias.

- En la segunda etapa, el proceso de evaluación combina las evidencias con las curvas mediante una operación multiplicativa. El resultado, al multiplicar curvas planas o con una forma parecida, influye en la distribución del conocimiento resultado haciendo que ésta se vea sesgada y que la estimación del conocimiento no sea apropiada. Es por ello que las restricciones con poca o ninguna evidencia no son adecuadas para realizar evaluación pues no son un reflejo fiable del conocimiento del alumno.

En la práctica, el proceso de calibración puede tomar como entrada valores por defecto de los parámetros que se van ajustando con las evidencias. De esta forma, si no hay evidencias, no se obtiene una curva plana completamente. No obstante, tanto con pocas como con ninguna evidencia, las curvas pueden tomar valores muy dispares, por lo que,

incluso con estas medidas, las restricciones asociadas son candidatas idóneas para ser omitidas del proceso de evaluación.

### 5.4.2. Mecanismo de determinación de la calidad de las restricciones mediante la TRI

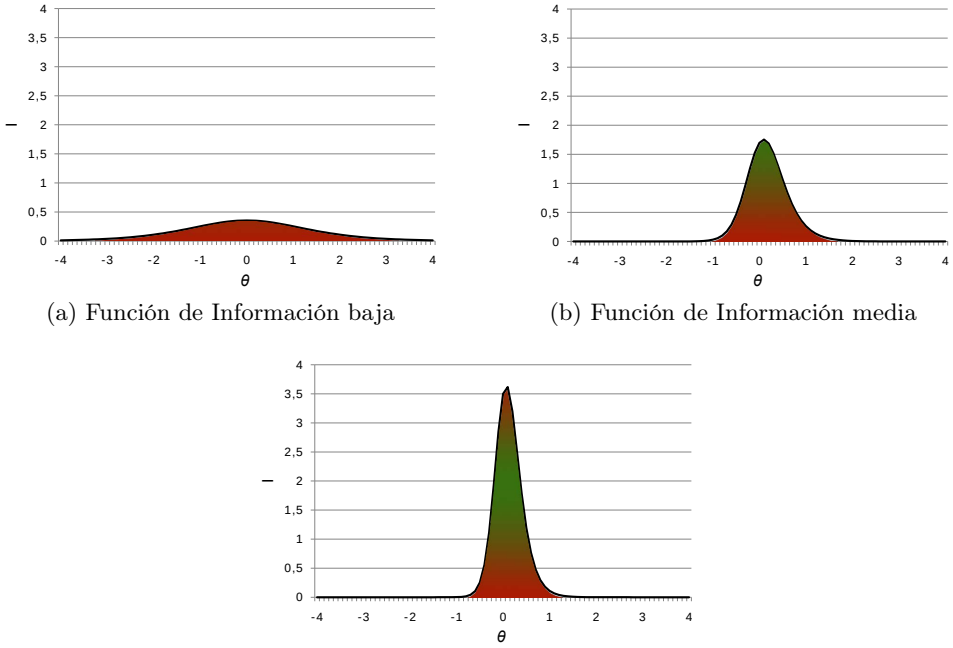
En un trabajo relativamente reciente, Martin et al. (2011) utilizan curvas de aprendizaje como instrumento para saber si las restricciones y las diferentes abstracciones en conceptos proporcionan un rendimiento adecuado en el aprendizaje. Este enfoque se basa en detectar los elementos las curvas de aprendizaje de aquellos elementos que no reflejan aprendizaje. De esta forma, la estrategia instructiva usa las generalizaciones o especializaciones cuya curva refleja un mayor aprendizaje. Los elementos que no están proporcionando un aprendizaje adecuado son candidatos para ser revisados. Además, los autores las usan para comparar sistemas diferentes, realizando una comparación con un sistema tutor cognitivo.

En esta tesis se propone un enfoque similar pero utilizando los mecanismos bien fundamentados de la TRI. Aunque el objetivo de detectar elementos no apropiados es común, el criterio usado para detectarlos difiere ligeramente: los mecanismos de la TRI están más enfocados a la evaluación mientras que las curvas de aprendizaje se centran en el aprendizaje. Sería necesario estudiar cuáles de ellos tienen más eficiencia, dependiendo del objetivo. Nuestra hipótesis respecto a este asunto es que, aunque las curvas de aprendizaje son efectivas para dirigir la estrategia instructiva en un sistema MBR tradicional, en un sistema MBR + TRI, el uso del mecanismo que se propone a continuación, tendría un mayor impacto positivo en el aprendizaje. No obstante esto se propone como líneas futuras, siendo necesaria una mayor investigación.

Para detectar las dos fuentes de error comentadas anteriormente, y haciendo uso de la analogía entre los ítems y las restricciones, los mecanismos que se utilizan para estudiar la calidad de los ítems en sistemas de tests pueden ser utilizados de la misma forma sobre las restricciones. El mecanismo al que nos referimos es la función de información. Esta función, además de para seleccionar ítems (ver sección 5.3.3), es usada para describir los ítems y los tests. La aplicación de esta función no implica más que adaptar la función de la ecuación 3.8 para las restricciones dando lugar a que se ha denominado *Función de Información de la Restricción* (FIR).

El resultado que puede darse tras aplicar la FIR puede variar dependiendo de la calibración realizada sobre las curvas. En la figura 5.9 se muestran tres ejemplos del resultado de aplicar la ecuación 3.9 para calcular la FIR de tres restricciones diferentes calibradas con el modelo 3PL. El resultado puede ser un ítem con una función de información baja (figura 5.9a); media (figura 5.9b); o alta (figura 5.9c). Todas estas curvas de información están centradas en torno al nivel de dificultad $\theta = 0$.

Encontrar una curva muy baja o muy alta es indicativo de que alguno de los posibles errores comentados anteriormente ha ocurrido. Para determinar una curva demasiado baja / alta requiere de compararla con el resto de curvas de las restricciones. A no ser que todas las restricciones tengan una anomalía o haya pocas evidencias, se podrán detectar situaciones anómalas. Una forma de cuantificar la función de información es usando el área de la curva ($A(I_r)$) (Hambleton et al., 1991), la cual vendría determinada por la ecuación 5.6




(a) Función de Información baja

(b) Función de Información media

(c) Función de Información alta

Figura 5.9: Diferentes tipos de Función de Información para el modelo 3PL.

$$A(I_r) = \int_{-\infty}^{\infty} I_r(\theta) d\theta \tag{5.6}$$

En el experimento realizado asociado con el uso de la función de información como índice de la calidad de las restricciones se utiliza esta función área sobre la función de información de las restricciones (ver sección 7.7 para más detalles). En el experimento se da un umbral usando como medidas de referencia la media y la desviación típica para dividir las restricciones "buenas" de las "malas". No obstante, este valor podría no ser válido en otro dominio, por lo que es necesario futuras investigaciones sobre este aspecto.

El hecho de detectar una curva muy baja respecto de las demás restricciones indica que la restricción no proporciona apenas información. En este caso la situación más probable es que para esta restricción no se disponían de suficientes evidencias en el proceso de calibración. También podría ser que esta restricción esté incorrectamente codificada o que, estando bien codificada, no sea relevante con frecuencia, lo que es indicativo de que puede haber algún error conceptual y que la restricción representa un principio demasiado específico.

Cuando se detecta una curva muy alta puede ser también que se haya codificado incorrectamente ésta, o bien que esté correctamente implementada pero el principio es conceptualmente incorrecto, puesto que agrupa muchas evidencias, en comparación con las demás. En este caso, lo más probable es que sea necesario dividir la restricción en varias que representen principios más específicos.

Lógicamente, errores comedios en una restricción cuya función de información es comparable a las del resto, pasarían desapercibidos usando esta técnica. En los otros dos casos, la función de información puede advertir justamente después de realizar un

proceso de calibración de aquellas restricciones que no son adecuadas para evaluar o que requieren de más evidencias para poder tener una estimación adecuada. Por tanto, esta función sirve como herramienta para analizar la idoneidad del modelo de dominio.

De la misma forma que se puede aplicar para restricciones, se puede aplicar para problemas del dominio con el fin de compararse y estudiar cuáles no son informativos y, por tanto, deberían descartarse. En este sentido, se usa la función información de la definición 5.3 pero considerando un problemas en lugar de un ítem compuesto, y restricciones, en lugar de componentes. Una propiedad que resalta de cualquier análisis simple sobre la función de información de problemas, similar a la de un test, es que el número de restricciones relevantes está relacionado con el valor de la información. Así pues, problemas con muchas restricciones, tendrán un valor alto.

Aunque es necesario realizar más estudios en este ámbito, tenemos la hipótesis de que problemas con mayor información son preferibles puesto que darían mayor información del conocimiento del alumno, en la misma duración que otros problemas. Con las primeras experiencias se ha podido comprobar que en relación con los tests, los problemas son más cortos y evalúan el conocimiento del alumno en mucho menos tiempo, ya que los "ítems" están implícitos mientras el alumno resuelve el problema.

Además de usar el área de la curva como indicativo de la información de una restricción o problema es posible usar otros valores que corresponden a otras propiedades de la curva como el valor máximo o la curtosis. No obstante estos parámetros todavía no han sido estudiados, por lo que sería necesario ampliar también el estudio a estos valores para ver la relación con los errores mencionados anteriormente y comparar la efectividad con el que se está usando actualmente.

### 5.4.3. Otras posibles utilidades

Como se ha visto a lo largo del capítulo, la TRI no sólo puede de utilizarse como mecanismo de evaluación sumativa o para extender las estrategias formativas del alumno en EIRP. En esta sección, se ha mostrado el uso de la función de información para estudiar la calidad de los elementos usados en los mecanismos de evaluación de la TRI. Pero además, gracias a la analogía entre sistemas de tests y sistemas MBR, se pueden utilizar otros mecanismos útiles de la TRI que no existen actualmente en el MBR.

La función de información utilizada en este capítulo es una de las herramientas más interesantes, pues además de la función explicada y de su uso en la selección adaptativa, puede servir para comparar el rendimiento de distintos sistemas. El uso de la eficiencia relativa (Hambleton et al., 1991), un cociente entre la función de información de dos sistemas, da una medida de cómo de informativo es un test respecto de otro. De la misma forma, se puede aplicar para sistemas MBR gracias al enlace establecido, permitiendo comparar cómo de útiles pueden ser los problemas de distintos sistemas, e incluso comparar tests sobre la materia, con problemas del MBR.

En (Hambleton et al., 1991) se mencionan otros mecanismos de la TRI para determinar ítems con otras anomalías, como es el caso de los *ítems con funcionamiento diferenciable* (en inglés *Diferential Item Functioning*), los cuales se definen como aquellos que para alumnos de distintos grupos pero con el mismo nivel de conocimiento, presentan diferente probabilidad de responder el ítem correctamente. Estos ítems también deberían ser considerados, pues pueden generar evaluaciones diferentes de acuerdo a la población donde se apliquen.

El estudio de estos métodos y otras posibles utilidades de la TRI se deja propuesto como parte del trabajo futuro, ya que antes de explorar herramientas que garanticen la calidad del aprendizaje es necesario afianzar muchos de los aspectos todavía pendientes en relación con las diferentes estrategias instructivas propuestas.

## 5.5. Conclusiones del capítulo

A lo largo de este capítulo se ha extendido el modelo de evaluación sumativa expuesto en el capítulo anterior con aquellos elementos que permiten realizar una evaluación formativa en base a métodos bien fundamentados. Estos elementos, que en el MBR están directamente relacionados con la adaptación mediante el refuerzo y la selección del siguiente problema, han sido extendidos con la TRI. Debido a la importancia de la adaptatividad en el MBR, los mecanismos de los TAI son deseables para estas tareas.

Con el objetivo de usar los mecanismos de los TAI para la selección de problemas, el capítulo comienza revisando la aplicación de la metodología propuesta en sistemas de tests. El objetivo es, además de mostrar el modelo aplicable para extender la limitación de estos sistemas, estudiar las implicaciones de usar la TRI para la tarea de selección de problemas en su entorno original. De esta forma las consideraciones a tener en cuenta quedan más claras que si se hiciera directamente sobre sistemas MBR.

Como primer paso para incorporar tareas complejas en sistemas de tests y para estudiar las capacidades de realizar adaptación sobre problemas se ha diseñado un nuevo tipo de ítems denominados *ítems compuestos*. Éstos agrupan una serie de ítems componente, como también sucede con los testlets. Los ítems componente pueden ser *reales*, si están directamente asociados a preguntas; o *virtuales*, si se asocian a evidencias de tareas complejas. Los ítems compuestos tienen una curva asociada que representa la probabilidad de responderlos correctamente dado un valor del conocimiento. De manera similar a la CCI tradicional esta curva se denomina CCIC y se calcula a partir de las CCI asociadas a los componentes. Estas curvas permiten aplicar los mecanismos de los TAI para selección adaptativa de ítems compuestos.

El uso de la TRI para realizar la adaptación en sistemas MBR es posible gracias a la analogía identificada en el capítulo anterior, la cual es extendida con los ítems compuestos. De esta forma, las restricciones son equivalentes a los ítems componente y los problemas a los ítems compuestos, los cuales son el sujeto de la adaptación. Para aplicar los mecanismos de la TRI es requerida una extensión de la estructura básica de los sistemas MBR. En el modelo de dominio, cada restricción se extiende mediante su CCR y los problemas con su correspondiente CCP, equivalente a la curva del ítem compuesto en sistemas de tests. Además, se añade una agrupación conceptual, similar a la utilizada previamente en algunos sistemas MBR, que agrupa restricciones y problemas y se utiliza para crear nuevas estrategias instructivas. En el modelo del alumno se sustituye el nivel basado en heurísticos por la estimación proporcionada por la TRI para cada concepto del modelo del dominio. La tercera componente básica extendida es el módulo pedagógico que incorpora las estrategias instructivas que utilizan el resto de elementos extendidos.

El mecanismo de evaluación sumativa del capítulo anterior posee el problema de que está diseñado para sesiones concretas por lo que la información asociada a la evolución del conocimiento no se tiene en cuenta. Para solventar esto, se propone una forma de hacer la traza del conocimiento en MBR mediante la introducción del concepto

de CK-sesiones. Una CK-sesión es una agrupación de evidencias donde se considera que el conocimiento del alumno no tiene cambio significativo. De esta forma, se puede aplicar la evaluación sumativa para obtener el conocimiento de un mismo alumno en diversos momentos en el tiempo mientras éste usa normalmente el sistema. Este nuevo mecanismo propone una forma de construir la matriz de rendimiento del alumno para aplicar la evaluación sumativa. Además, se discuten varios de los métodos posibles para realizar la agrupación de evidencias, entre los cuales, se menciona el establecimiento de un umbral que discrimina el tiempo trascurrido entre sesiones, o la agrupación en intervalos de tiempo donde se desea tener una estimación del conocimiento.

La ventaja de este mecanismo es que permite realizar la calibración de las restricciones usando datos donde ha tenido lugar aprendizaje. Sin embargo, para usarse "en vivo" mientras el alumno está utilizando el sistema, tiene el problema de que no es capaz de modelar el efecto del aprendizaje y, por tanto, no es preciso. Para solventar esto se propone el uso de modelos de la TRI que tengan en cuenta el aprendizaje como el propuesto por Lee et al. (2008) y se da una guía de cómo la generalización del modelo de evaluación permitiría modelar el aprendizaje usando el refuerzo como evidencia. No obstante, estos modelos no suelen encontrarse en la literatura y parece que necesitan más investigación para alcanzar cierto grado de madurez. Es por ello que la aplicación y el estudio de éstos quedan fuera del alcance de esta tesis, la cual sólo trata con modelos tradicionales de la TRI. El resto de la propuesta formativa, por tanto, asume que se está usando un modelo de la TRI con tratamiento del aprendizaje y que éste se basa en el uso de las CCI, como cualquier modelo TRI tradicional.

Para realizar la formación del estudiante, se plantea el uso de una evaluación formativa que irá dirigida por unos objetivos de aprendizaje, los cuales serán evaluados en una evaluación sumativa final. Para estos objetivos se utiliza una generalización en forma de árbol sobre los elementos del modelo de dominio: las restricciones y los problemas. Esta generalización agrupa las restricciones en conceptos y los problemas en tipos, sobre los cuales, un profesor puede establecer mínimos del conocimiento que deben cumplirse. En base a las evidencias de cada concepto se puede generar una estimación sobre el conocimiento asociado, lo que permite evaluar el cumplimiento de los objetivos de aprendizaje.

Como eje central del modelo propuesto, para realizar la estrategia instructiva, la aplicación de los mecanismos de la TRI se realiza sobre los dos elementos sobre los que la adaptación recae: la selección de problemas y el refuerzo. Cada forma de adaptación realizado en el MBR tiene un método homólogo en el modelo combinado MBR + TRI, con la diferencia sustancial de la base bien fundamentada de estos últimos. Para realizar la instrucción se proponen un conjunto de métodos que permiten seleccionar adaptativamente, tanto el refuerzo a presentar, como siguiente problema. La selección se puede realizar en base a diversos criterios que pueden ir dirigidos por los objetivos de aprendizaje o realizarse como tradicionalmente se hace en la TRI. Además, una componente importante del refuerzo que el alumno debe recibir para saber en qué debe mejorar, es el modelo abierto en el que se muestren los objetivos de aprendizaje y el grado de consecución .

En base a los diferentes elementos de formación se proponen dos modos de funcionamiento principales que buscan proporcionar o evitar el aprendizaje. El uso del sistema sin aprendizaje es necesario cuando se calibran las restricciones o cuando se realiza la evaluación sumativa final, mientras que el uso de aprendizaje se corresponde a la evaluación formativa del alumno. Cada modo de uso se caracteriza por utilizar los

mecanismos de selección adaptativa más apropiados, en conjunción con una forma de agrupar en CK-sesiones y un método de recolección de evidencias. Un inconveniente de la propuesta es que para mantener la seguridad, se debería disponer de un mayor número de problemas diferentes, de forma que los problemas comprometidos durante el aprendizaje no se presenten de nuevo en la fase de evaluación sumativa final. Además, el modo de uso formativo todavía no ha sido implementado, puesto que requiere del uso de modelos de la TRI que traten el aprendizaje del alumno.

Para muchos de los mecanismos propuestos todavía no se ha probado su efectividad. En este sentido, la limitación principal de la propuesta en el ámbito formativo es que no se conoce todavía cuáles de las estrategias de instrucción propuestas son más efectivas en un entorno real, y en base a los diferentes modos de utilización propuestos. Puesto que se han reemplazado los heurísticos del MBR por un modelo probabilístico bien fundamentado, nuestra hipótesis es que el comportamiento del sistema será más efectivo con esta metodología y que esta es la ventaja principal de la propuesta. No obstante, es necesario realizar investigaciones futuras que demuestren el grado de cumplimiento de la hipótesis.

El uso de la TRI no sólo se queda en la selección adaptativa y en la evaluación del alumno. Además, puede usarse para determinar la idoneidad de las restricciones como instrumento de evaluación del conocimiento. Para ello se propone el uso de la función de información sobre las restricciones. De esta forma se pueden detectar aquellas restricciones que no tienen suficiente evidencia, que hayan sido incorrectamente codificadas, o que conceptualmente estén representando principios inadecuados. Esto puede aplicarse como herramienta de análisis sobre el modelo de dominio para filtrar aquellas restricciones inadecuadas y generar una evaluación más fiable. Además, la técnica también puede usarse sobre problemas para determinar problemas poco útiles, desde el punto de vista de la información que proporcionan.

Si recordamos el DBE presentado en la sección 3.4.2, la metodología presentada, tanto en el funcionamiento, como en la arquitectura extendida del MBR, es un caso particular de este marco de trabajo: ésta reduce el nivel de generalidad, ya que se aplica solamente a EIRP, pero no lo pierde completamente, puesto que puede aplicarse a múltiples dominios. La ventaja de usar el MBR radica en que se establecen unas pautas más concretas que permiten guiar la construcción de nuevos sistemas y que no requieren de estudiar qué va a ser el modelo del alumno cada vez, ni de determinar para cada sistema qué constituye una evidencia, pues ya está implícito con el uso de este paradigma.

De acuerdo al marco de trabajo DBE se puede establecer una correspondencia más específica que la mostrada en los ejemplos de uso y los ejemplos de uso del apartado 3.4.2.2 en las siguiente componentes: el modelo del alumno estaría compuesto por los elementos tradicionales del modelo del alumno en el MBR, junto con la agrupación conceptual propuesta basada en estimaciones del conocimiento mediante la TRI. Dentro del modelo de evidencias, las reglas de evidencia estarían formadas por la comprobación de restricciones violadas y satisfechas. El modelo de estado sería la unión del método de las CK-sesiones para agrupar las evidencias y los mecanismos de la TRI para actualizar el conocimiento en el modelo del alumno. Dentro del modelo de tareas, las tareas, representadas por los problemas, tendrían como características las CCP; y los productos de trabajo estarían representados por las soluciones proporcionadas. El modelo de presentación se corresponde a la interfaz del EIRP y el modelo de ensamblado es el EIRP en sí. Como parte de esta tesis también se ha diseñado un sistema

que agrupa todas estas características, como si fuera el modelo de ensamblado, el cual se explicará en la sección 6.5. La equivalencia con la arquitectura de cuatro procesos es exactamente igual que en el ejemplo de la sección 3.4.2.2 pero usando las técnicas bien fundamentadas de la TRI en lugar de los heurísticos.

Como se puede observar existe un número de combinaciones enorme derivado de los mecanismos mencionados asociados a las estrategias de formación, y las formas de agrupar en CK-sesiones. Esto unido con la existencia de otros mecanismos que todavía no se han identificado y que probablemente sean incluso más efectivos que los mencionados, abre un universo de posibilidades todavía por explorar.

# Parte IV

# Implementación

En esta parte se presenta en detalle cada una de las herramientas / aplicaciones que se han desarrollado o extendido como parte de la investigación realizada para implementar y probar el modelo teórico presentado en la parte anterior.

# Capítulo 6

# Herramientas implementadas

*¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil, nos aporta tan poca felicidad? La repuesta es esta, simplemente: porque aún no hemos aprendido a usarla con tino*

Albert Einstein (1879 - 1955)

**Resumen:** En este capítulo se detallan las características principales de las herramientas que se han desarrollado o aquellas sobre las que se ha realizado algún aporte relacionado con la implementación y prueba de los modelos expuestos en los capítulos anteriores.

Con el fin de implementar los modelos teóricos detallados en los capítulos 4 y 5, se han desarrollado diversas herramientas. Este desarrollo ha sido realizado mediante un enfoque desde lo más específico a lo más general (más conocido como *Bottom-up*). De esta forma, primero se desarrollaron herramientas específicas sobre elementos concretos para, a partir de éstas, abstraer elementos comunes y formular modelos más generales. Para cada una de las herramientas desarrolladas se mencionará el dominio educativo sobre el que se enmarcan, identificando el tipo de problemas y actividades que permiten resolver, y ubicándolas de acuerdo a la clasificación de Mitrovic y Weerasinghe (2009) que se explicó al principio del capítulo 2. Puesto que algunas de las herramientas desarrolladas no son las únicas en su dominio, donde corresponda, se hará un repaso sobre los trabajos existentes en el ámbito educativo en cuestión. Además, para cada herramienta construida, se explicará de forma concisa la funcionalidad de la misma y se comentarán las diferencias con las herramientas propias desarrolladas previamente, justificando su desarrollo con las aportaciones y el valor añadido respecto de las ya existentes.

También, se mencionará brevemente el conjunto de tecnologías empleadas para desarrollar las aplicaciones. En este sentido, los detalles se centrarán en la justificación de su uso y la aportación que cada una proporciona, tanto en relación a las características más relevantes sobre la puesta en práctica de los modelos teóricos, como a los detalles técnicos de innovación que han facilitado su desarrollo. Como característica común a todas ellas, se ha utilizado el lenguaje Java para la construcción de entornos

Web, debido a combinación de accesibilidad, disponibilidad y entorno multiplataforma de ejecución, que ambas tecnologías proporcionan.

Desde un punto de vista un poco más técnico que científico, se detallará el diseño de la arquitectura, desglosando cada una de las componentes y explicando su correspondencia o el efecto que tienen sobre el modelo teórico planteado en capítulos anteriores. Con respecto a este tema, se mostrarán ejemplos de la apariencia de las herramientas, conexión entre las diversas componentes, funcionamiento interno, y manejo de los modelos del alumno y del dominio, asociados al MBR. Como parte fundamental del modelo de dominio, se pondrá un ejemplo de las reglas utilizadas en cada herramienta mencionando las diferencias de implementación. En este sentido, dado que Ohlsson (1994) no impone ninguna restricción sobre cómo codificar y / o implementar las restricciones, las diversas formas de implementarlas son igualmente válidas, siempre y cuando se comporten de acuerdo al funcionamiento teórico, descrito en la sección 2.3.2.

## 6.1. OOPS

El sistema de resolución de problemas OOPS (Gálvez et al., 2007, 2009a,b), cuyo nombre es la abreviatura del nombre completo en inglés (*Object Oriented Programming System*), se centra en el dominio de la *Programación Orientada a Objetos* (POO), y concretamente en los fundamentos básicos de este tipo de programación. De acuerdo con la clasificación de Mitrovic y Weerasinghe (2009), el dominio de aplicación se enmarca dentro de los que se definen como bien definidos pero con unas tareas a desarrollar débilmente definidas. Esto es así puesto que en el que el espacio de soluciones de las actividades no está acotado y hay infinidad de formas de construir una solución a un problema.

Este sistema fue el primero que se construyó utilizando el paradigma del MBR en el grupo de investigación dentro del cual se enmarca la investigación aquí presentada. Se puede considerar como uno de los puntos de partida de esta tesis ya que de él nació la investigación aquí presentada. Originalmente, OOPS surgió a partir de un proyecto fin de carrera (Gómez y Guzmán, 2006) como un sistema de apoyo a la enseñanza en la asignatura de Elementos de Programación del primer curso de las tres titulaciones técnicas de la E.T.S.I. de Telecomunicación de la Universidad de Málaga. El objetivo era ayudar a paliar la dificultad que encontraban los alumnos de esta asignatura en el aprendizaje de los Fundamentos de la Programación Orientada a Objetos (FPOO), más concretamente en la abstracción de datos. OOPS proporcionaba un entorno que permitía a los alumnos aprender los conceptos sobre los que tenían más dificultades, mientras usaban el pseudolenguaje propio de la asignatura. Actualmente OOPS se ha dejado de usar con la extinción de la asignatura de los actuales planes de estudio. Si bien el pseudolenguaje ha desaparecido del panorama académico, no sería muy problemático adaptarse a otros lenguajes actuales como Java.

En el momento en que OOPS se comenzó a desarrollar había un número muy reducido dentro del dominio de POO, el cual se ha incrementado en la actualidad. Por ejemplo, ViRPlay (Jiménez-Díaz et al., 2005) se centra en las interacciones en programas escritos en Java. Este sistema trata de enseñar la ejecución de un programa usando un juego de simulación en un entorno virtual en 3D. Sin embargo, el sistema parece carecer de un modelo del estudiante y de estrategias de adaptación. SmallTutor (Morschel, 1993; Zekl y Morschel, 1994) se enmarca en el dominio de la POO usando el

lenguaje Smalltalk. Este sistema adapta la instrucción de planificación usando técnicas de IA, de acuerdo a una serie de conceptos que los estudiantes deben aprender y a un gráfico de los requisitos previos. En (Pillay, 2000) también se usan grafos conceptuales pero en este caso para describir el dominio de aplicación. Este trabajo propone una arquitectura genérica para los tutores inteligentes de programación y se centra en la inducción de soluciones de programas orientados a objetos utilizando técnicas de algoritmos genéticos. Aunque estos enfoques se centran en diferentes aspectos del paradigma orientado a objetos, ninguno de ellos utiliza el MBR.

En la literatura, sólo se han encontrado dos sistemas MBR para el aprendizaje de conceptos orientados a objetos, el primero, Collect-UML (Baghaei, 2006), se centra en el aprendizaje de temas de análisis y diseño UML sin tratar temas de programación. El segundo es J-LATTE (Holland et al., 2009), y se centra en la programación mediante Java. Aunque la interacción de J-LATTE es bastante similar a la que posee OOPS, el sistema se publicó en 2009, dos años después de que OOPS lo hiciera en (Gálvez et al., 2007). Los problemas que se pueden resolver están limitados a la elaboración del contenido de un método, que implemente cierta tarea, mientras que OOPS permite la construcción de clases completas con multitud de métodos. El enfoque de J-LATTE es más específico y enfocado a sentencias concretas, mientras que OOPS trata los conceptos más generales relacionados con la POO. Más información sobre estos dos tutores se puede ver en el apartado 2.3.7.1.

La arquitectura y diseño se realizó partiendo de los cuatro elementos estructurales básicos de los STI explicados en la sección 2.1. Por este motivo, la estructura de OOPS está compuesta de cuatro partes bien diferenciadas: la primera sería la **interfaz** Web que permite al alumno la resolución de problemas de forma visual sin tener que escribir código fuente directamente. Ésta dispone también de una componente de autoría a través de la cual los profesores pueden añadir problemas e incluir su solución en el sistema. Otra parte es el **modelo del dominio** que contiene la representación, realizada por los expertos en la materia, del conocimiento que se va enseñar. Esta parte es la encargada de almacenar los problemas y las restricciones. El **modelo del alumno** reúne los conocimientos del estudiante y se usa para adaptar el proceso de enseñanza a las capacidades de aprendizaje y al nivel estimado de cada alumno. Por último, el **módulo pedagógico** comprueba los errores del alumno y define las estrategias de enseñanza, las cuales establecen cómo mostrar la información al alumno durante la resolución de problemas y seleccionan el problema que debe ser presentado al alumno en cada momento.

### 6.1.1. Interfaz de OOPS

La herramienta OOPS posee una interfaz Web, de forma que los alumnos pueden aprovechar las facilidades que hoy en día proporciona Internet de accesibilidad y disponibilidad al poder utilizar este sistema desde cualquier lugar y en cualquier momento. Desde el punto de vista tecnológico, la herramienta está implementada utilizando Applets de Java y una arquitectura cliente-servidor inherente a éstos. Se optó por usar esta tecnología ya que, en el momento de desarrollo del sistema, era una de las pocas que permitía implementar una interacción compleja como la requerida en el tipo de problemas cubiertos en este dominio y que se detallará a continuación.

La figura 6.1 muestra el aspecto de la interfaz correspondiente a la primera versión de OOPS. En esta interfaz se le plantea al alumno la implementación de una clase

determinada con una serie de métodos y atributos, de forma que se tenga que aplicar los FPOO para resolver el problema. A través de ella, el alumno puede insertar declaraciones y sentencias mediante un mecanismo de selección y arrastre a partir de una lista de elementos disponibles. Los elementos que se proporcionan al alumno se corresponden a elementos particulares del pseudolenguaje creado para la asignatura para la cual fue diseñado OOPS. Aunque las sentencias que se ofrecen son un subconjunto del lenguaje completo necesario para desarrollar cualquier programa, éstas son las precisas para realizar los ejercicios que se plantean a los estudiantes que comienzan a aprender FPOO.

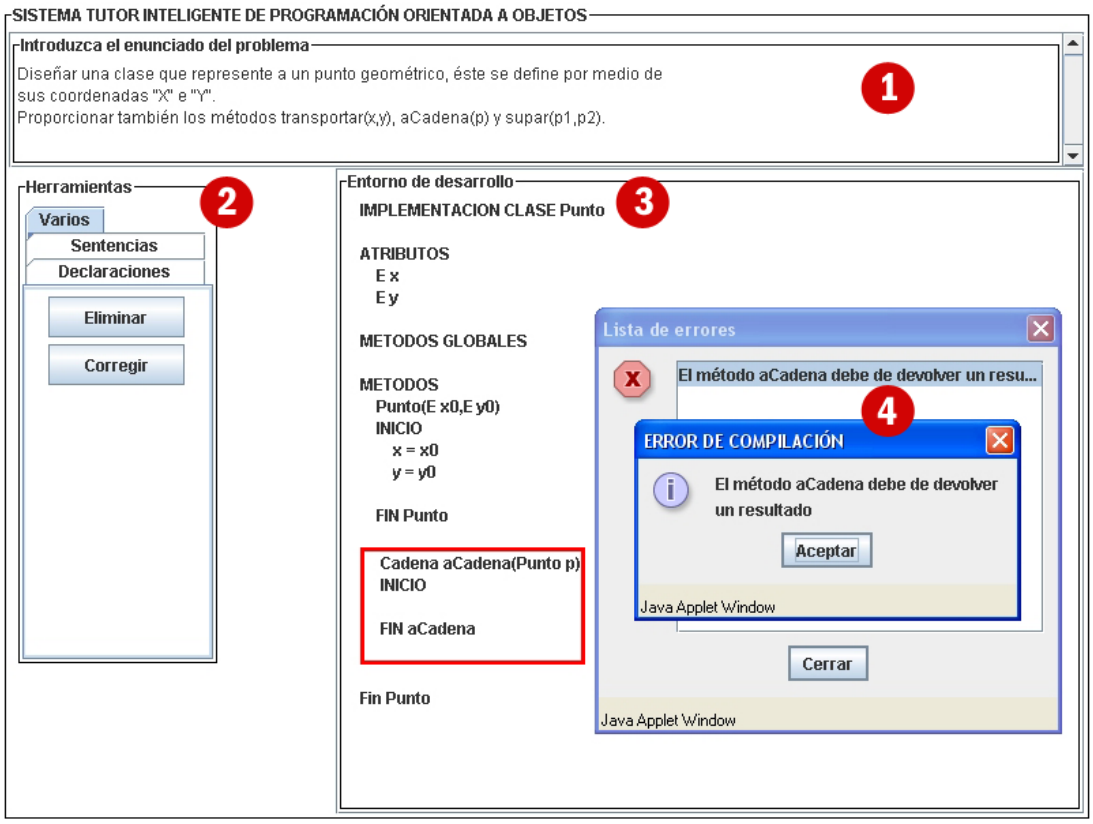



Figura 6.1: Interfaz de OOPS en su primera versión.

Las partes en las que ésta se estructura se enumeran a continuación:

- Etiquetada con (1), en la parte superior, se puede ver el enunciado del problema, donde se especifican los requisitos del problema que se pide al alumno y los métodos que debería implementar.

- La barra de herramientas, situada en el marco inferior izquierdo de la figura y etiquetada con (2) proporciona las acciones disponibles para ir construyendo una solución a través de tres pestañas:

  La primera, con el texto "Sentencias", contiene un botón por cada una de las sentencias que ofrece el pseudolenguaje tales como asignación, operaciones aritméticas, invocación a métodos, devolver valores dentro de las funciones, etc.

A continuación, y con el texto "Declaraciones", se encuentran el conjunto de botones que permiten declarar las diversas componentes del lenguaje. Se pueden crear o definir clases, atributos, métodos o variables.

Por último, con el texto "Varios", se proporcionan funciones para la eliminación de código y para compilar la solución actual. Nótese que los elementos de las dos primeras pestañas son arrastrables sobre el entorno de desarrollo. Cada vez que se desea añadir código al programa que se está construyendo, se debe hacer arrastrando el botón correspondiente. Cuando el arrastre se produce, y antes de soltar el elemento, la interfaz indica con cuadros verdes las posibles zonas en las cuales se puede soltar, las cuales variarán dependiendo del tipo de elemento y de la componente sobre la cual se desea añadir.

- El entorno de desarrollo, con la etiqueta (3) es la parte de la aplicación en la cual se va construyendo el código del programa que se desea realizar como solución. El código que se muestra aquí es interactivo; a la hora de añadir código a las diferentes partes del programa, éstas son resaltadas para que se pueda seleccionar dónde se desea añadir. El código que ya se ha añadido al programa también se puede editar por medio de un doble clic, acción que mostrará una ventana para cambiar los valores o el contenido que se esté editando.

- La ventana de errores, etiquetada con (4), es un elemento de la interfaz que sólo está disponible cuando el alumno ha seleccionado comprobar los errores que contiene el programa actual. Aquí se muestran la lista de errores que han detectado y que corresponden a las restricciones violadas.

Durante la resolución de un problema se ofrece al alumno, en todo momento, el conjunto completo de sentencias existentes en el sistema, tanto si son correctas, como si no. Con esto se consigue evaluar si la intención del alumno a la hora de programar en pseudolenguaje es correcta o no. Además, la interfaz evita el uso de ventanas desplegables y elementos para auto-completar código. Se ha considerado que esta característica no es adecuada cuando el usuario está aprendiendo, ya que muchas veces ayuda al alumno a evitar posibles errores que el alumno podría cometer, y por tanto disminuiría las capacidades instructivas del sistema.

Tras las diferentes pruebas realizadas con el sistema OOPS se recopiló un conjunto de recomendaciones para mejorar el sistema por parte de los alumnos participantes. Además, en la primera conferencia donde se presentó el sistema (Gálvez et al., 2007), también se recibió refuerzo por parte de expertos en el ámbito de los STI. Tales recomendaciones se tuvieron en cuenta para modificar la interfaz y la funcionalidad y producir una nueva versión, la cual se desarrolló en el marco de otro proyecto fin de carrera (Rubio et al., 2009). La interfaz de esta versión, que se puede ver en la figura 6.2, evolucionó para añadir nuevas componentes del pseudolenguaje tales como tipos de datos en los métodos, distinción entre la parte pública y privada de una clase, guardado de la solución para evitar pérdidas o seguir en sesiones posteriores, incorporación de una parte de gestión de las reglas asociadas al modelo de dominio del sistema, gestión de usuarios y corrección de diversos errores de funcionamiento.

De la misma forma que la esta versión fue mejorada, tras varios experimentos realizados, se tuvieron en cuenta las sugerencias de los alumnos y profesores para desarrollar OOPS 2.0, también en el marco de un proyecto fin de carrera (Sans et al., 2010). La interfaz de esta nueva versión, que se puede ver en la figura 6.3, introducía cambios

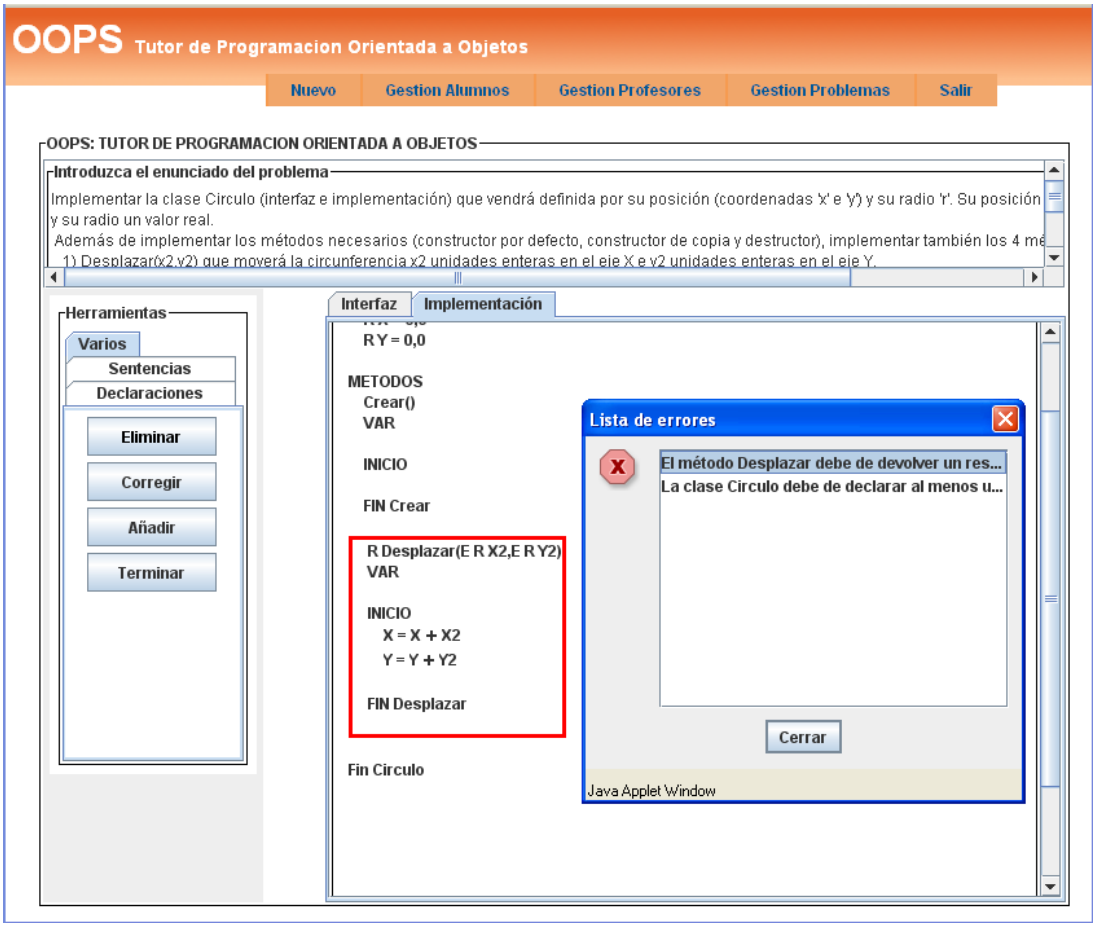



Figura 6.2: Interfaz de OOPS en la extensión de la primera versión.

radicales con respecto a sus predecesores. Concretamente, sustituyó los Applets de Java para la interacción con el alumno por el uso de las últimas tecnologías Web que proporcionaban una interfaz más vistosa, usable, e intuitiva. Dicha interfaz se construyó mediante Java Server Pages y una combinación de librerías Javascript que permiten emular las aplicaciones de escritorio tradicionales en la Web. Además, se reestructuró la arquitectura interna en un intento de abstraer una metodología para la construcción de sistemas tutores que combinaran el MBR con la TRI, y que acabaría evolucionando posteriormente en la herramienta de la sección 6.5. Por desgracia, esta versión no pudo ponerse en práctica ya que en el momento en el que se terminó, justamente la asignatura acababa de extinguirse del plan de estudios.

### Representación del conocimiento

Para que el sistema pueda inferir el conocimiento del alumno mediante la interacción con éste, es necesario representar la solución escrita en la interfaz mediante información que pueda ser procesada y evaluada. Concretamente, la representación interna se realiza de forma declarativa mediante una serie de hechos, los cuales se van obteniendo simultáneamente con la escritura del código en el espacio de trabajo del sistema.

Cada elemento del código tiene asociado uno o más hechos que pueden estar a su

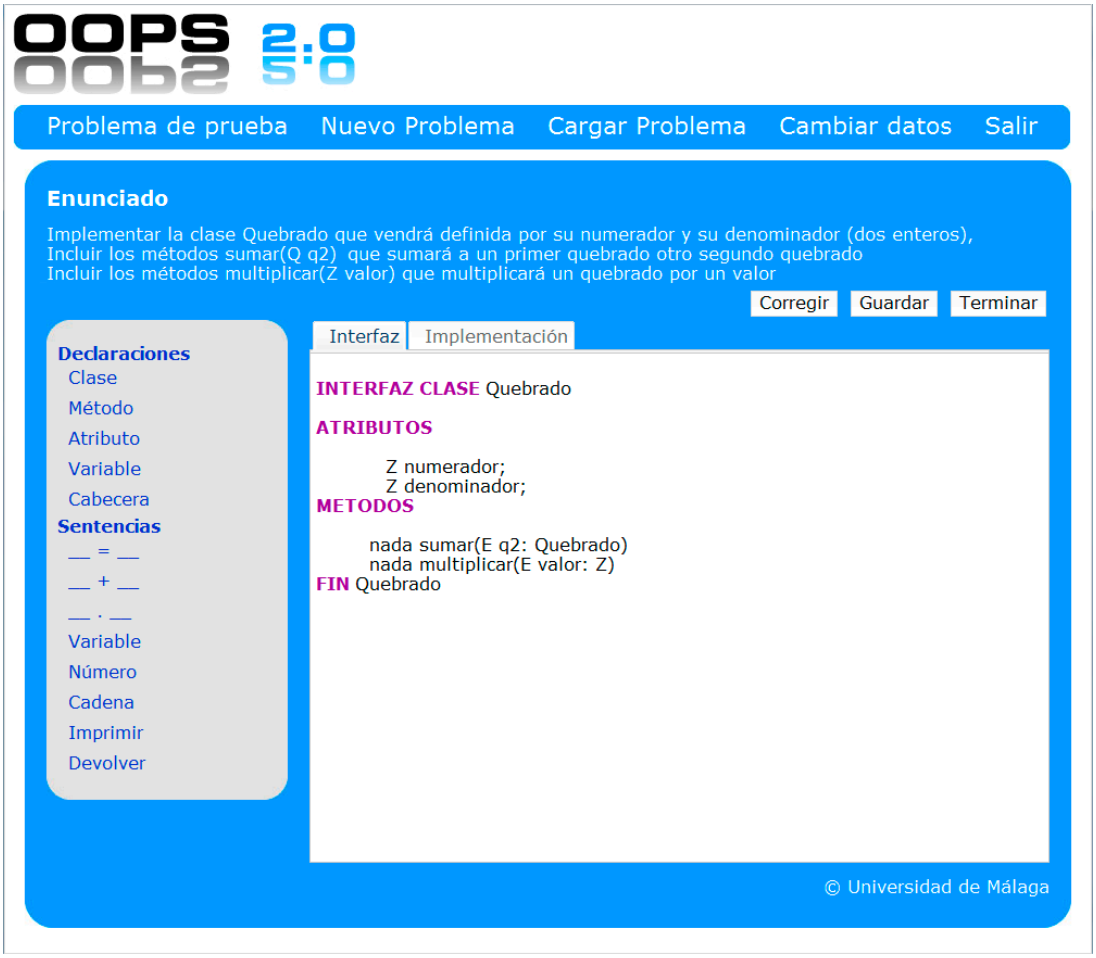



Figura 6.3: Interfaz de OOPS en la versión 2.0.

vez relacionados con otros mediante referencias. Esta relación está determinada por la situación de los elementos en el código: métodos de una clase, parámetros de un método, tipos de una variable, etc. Por ejemplo, si una variable está definida dentro de un método, existirán referencias entre el hecho que representa a la variable y el que representa al método. De esta forma, una solución propuesta por un alumno estará representada por un conjunto de hechos relacionados entre sí formando una estructura jerárquica en forma de árbol. En esta estructura, los nodos sucesores o hijos representan los elementos del lenguaje que están contenidos en otros y los nodos padre son los contenedores.

Un ejemplo de la representación de la información que se hace en el sistema se puede ver en la figura 6.4, correspondiente a una solución de OOPS 2.0. En este ejemplo, algunos hechos corresponden a elementos de la parte pública de una clase (recuadro verde de la izquierda) y otros a la parte privada. El hecho raíz es la clase $C$ que se está construyendo, la cual contiene los atributos $A_1$ y $A_2$; y un método $M_1$, el cual se define en la parte pública y se implementa en la parte privada. Este método, a su vez, tiene un tipo devuelto, un parámetro $P_1$ y una sentencia de asignación en su cuerpo.

Dentro de los hechos que se manejan por el sistema hay que distinguir tres tipos: a) Los predefinidos por el sistema que se corresponden a elementos propios del pseudolen-

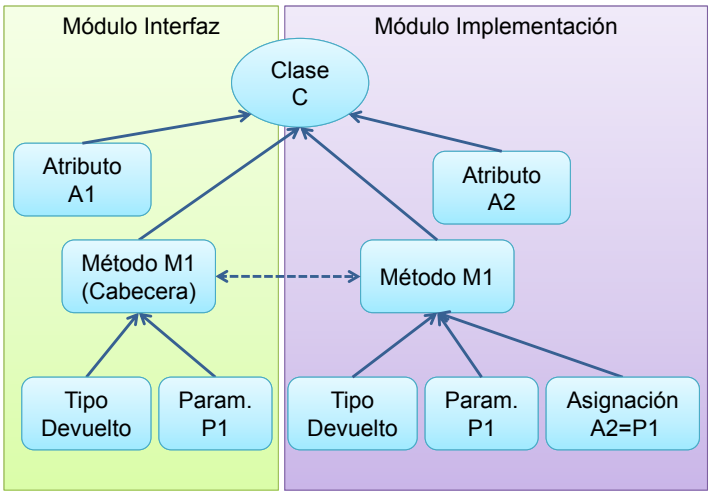



Figura 6.4: Ejemplo de la estructura jerárquica de hechos generados en OOPS.

guaje, tales como los tipos básicos (real, entero, cadena de caracteres, etc.) y métodos básicos de entrada salida para escribir por pantalla. b) Los incluidos por el profesor junto con el enunciado del problema. Éstos representan algoritmos predefinidos que el profesor proporciona al alumno para construir la solución del problema. c) Los que se van añadiendo dinámicamente conforme el alumno va elaborando la solución al problema, a través del espacio de trabajo. Éstos podrán verse modificados o eliminados dependiendo de si el alumno cambia o elimina el código asociado, afectando también a aquellos hechos que estén involucrados.

Los hechos a su vez, son representados mediante plantillas. Teniendo en cuenta la asociación entre código-hechos-plantillas existirán diferentes tipos de plantillas correspondientes a las diferentes sentencias del código: clases, métodos, atributos, tipos básicos, parámetros, operadores aritméticos, llamadas a métodos, etc. Todas estas plantillas tienen en común una serie de campos: un identificador unívoco, el nombre de la plantilla, una referencia a la plantilla en la que se define la sentencia asociada, y otros campos que opcionalmente pueden hacer referencia a operadores o tipos implicados en la expresión. Por ejemplo, un método tendrá un campo adicional para mantener las referencias a sus parámetros.

### 6.1.2. Modelo del Dominio

El dominio de la POO sobre el que se define el sistema es muy complejo. No hay una secuencia fija de acciones que llevan a una solución, ni tampoco hay una única solución para un problema dado. De hecho, hay un número infinito de secuencias y combinaciones que llevaría al usuario a una solución válida. En este aspecto, la utilización del MBR es útil para manejar el amplio espacio de soluciones ya que trabaja con la solución final, independientemente de los pasos que se hayan realizado para llegar a esa solución.

Siguiendo con la metodología explicada en el capítulo 2, las restricciones del dominio se han modelado como un conjunto de reglas de inferencia, cada una asociada al principio que no puede violarse y con un refuerzo determinado. En este primer sistema se utilizó el lenguaje CLIPS (*C Language Integrated Production Sistema*) (Wygant, 1989) que podía ser integrado fácilmente con la tecnología utilizada en el desarrollo

del mismo. Las reglas están escritas en colaboración con los profesores de la materia, que pueden considerarse como expertos. Un total de 86 restricciones / reglas fueron identificadas y agrupadas en seis categorías diferentes que se detallan en la tabla 6.1. En ésta, se muestra la siguiente información: la columna *Categoría* contiene un nombre simbólico de la categoría de agrupación de las reglas; la columna *Descripción* explica más detalladamente en qué consiste cada categoría; y la columna *Número* muestra el número de restricciones en esa categoría.

| **Categoría** | **Descripción** | **Número** |
|---|---|---|
| Tipos | Intentan buscar incoherencias entre los tipos de las operaciones aritméticas, llamadas a métodos, devolución de variables, etc. | 10 |
| Sintaxis | Relacionadas con los errores sintácticos y las infracciones de reglas gramaticales | 14 |
| Correspondencias y módulos | Comprueba la correspondencia entre los elementos de la interfaz (parte pública) de una clase y los elementos situados en la parte privada, como la implementación de una cabecera específica definida en la interfaz | 6 |
| Visibilidad | Trata los errores derivados de la ignorancia de las reglas de ámbito de las clases, métodos, variables, parámetros, etc | 15 |
| Formato de los nombres | Principios relacionados con los nombres de las variables locales, métodos, etc. Aquí se incluyen, tanto la definición duplicada de elementos, como las convenciones de nombrado de elementos mediante mayúsculas al comienzo de un nombre de clase y otras reglas de nomenclatura. Aunque supuestamente son sólo recomendaciones, se ha optado por considerarlas errores para fomentar la utilización de un convenio en el nombrado desde el comienzo del aprendizaje | 28 |
| Referencias perdidas | Errores que pueden producirse como consecuencia de la eliminación de código. Esto sucede cuando un estudiante elimina algún atributo, variable o un método que se usaba en el código pero todavía existen referencias a este elemento inexistente | 13 |

Tabla 6.1: Categorías de reglas del Modelo de Dominio de OOPS.

Dentro del sistema se pueden gestionar los dos elementos principales del Modelo de Dominio: los problemas y las restricciones. Los profesores con permisos pueden añadir nuevos problemas, editar los existentes, o borrar los que no se vayan a utilizar. Para la creación de un problema se debe proporcionar un enunciado y la que sería la solución ideal al mismo. En cuanto a la gestión de restricciones, sólo disponible en OOPS 2.0, los profesores pueden gestionar el contenido de las reglas, aunque para ello es requerido conocimiento sobre el lenguaje CLIPS en el que se deben escribir.

### 6.1.3. Modelado del alumno

El modelo del alumno de OOPS almacena las sesiones y cada intento realizado, junto con las restricciones violadas o satisfechas. Además, se infiere y almacena su nivel de conocimiento de manera automática. Éste se representa mediante un valor comprendido en el intervalo $[0, 10]$, distinguiendo tres categorías o subniveles: *bajo*, *medio* y *alto*, que dividen al intervalo anterior en tres partes de igual tamaño. El cálculo del nivel se realiza después de la última compilación del problema que se esté resolviendo, y consiste en comparar la puntuación obtenida por el alumno en los últimos problemas (puede configurarse el número de problemas), y el resto de estudiantes de su mismo nivel. Si la puntuación es superior, entonces se aumenta de nivel, en otro caso, continúa en el mismo.

### 6.1.4. Módulo pedagógico

Este módulo asiste al alumno durante su proceso de aprendizaje de dos formas diferentes. Por una parte, cuando el alumno aborda la realización del problema, se buscan deficiencias en su conocimiento mediante la aplicación de la inferencia MBR, y, posteriormente, seleccionando de forma adaptativa el ejercicio que debe resolver para poder actuar sobres las deficiencias detectadas.

#### 6.1.4.1. Selección Adaptativa de Problemas

Para seleccionar el problema más adecuado, se utiliza el nivel de conocimiento de éste, la puntuación de las reglas y la dificultad del problema. En las versiones iniciales de OOPS todavía no se había avanzado en la investigación y este mecanismo se realizaba con los valores estimados a partir de los heurísticos. Por tanto, la estrategia explicada en este apartado no es bien-fundamentada pero se detalla al ser la primera implementación del MBR realizada.

Cada regla tiene una puntuación que representa el nivel de error que supone violar la restricción asociada, el cual se encuentra en el intervalo $(0, 10]$, siendo un error más grave cuanto mayor es la puntuación. Este intervalo se divide en tres partes iguales representando diferentes niveles de gravedad: *no importante*, *medio* y *grave*.

Las puntuaciones iniciales fueron asignadas por los expertos (profesores de la asignatura) que se encargaron de recopilar las restricciones necesarias y conocían la gravedad de cada una. Posteriormente, la puntuación de cada regla se recalcula con cada compilación. Para ello se incrementa en uno el número de veces que se ha infringido la regla y se compara con la media de las demás reglas. Cuanto mayor sea la frecuencia del error cometido, mayor será su gravedad. Así, la puntuación $p_r(t+1)$ de la regla $r$ tras la compilación $t+1$ se puede expresar como:

$$p_r(t+1) = p_r(t) \times \frac{\delta}{\overline{p(t)}} \tag{6.1}$$

donde $\delta$ es el valor medio de la categoría a la que pertenece la regla, es decir, 1,65 si el grupo al que pertenece la regla es *no importante*, 5 si *medio*, y 8,35 si *grave*. $\overline{p(t)}$, la puntuación media de la regla, se calcula aplicando la fórmula de la ecuación 6.2, en la que $N$ es el número total de reglas.

$$\overline{p(t)} = \frac{\sum_{i=1}^{N} p_i(t)}{N} \tag{6.2}$$

La dificultad de un problema también se representa mediante un valor del intervalo $(0, 10]$. Para determinarla, cada vez que el alumno realiza la compilación, se cuantifican y almacenan los errores cometidos sumando la puntuación de las reglas correspondientes (ecuación 6.3). Los errores repetidos en diferentes compilaciones no se tienen en cuenta en este cálculo con el fin de evitar la acumulación de errores en la puntuación total (suma de puntuaciones de cada compilación). La fórmula heurística de la ecuación 6.3 calcula la dificultad de un problema mediante la comparación de la puntuación media del problema para todos los alumnos que lo han resuelto (ecuación 6.4) con la puntuación media de todos los problemas existentes (ecuación 6.5). La notación empleada significa lo siguiente: $p$ es el problema en cuestión; $t_{pi}$ es la puntuación del alumno $i$ en el problema $p$; $M$ es el número total de alumnos que lo han resuelto; y $P$ es el número total de problemas.

$$d_p = \frac{\overline{t_p}}{\overline{P_{TP}}} \times 5 \tag{6.3}$$

$$\overline{t_p} = \frac{\sum_{i=1}^{M} t_{pi}}{M} \tag{6.4}$$

$$\overline{P_{TP}} = \frac{\sum_{j=1}^{P} d_j}{P} \tag{6.5}$$

Una vez determinados estos parámetros, para seleccionar el problema más adecuado, hay que contemplar dos características: el nivel del alumno, para presentarle un problema conforme a éste; y sus deficiencias, para poder saber qué problema las puede tratar mejor. La selección del problema se realiza en varios pasos: primero se seleccionan todos los problemas del mismo nivel que posee el alumno, y se ordenan de mayor a menor dificultad. Si no existen problemas en el mismo nivel, es necesario ordenar de la forma adecuada el resto de problemas para seleccionar uno que sea apropiado. Para ello, se ordenan de menor a mayor, si el alumno tiene un nivel *bajo*, o de mayor a menor, si posee cualquiera de los otros dos niveles. Este mecanismo trata de proporcionar problemas dentro de la zona de desarrollo próximo del estudiante (Vigotsky, 1978).

#### 6.1.4.2. Resolución Asistida de Problemas

Siguiendo con el paradigma del MBR, para determinar las deficiencias del conocimiento del alumno, se requiere de un motor de inferencia. En todas las versiones de OOPS se ha utilizado JESS (*Java Expert System Shell*) (Friedman-Hill, 1997), ya que puede ser incorporado por las aplicaciones escritas en el lenguaje Java de forma sencilla. JESS se caracteriza por tener un tamaño muy reducido y una gran velocidad de ejecución, hecho más que interesante cuando se desarrolla una aplicación que funciona a través de Internet. Este motor de inferencia utiliza una versión mejorada del algoritmo RETE (Forgy, 1982) para determinar las restricciones violadas mediante un mecanismo de semejanza de patrones.

Inicialmente, en el motor de inferencia se inicializa con las reglas que representan las restricciones del dominio escritas en el lenguaje CLIPS asociado a JESS. Además, se introducen las funciones CLIPS auxiliares que son requeridas por las reglas para realizar comprobaciones. Cuando el alumno empieza un problema, se introducen los hechos predefinidos y aquellos que el profesor ha puesto a disposición del alumno para el problema en cuestión. Con la interacción del alumno, se van añadiendo al motor los hechos que representan la solución, siguiendo con el formato mencionado en el apartado 6.1.1. Una vez el alumno ha seleccionado comprobar los errores, la inferencia

del motor proporcionará a la interfaz los mensajes asociados a los errores cometidos, momento en el cual se "asiste" al alumno en la resolución del problema.

Para dar una idea más clara sobre la forma de implementar las restricciones en OOPS, a continuación se muestra un ejemplo de una de las reglas utilizadas. La regla, perteneciente a la categoría de restricciones sintácticas, comprueba que la declaración de las variables locales se realiza en el cuerpo de un método. Según se puede ver en la figura 6.5, la condición de satisfacción y de relevancia propias del MBR, etiquetadas con $Cr$ y $Cs$, respectivamente, se sitúan en el antecedente. En el consecuente de la regla se realiza el tratamiento del error definiendo distintos niveles de detalle de información e invocando a una función CLIPS que almacenará la violación en el modelo del alumno. El refuerzo a mostrar se adapta dinámicamente al nivel de conocimiento del estudiante, pudiendo ser muy precisa o aproximándose prácticamente a la que daría cualquier compilador actual, si el nivel es alto.

```
(defrule declared-var-wrong-sintax
   (status (compiling TRUE))
   (local-var (id ?id) (parent-id ?p-id))                          Cr
   (test (eq (is-valid-id ?p-id) TRUE))
   (test (neq (fact-slot-value (fact-id ?p-id) template) imple-method))   Cs
   =>
   (bind ?ids (create$ ?id))
   (bind ?error-text-easy (create$ "Una declaración de una variable local
                solo puede hacerse directamente en el cuerpo del método"))
   (bind ?error-text (create$ "Sintaxis incorrecta"))
   (bind ?rule-name (create$ (rule-name)))
   (throw-error ?rule-name ?ids ?error-text-easy ?error-text)
   (printout t "VARIABLE DECLARADA FUERA DEL MÉTODO" crlf)          Consec.
)
```

Figura 6.5: Restricción implementada en OOPS.

## 6.2. Simplex Tutor

El sistema *Simplex Tutor* (Gálvez, 2009), cuyo nombre viene del original en inglés, es un STI Web centrado en el dominio de la Optimización Lineal. Concretamente, en la aplicación del Algoritmo Simplex y el Algoritmo de las Dos Fases, una variante del anterior (Dantzig, 1940). Este dominio, atendiendo a la clasificación hecha por Mitrovic y Weerasinghe (2009), entra dentro de la categoría de bien definido, y dentro del cual, los problemas a resolver también son considerados bien definidos. Esto es así puesto que los algoritmos sobre los que se basa el sistema tienen una serie de pasos concretos y una solución compuesta por un número reducido y constante de elementos.

De la misma forma que en OOPS y siguiendo con el paradigma MBR, el dominio que se enseña es modelado por aquellos principios generales que no deben ser violados por ninguna solución a un problema dado. La implementación de las restricciones como reglas de inferencia también requiere del uso de un motor de inferencia en el sistema que permita detectar las restricciones satisfechas y violadas. El modelo del dominio ha sido construido en colaboración con la profesora Dra. Dña. Eva Millán, la cual imparte docencia en la E.T.S. de Ingeniería Informática, en la Universidad de Málaga, una asignatura donde se enseña la materia relacionada con los dos algoritmos. La

experiencia de ella ha servido de base como fuente de conocimiento experto para esta labor.

Existen diversas herramientas en el dominio al cual se refiere el sistema, tales como EPLAR (Millán et al., 1999), el cual es básicamente un software de resolución de problemas automático que carece de la posibilidad de usarse para resolver ejercicios; ILESA (López et al., 1998) es un STI Web que utiliza un módulo de resolución de problemas para ayudar al estudiante; y TAPLI (Millán et al., 2003), un sistema adaptativo que usa la TRI pero lo hace de una forma muy general sobre el resultado del problema (correcto o incorrecto) y generando un refuerzo que no está integrado como parte del mecanismo de diagnóstico, sino que está fuertemente ligada al dominio concreto.

Nuestro sistema difiere de los ya existentes en varios aspectos: a) ha sido construido usando el paradigma del MBR, lo que permite tener un mecanismo de refuerzo integrado como parte del modelo; b) tanto la evaluación como la adaptación se realizan de acuerdo a teorías bien fundamentadas y sobre principios fundamentales, permitiendo realizar una adaptación más precisa; c) es flexible en el sentido de que se pueden incluir fácilmente nuevos problemas en el sistema; d) ha sido diseñado de tal forma que las operaciones matemáticas son realizadas automáticamente por el sistema, lo cual reduce la sobrecarga cognitiva inherente al cálculo (Sweller et al., 1998), permitiéndole centrarse en la aplicación de su conocimiento para la resolución del algoritmo; e) en todo momento el paso de resolución y las diferentes opciones que se pueden aplicar para transformar la solución, o los posibles finales, están siempre disponibles.

### 6.2.1. Arquitectura del sistema

Este sistema se ha construido como una aplicación Web, que está implementada, desde un punto de vista más tecnológico, mediante la plataforma Java Enterprise Edition, siguiendo un patrón *Modelo-Vista-Controlador* (MVC) de diseño arquitectónico. La arquitectura de Simplex Tutor, esbozada en la figura 6.6, permitió establecer los primeros patrones comunes para la construcción de sistemas que se reflejaron en (Gálvez et al., 2008) y que darían posteriormente lugar al marco de trabajo de la sección 6.5. Ésta posee tres niveles que se corresponden a las siguientes capas: la capa de interfaz, la capa de negocio, y la de persistencia, cada uno implementado mediante diferentes tecnologías. Cada capa y su correspondencia con el modelo de evaluación principal de esta tesis se explican en detalle en los subsiguientes apartados de esta sección.

#### Capa de Interfaz o de Presentación

La capa de interfaz, utilizada principalmente para la presentación del problema y la interacción con el estudiante, ha sido desarrollada mediante el estándar de tecnología *Java Server Faces* (JSF) (Schalk et al., 2006). JSF es un marco de trabajo que simplifica la construcción de interfaces Web de usuario mediante el ensamblado de componentes reutilizables. Hemos elegido esta tecnología debido a la facilidad de aplicación para el desarrollo, sus resultados y la metodología que proporciona. Las interfaces que se obtienen con esta tecnología son muy similares a las suministradas por las aplicaciones de escritorio y su clara separación de la lógica de negocio permite dedicar mucho más tiempo y esfuerzo a la obtención del dominio y la planificación de estrategias de aprendizaje. Para presentar la información hemos utilizado la tecnología *Asynchronous JavaScript And XML* (Ajax), evitando así la necesidad de volver a cargar los conteni-

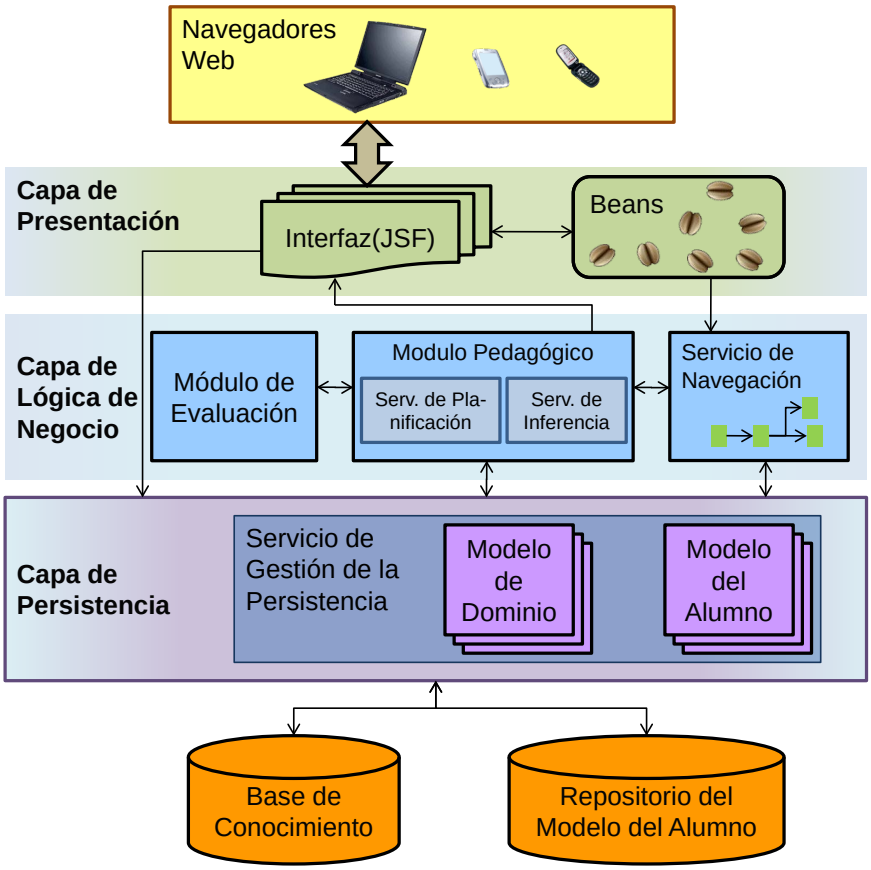



Figura 6.6: Arquitectura de Simplex Tutor.

dos de la página Web cada vez que se actualiza la información. Por ejemplo, nuestro tutor permite a los estudiantes solicitar ayuda si no están seguros de cómo seguir. En estos casos, el tutor realiza una petición asíncrona con el servidor usando Ajax, y como consecuencia, se muestra la ayuda adecuada.

La apariencia de la interfaz puede verse en la figura 6.7. Aquí se muestra un ejemplo donde el estudiante se encuentra en el primero de los tres pasos del algoritmo Simplex (véase la sección 6.2.2 para más información sobre los diferentes pasos). Este ejemplo puede dar una idea al lector del tipo de interacción que se puede realizar para resolver un problema. El ejercicio propuesto se puede observar en la parte superior izquierda de la figura y corresponde al problema de optimización lineal dado por el enunciado:

Maximizar la función objetivo

$$13X_1 - 2X_2$$

Sujeta a las restricciones:

$$2X_1 + 1X_2 \leq 8$$
$$1X_1 - 1X_2 \leq -5$$
$$X_1, X_2 \geq 0$$

En la figura se pueden identificar cuatro partes importantes: la primera, con la etiqueta (1), muestra el estado actual de la solución que se está construyendo; la si-

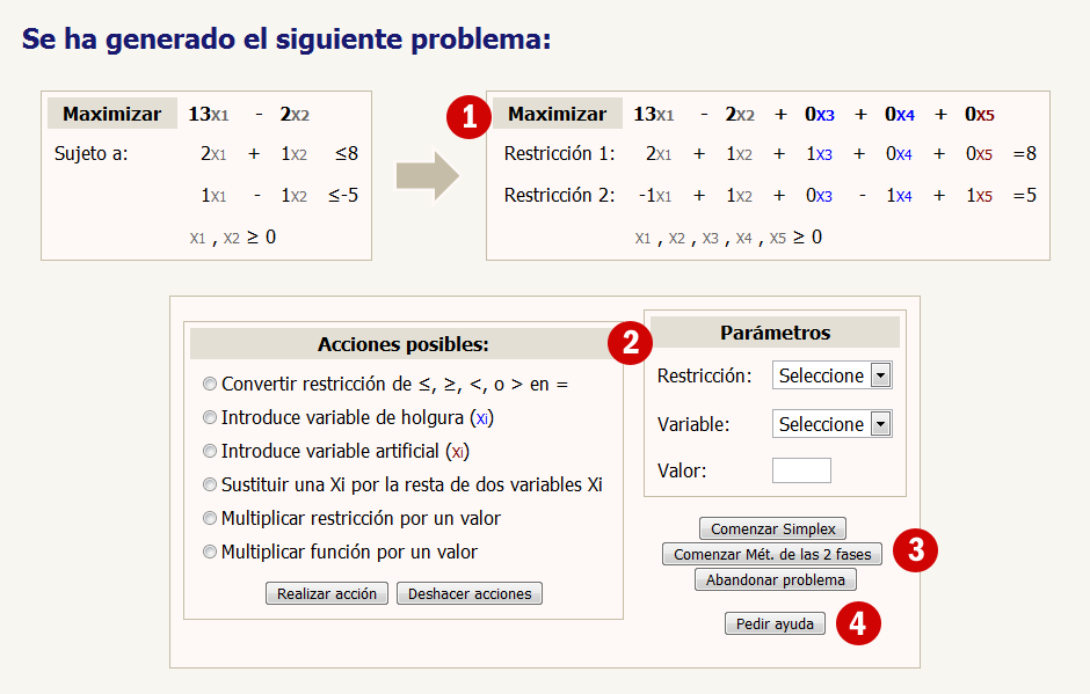



Figura 6.7: Interfaz de Simplex Tutor en el paso inicial de resolución.

guiente, etiquetada con (2), proporciona al usuario todas las acciones disponibles que pueden (o no) ser aplicadas para transformar el estado actual; la tercera parte, con la etiqueta (3), presenta todos los finales posibles para un paso o una fase del proceso de resolución y permite salir del problema; y finalmente (4), ofrece a los estudiantes un botón por si deseen solicitar ayuda, la cual consiste en información teórica relacionada con el problema particular que el estudiante está tratando de resolver. Aunque esta ayuda es inicialmente muy simple en comparación con las propuestas existentes en la literatura (Mitrovic y Martin, 2000; Mitrovic et al., 2002; Barrow et al., 2008), ésta podría ser personalizada de acuerdo a las diferentes necesidades del alumno usando los mecanismos de la sección 5.3.3.2.

En la etapa de resolución asociada a la figura, el usuario debe transformar la representación del problema inicial a su formato estándar (Dantzig, 1940). El estado actual de la transformación, o solución parcial, se muestra en la parte superior derecha de la figura. Como puede verse, el problema original ya ha sido transformado mediante la introducción de variables de holgura (Dantzig, 1940). Para lograr esto, el usuario utiliza las acciones disponibles con los parámetros o valores requeridos, representados por los recuadros a ambos lados de la etiqueta (2) y bajo los encabezados “Acciones posibles” y “ Parámetros”, respectivamente. Hay tres posibles finales para esta etapa del proceso de resolución de problemas: a) iniciar el algoritmo Simplex, b) iniciar el método de Las Dos Fases, o c) dejar el problema. Cada paso a su vez tiene los finales adecuados para pasar a otro paso o para finalizar la resolución.

Otro ejemplo de la interfaz, esta vez correspondiente a la versión en inglés del sistema, se muestra en la figura 6.8. Aunque esta figura refleja otro paso del proceso de resolución del algoritmo de las Dos Fases, se puede observar también las cuatro partes fundamentales descritas en el ejemplo anterior. Estas partes, si bien contienen diferente

información en todos los pasos, es un elemento común presente en cada uno de ellos, proporcionando al alumno con las herramientas necesarias para ir construyendo una solución al problema. De este paso cabe destacar la posibilidad de realizar combinaciones lineales sobre las filas de la matriz, las cuales son calculadas automáticamente sobre los valores existentes. Esta facilidad fue específicamente añadida al sistema para evitar la distracción y la sobrecarga cognitiva de tener que realizar los cálculos de las combinaciones, que se acentúa si se utilizan quebrados.

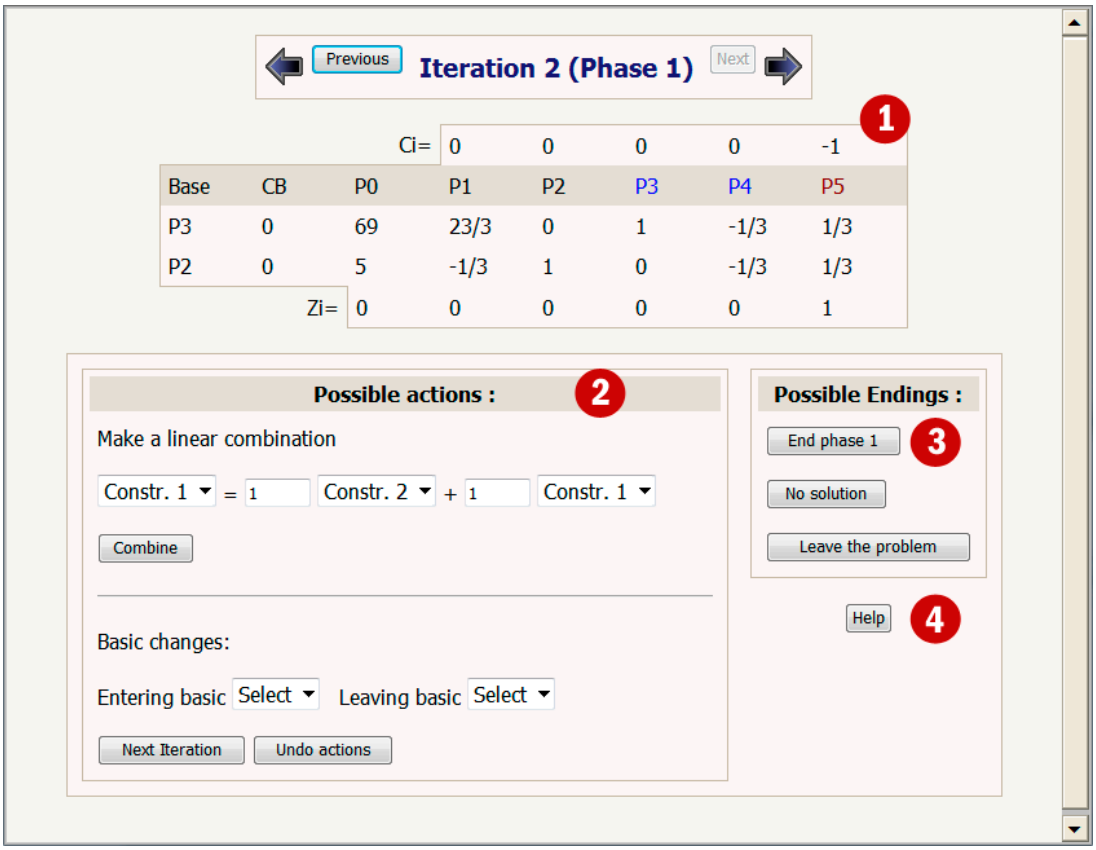



Figura 6.8: Interfaz de Simplex Tutor para el algoritmo de las Dos Fases.

En cada paso, la representación de la solución que se está construyendo es recogida en componentes Java, concretamente Beans, que permiten almacenar información y que desempeñan el papel de modelo temporal del estudiante. Mientras el estudiante está construyendo su solución, la información contenida en el correspondiente Java Bean cambia de manera dinámica con cada acción realizada a través de la interfaz.

### Capa de la lógica de negocio

Esta capa contiene aquellos componentes que suministran los servicios generales encargados de determinar la lógica de negocio de la aplicación, siguiendo un enfoque orientado a servicios (en inglés *Service Oriented Arquitecture* (SOA)). Esta capa procesa la información obtenida a través de la interfaz y aplica las estrategias de enseñanza de acuerdo a las inferencias que se hagan. Uno de los componentes más importantes de esta parte es el módulo pedagógico, que determina la corrección de la solución que se

está construyendo y controla la secuencia de elementos del currículo.

Con el fin de comprobar la corrección de la solución, el módulo pedagógico utiliza como entrada la respuesta dada por el estudiante en la capa de presentación y la almacena en el modelo del estudiante, actualizándolo de esta forma. La capa de gestión de la persistencia proporciona a este módulo las reglas asociadas al dominio. Éstas, junto con la respuesta del estudiante, se introducen en un motor de inferencia, accesible a través del servicio de inferencia, para determinar las restricciones violadas. Estas restricciones violadas también se registran en el modelo del estudiante como una representación a corto plazo de los errores y el conocimiento de los estudiantes.

Desde el punto de vista tecnológico, el servicio de inferencia incorpora el motor de inferencia JBoss Rules (Bali, 2009) para aplicar los principios del MBR en la evaluación de los estudiantes. JBoss Rules usa un lenguaje específico, llamado DROOLS (Bali, 2009), que proporciona una sintaxis intuitiva a diferencia de otros motores de uso general para la aplicación de sistemas basados en reglas (como JESS o Allegro Common Lisp). Esta sintaxis permite utilizar sentencias Java en las reglas que representan el modelo de dominio, haciendo que el motor sea más potente que los tradicionales y que sea más fácil el proceso de elicitación de esta base de conocimiento. Esta misma sintaxis permite además establecer un orden por defecto para mostrar el refuerzo de las restricciones: el orden de definición. Otros enfoques como el propuesto por Mitrovic (1998b) pueden ser igualmente simples de implementar haciendo uso de interfaces Java sobre los objetos que representan la violación de las reglas, funcionalidad que como se comentó en el apartado 2.3.4, no resulta fácil de implementar en otros entornos. El motor de inferencia permite a los sistemas que lo usen olvidarse de la eficiencia, ya que se encarga de compilar las reglas en redes RETEOO, una implementación del algoritmo RETE extendida y mejorada para sistemas Orientados a Objetos, que incorpora las mejoras propuestas por Doorenbos (1995). Además DROOLS es capaz de gestionar automáticamente las diferentes sesiones o, como se denominan en JBoss Rules, "Knowledge bases" (bases de conocimiento). Como valor añadido, la idoneidad de este motor para nuestro sistema es aún mayor debido al hecho de que JBoss Rules se ha desarrollado como una herramienta para aplicaciones Web.

Una vez que las inferencias se han hecho y las restricciones violadas / satisfechas han sido comprobadas, el módulo pedagógico pasa esta información al módulo de evaluación. Este componente determinará el nivel de conocimiento del alumno en los conceptos evaluados. Para este fin, el sistema cuenta con dos servicios alternativos, dependiendo de la forma en que se esté utilizando el sistema. Cuando el sistema se utiliza para fines de aprendizaje, se utiliza un servicio que implementa una función heurística que tiene en cuenta el número de veces que cada restricción se viola y se comparan los errores cometidos con los de otros estudiantes. Por otro lado, si se usa el sistema para evaluar, existe otro servicio que utiliza la TRI para determinar el nivel del estudiante. Por último, el módulo de evaluación registra las nuevas estimaciones del conocimiento en cada modelo del alumno usando la capa de gestión de persistencia.

La capa de negocio también es responsable de la secuenciación de problemas, decidiendo el siguiente problema a presentar durante una sesión de uso. Para implementar esta funcionalidad se han utilizado reglas de navegación JSF. Esta tecnología permite la definición de este tipo de reglas, que gestionan las transiciones entre los componentes de la interfaz JSF de la capa de presentación. Teniendo en cuenta las estimaciones del conocimiento del estudiante previamente inferidas, el servicio de planificación del módulo pedagógico puede decidir el problema más apropiado a mostrar a continuación,

o terminar la sesión de resolución de problemas. Los procedimientos teóricos que toman la decisión de dejar (o no) la sesión de resolución de problemas se inspiran en las decisiones análogas a los TAI.

En cuanto a la decisión sobre el siguiente problema a mostrar, se puede tomar de dos maneras diferentes, o bien adaptativamente, o bien siguiendo una secuencia preestablecida. La primera opción consiste en determinar de forma dinámica el tipo más adecuado de los problemas de un estudiante concreto de acuerdo a los mecanismos de selección de la TRI. Cada problema tiene un nivel de dificultad que es utilizado para identificar su idoneidad respecto del conocimiento estimado del modelo del estudiante. En la versión existente todavía no se utilizaba la dificultad en base a las restricciones y el mecanismo de evaluación no estaba completamente integrado.

La segunda forma de decidir el siguiente problema que se muestra sigue una secuencia preestablecida de problemas. Esta alternativa se ofrece para aquellos profesores que deseen utilizar el sistema para evaluar al alumno sobre unos conceptos específicos. De esta manera, los profesores pueden definir todos los datos del problema, es decir, seleccionando todos los valores para los que quieren evaluar al estudiante, o bien pueden elegir una secuencia de tipos de problemas a presentar.

Una vez que el tipo de problemas se ha seleccionado, el problema puede ser generado aleatoriamente por el sistema. Esta generación se inspira en el mecanismo presentado por López et al. (1998); Roberts y Engel (2001), y se basa en la generación aleatoria de un conjunto de coeficientes de la función objetivo. Mediante la aplicación de ciertas combinaciones lineales sobre estos valores, se pueden obtener los coeficientes de las restricciones que forman parte del enunciado en cada tipo de problema Simplex (no confundir con las restricciones MBR).

### Capa de persistencia

Esta capa provee de los servicios encargados de almacenar y gestionar los modelos del estudiante y del dominio, el registro de la actividad y cualquier información relevante. Esta funcionalidad se ha implementado mediante la consistencia y la abstracción proporcionada por el marco de trabajo JPA (*Java Persistence API*) (DeMichiel y Keith, 2006). Éste permite la gestión de bases de datos relacionales utilizando el lenguaje Java, y asegura la consistencia y la coherencia entre los datos manejados por la aplicación y la información almacenada en la base de datos.

El sistema mantiene un registro que contiene información sobre todas las acciones llevadas a cabo por el estudiante durante las diferentes etapas de la resolución de problemas. Como se puede observar en la figura 6.9, esta información se almacena y se organiza mediante una jerarquía en la que el elemento superior es la sesión. Cada sesión contiene los problemas que el estudiante trata de resolver desde el momento en el que entra en el sistema hasta el momento en que cierra la sesión. Los problemas están organizados en intentos. Un intento contiene cada solución que el estudiante haya presentado. Por último, una solución dispara una o más restricciones de dominio, siendo éstas satisfechas o violadas. El tutor Simplex hace que toda esta información disponible al profesor a través de diferentes tipos de informes sobre el estudiante.

El modelo del estudiante de Simplex Tutor es creado y actualizado en este nivel. Este modelo se compone principalmente de dos partes. La primera es el conjunto de restricciones que el estudiante ha violado o satisfecho durante la resolución de problemas. Como se mencionó antes, estas restricciones representan los principios que el

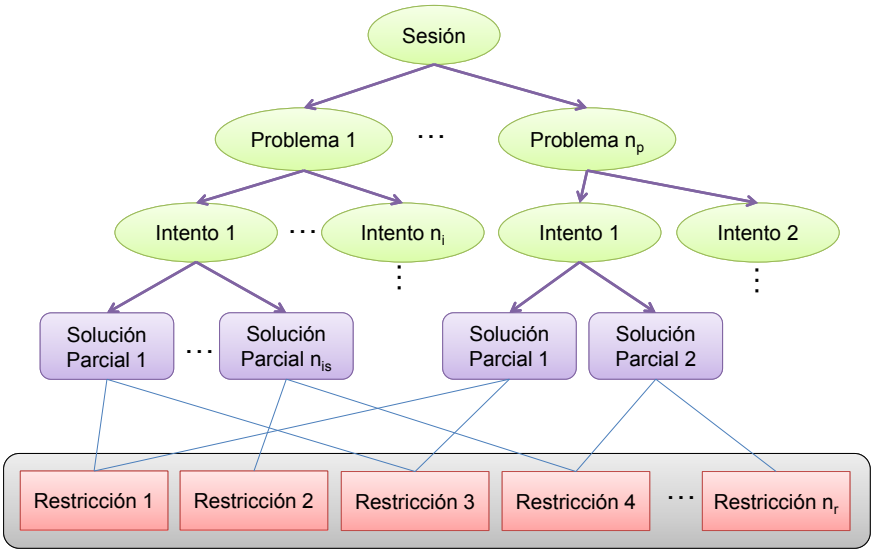



Figura 6.9: Estructura de la información del estudiante registrada en Simplex Tutor.

estudiante no ha aprendido correctamente y, también, aquellos en los que no comete errores. La segunda parte del modelo es el conjunto de estimaciones cuantitativas del conocimiento del estudiante inferidas por el módulo de evaluación.

La capa de persistencia también gestiona el modelo de dominio, el cual, de acuerdo a la formulación del MBR, contiene los conjuntos de restricciones y los problemas que se pueden presentar al estudiante. En cuanto a las restricciones, a pesar de que la mayoría de los sistemas basados en reglas las almacenan en un archivo de texto, hemos optado por una base de datos relacional para este propósito, ya que esta estructura permite una mejor gestión de las mismas al ser posible manipularse desde la misma aplicación. En la base de datos, se han agrupado restricciones en tres categorías diferentes, según la fase en la que se aplican, es decir, transformación del problema en Formato Estándar (4 restricciones), iteraciones del algoritmo (5 restricciones) o finales del algoritmo (9 restricciones). Esto supone un total de 18 restricciones, un número reducido en comparación con otros sistemas MBR. Esto se debe principalmente a la naturaleza bien definida del dominio, el cual requiere unos conocimientos muy específicos para aplicar el método Simplex y, por tanto, cuenta con un conjunto reducido de principios.

Para comprender mejor la potencia del conjunto de tecnologías empleado en este sistema, se muestra un ejemplo de implementación de una restricción en la figura 6.10. Las tres componentes de una restricción en el MBR se pueden ver en las etiquetas *Cr* para la condición de relevancia, *Cs* para la condición de satisfacción, y *Consec.* para el consecuente de la regla, que contiene las acciones a tomar para el tratamiento del error detectado. En esta regla se puede observar la potencia del lenguaje respecto a las implementaciones realizadas en OOPS: se pueden usar objetos Java como contenedores de la información directamente en el motor de inferencia. Además, sobre estos objetos se puede usar código Java, tanto en el consecuente (llamada a `System.out.println` o al método `nuevoError`), como en el antecedente (invocación al método `isBaseVectorEntrada`).

Además, el uso de Java Beans pertenecientes a la capa de interfaz, no sólo hace que los mecanismos de persistencia sean más fácil de manejar que otros enfoques, sino

```
rule "Base vector entrada"
    no-loop true
    when
        t : SimplexTablaDatos(it1:iteracion, finalFase1==false)
        t2 : SimplexTablaDatos(it2:iteracion -> (it2 == it1 + 1))
        eval (t2.isBaseVectorEntradaMal(t))
    then
        System.out.println("No se ha formado una base en el vector" +
            "de entrada (iteración "+ it2+ ")");
        errores.nuevoError(new ReglaViolada("reglaBaseVectorEntrada"));
end
```

$C_r$

$C_s$

Consec.

Figura 6.10: Ejemplo de regla en Simplex Tutor.

que también facilita la comunicación entre las capas de interfaz y de negocio. Nuestro EIRP utiliza las mismas estructuras para la representación de la solución en la interfaz y como hechos que se pueden introducir en el motor de inferencia. Esto significa que no se requiere ninguna transformación sobre la representación de la solución para comprobar las restricciones violadas en el motor de inferencia, ni siquiera para almacenar los resultados de esta comprobación en la base de datos.

### 6.2.2. Modo de funcionamiento

El escenario más habitual de funcionamiento de Simplex Tutor se describe a continuación (ver el diagrama de flujo de la figura 6.11). La primera vez que un estudiante entra en el sistema, la capa de presentación solicita al módulo pedagógico la construcción e inicialización del modelo del estudiante. A continuación, el módulo pedagógico realiza una solicitud al servicio de planificación para mostrar el problema adecuado (esto es la selección de problemas). Como se ha explicado anteriormente, esto puede hacerse de forma adaptativa, de acuerdo con las estimaciones del modelo del estudiante, o seleccionando el problema de una secuencia fija. Estos datos son enviados a la capa de persistencia para almacenar el intento correspondiente al problema.

Una vez presentado el problema, el estudiante debe llevar a cabo tres pasos diferentes con el fin de alcanzar una solución (proceso de resolución de problemas). Al igual que pasaba con la herramienta NORMIT, explicado en la sección 2.3.7.4, cada paso se corresponde con una tarea que se debe realizar. En primer lugar, el problema original debe ser transformado a Formato Estándar (figura 6.7), que se requiere para iniciar el algoritmo Simplex y el método de Dos Fases. Una vez hecho esto, el sistema rellena la tabla correspondiente y se inicia el segundo paso: un proceso iterativo donde la tabla tiene que ser transformada por el estudiante, con el fin de alcanzar un estado de solución.

La última parte de la resolución del problema se produce cuando el estudiante identifica un estado final en el algoritmo. Entonces, el tipo de solución debe ser identificado y los valores óptimos resultantes interpretados a partir de la tabla final. La identificación del tipo de solución se hace seleccionando uno de los posibles finales (solución única, soluciones alternativas, o solución ilimitada) en el paso anterior. Cuando se hace esto, el sistema muestra la última tabla (o las dos últimas tablas en el caso de solución ilimitada) como se muestra en la figura 6.12. Posteriormente, el alumno debe rellenar los huecos con el valor óptimo para la función objetivo y el punto (o par de puntos en un problema de solución ilimitada), que hace que las restricciones del problema se satisfagan.

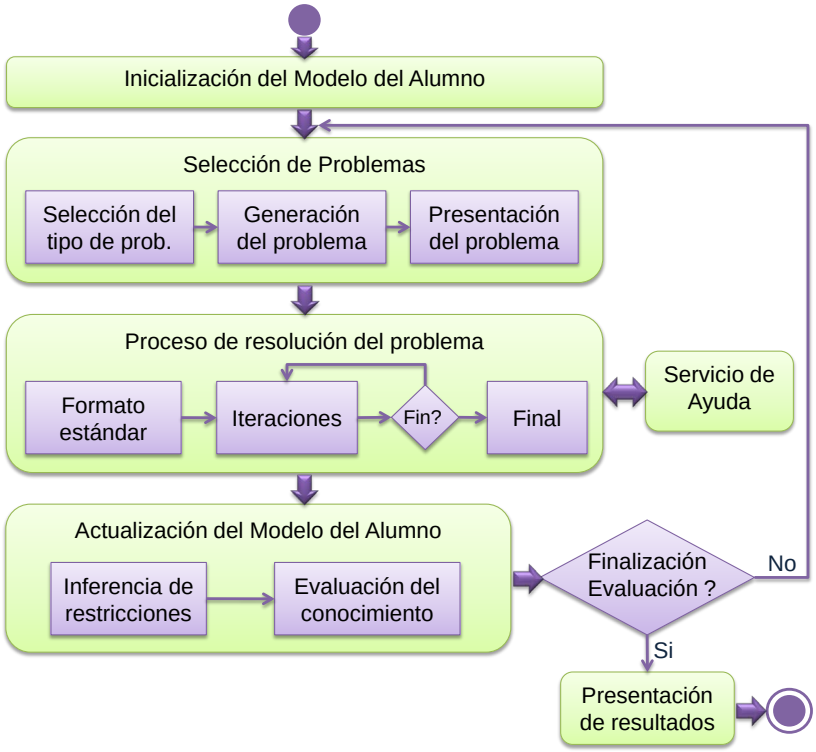



Figura 6.11: Diagrama de flujo del modo de operación de Simplex Tutor.


Figura 6.12: Paso final del proceso de resolución de un problema.

Cada acción tomada a lo largo de estos tres pasos modifica el correspondiente Java Bean que representa el estado solución. Al final de cada paso, la solución se envía a la capa de persistencia y al servicio de inferencia (en la capa de negocio), donde se realiza el proceso de inferencia para determinar las restricciones violadas. Estos errores se mostraran al estudiante a través de la interfaz y serán enviados, de la misma forma, al servicio de evaluación, que permitirá actualizar el modelo del estudiante con las nuevas estimaciones del conocimiento.

A continuación, el servicio de planificación decide el siguiente paso a llevar a cabo, genera el caso de navegación correspondiente y comunica esta decisión al servicio de navegación, pasando al siguiente problema o finalizando la sesión. Por último, el servicio de navegación invocará al módulo adecuado, ya sea para pedir un nuevo problema o para mostrar al alumno la información de su conocimiento. Nótese que, en el primer caso, todos los pasos descritos en esta sección se repetirán hasta que se satisfagan los criterios de finalización.

## 6.3. Siette

El sistema Siette (*Sistema Inteligente de Evaluación mediante Tests para la Tele-Educación*) (Conejo et al., 2004; Guzmán, 2005; Guzmán et al., 2007) es un entorno Web para la creación y mantenimiento de bancos de ítems, y realización de tests por ordenador. Siette ofrece un extensísimo abanico de funcionalidades que, unido a los fundamentos teóricos en los que se basa y los diversos estudios empíricos que avalan su efectividad en diferentes áreas del e-learning, lo convierten en una de las herramientas más completas en su categoría. En relación con la evaluación, Siette implementa las dos teorías de evaluación más importantes en sistemas de tests, explicadas en el capítulo 3: la TCT y la TRI. Mediante esta última, Siette implementa una de sus características más importantes que consiste en la posibilidad de realizar TAI.

En esta sección se detallan de forma general las características de este sistema y se explica la extensión realizada en el mismo como parte de la implementación del modelo teórico de evaluación de dominios procedimentales mediante tests, detallado en la sección 5.1. Además, las características evaluativas de Siette han sido utilizadas para la implementación de un marco de trabajo genérico que combina el modelo de evaluación propuesto y que se detallará en la sección 6.5.

Aunque en la primera versión (Ríos et al., 1998) se implementaba un mecanismo para realizar TAI (Ríos et al., 1999a,b), la versión actual surge de una implementación posterior que parte de cero. Esta nueva implementación, cuya interfaz se puede observar en la figura 6.13 utiliza una nueva base tecnológica y extiende el sistema con un nuevo modelo de evaluación basado en la TRI (Guzmán y Conejo, 2004a,b).

### 6.3.1. Contenidos en Siette

Los contenidos en Siette se estructuran en **asignaturas** asociadas a un dominio educativo concreto. Cada asignatura viene representada por su árbol curricular de **temas**, los cuales tienen asignados un conjunto de **ítems** que permiten evaluar los conceptos asociados. Los ítems que Siette posee pueden ser de distintos tipos o categorías:

- *Ítems básicos*: Es una categoría que engloba los ítems más simples del sistema. A partir de ellas se puede definir cualquier otro tipo de ítems de los existentes en el sistema. Dentro de esta categoría se pueden encontrar varios tipos:
  - *Ítems de múltiple opción y respuesta simple*: Son aquellos en los que los alumnos tienen que seleccionar una única opción de entre un conjunto de posibles respuestas, pudiendo dejar el ítem sin señalar ninguna respuesta.
  - *Ítems de múltiple opción y respuesta múltiple*: Son similares en formato a los anteriores, pero en este caso los alumnos pueden seleccionar más de una respuesta de entre el conjunto de alternativas.


Figura 6.13: Interfaz de la versión actual de Siette.

- *Ítems de respuesta corta*: En este tipo de ítems los alumnos tienen que dar una respuesta escrita, relativamente corta, dado cierto enunciado. Las opciones que pueden darse como respuesta se representan mediante patrones de respuesta, los cuales se pueden detectar aplicando al texto introducido diversos mecanismos:
  - Patrón correspondencia: La respuesta suministrada por el alumno debe coincidir exactamente con los patrones de respuesta almacenados.
  - Patrón Siette: Permite la utilización de expresiones regulares en cada patrón para detectar la opción a la que se corresponde la respuesta dada.
  - Expresiones regulares Java: Es un mecanismo análogo al patrón Siette pero ofrecen la posibilidad de incluir expresiones más complejas que se construyen mediante sentencias Java.
  - Patrón OpenMath: Comprueba mediante una heurística la equivalencia de fórmulas algebraicas de una variable basándose de un muestreo de puntos.

- *Ítems generativos*: Son plantillas en las que los parámetros que las definen se generan a la hora de ser presentadas. De esta forma, los valores del enunciado pertenecen a instancias diferentes pero el ítem es el mismo. Estos ítems se implementan mediante lenguajes embebidos en HTML como *Java Server Pages* (JSP).

- *Ítems externos*: Éstos están ubicados fuera del sistema, y por lo tanto, su presentación no está controlada directamente por Siette, sino que alguna aplicación externa se encarga de realizarla. A pesar de que no residen en la base de conocimiento, se almacena cierta información como la dirección Web donde se encuentra

ubicado ese ítem, los parámetros necesarios para invocarlo, el conjunto de posibles respuestas que ese ítem devolverá a Siette, y las propiedades psicométricas del mismo.

- *Ítems Autocorregidos*: Son ítems que permiten presentar una interacción más compleja que la que ofrecen el resto. Para ello, utilizan Applets o Flash para presentar la información al alumno. La corrección de la respuesta se hace en el Applet u objeto flash y se manda a Siette solamente solución.

Además de los diferentes tipos de ítems, éstos pueden personalizarse con multitud de opciones, tales como la temporización de un ítem particular; proporción de pistas para resolver el ítem; utilización de refuerzo sobre las respuestas dadas con el fin de ayudarles a comprender sus errores y mejorar su aprendizaje; y otras opciones de presentación. Incluso puede definirse el siguiente ítem a mostrar en base a las diferentes opciones de respuesta, dando lugar a diferentes ramas y convirtiendo los diferentes ítems en ramificados. Así pues, se puede definir una estructura de ítems interconectados que se quiera usar para evaluar al alumno.

Asociados a los temas de la asignatura, se encuentran los **tests**, los cuales se pueden definir sobre varios temas o, de forma general, sobre la asignatura. Los tests en Siette pueden personalizarse en base a múltiples criterios y con diversas opciones:

- En cuanto a la evaluación, se puede personalizar el método concreto, bien usando alguno de los métodos de la TCT como el porcentual o el por puntos; o bien mediante la TRI, pudiendo usarse modelos discretos o continuos. La evaluación puede incluso repetirse a posteriori si se ha encontrado algún error en los ítems y se ha cambiado algo.

- En relación con la selección de ítems, se pueden establecer que sea la propia de los TAI, de manera adaptativa; se pueden seleccionar aleatoriamente; aleatoriamente pero en base a una ponderación que determina la proporción de ítems de cada tema a mostrar; en base a la más apropiada de acuerdo con el método del repaso espaciado (Baddeley, 1997); mediante un conjunto de filtros; por pertenencia o no pertenencia a categorías; por autor; y muchas opciones más.

- Respecto a la seguridad, se puede restringir el uso en los tests mediante contraseña, a determinados grupos realizados por el profesor, por fecha, por sistema de procedencia, por un patrón IP sobre la máquina en la cual se está realizando el test, o mediante un área geográfica concreta.

- El test puede realizarse de manera individual o bien de forma colaborativa. Para ello se proporciona un entorno en el que los estudiantes pueden colaborar en la resolución de tests.

- Otras características: se pueden personalizar multitud de otras opciones para el test, como la presentación de ayuda y la corrección, entre ítems o al final del test; la temporización; posibilidad de añadir comentarios con las respuestas; presentación de los ítems; posibilidad de navegar entre los ítems realizados previamente a la finalización (en tests no adaptativos); posibilidad de reanudar un test tras algún cierre inesperado; y otras muchas más.

### 6.3.2. Arquitectura y funcionamiento

Un esquema de la arquitectura de Siette se puede ver en la figura 6.14, la cual es la versión actual de la arquitectura presentada en (Guzmán y Conejo, 2004a). Siette puede ser utilizado por los usuarios en la interfaz propia del sistema, o puede actuar como una componente más de soporte para un sistema externo. A continuación, se resume brevemente estos dos usos y la funcionalidad que puede realizarse con ellos.

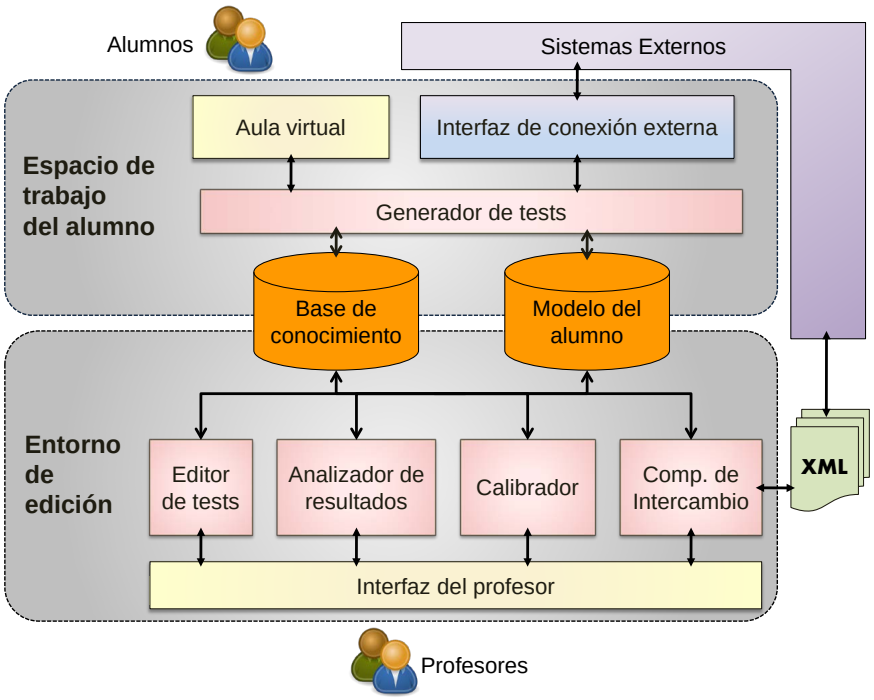



Figura 6.14: Esquema de la arquitectura de Siette.

En cuanto al uso normal de Siette, éste cuenta con diferentes roles de usuario, cada uno con una serie de permisos en el sistema. En el nivel superior de la jerarquía de usuarios está el administrador, que no está reflejado como tal en el sistema dado que es un profesor con permisos especiales. El administrador puede modificar la configuración general del sistema, gestionar los usuarios y actuar como profesor.

Un profesor en Siette podrá crear sus asignaturas y gestionarlas mediante el *editor de tests*. Dentro de la gestión básica de una asignatura, el profesor puede definir, eliminar y modificar los diversos contenidos de la asignatura que se almacenarán en la *base de conocimiento*. El objetivo del profesor consiste en la definición del conjunto de ítems que se usarán para la evaluación. Una vez están realizados y configurados los ítems, se definen los tests sobre los temas correspondientes, siendo una parte importante de esta etapa la configuración de las opciones de acceso, evaluación, temporización, etc.

Después de que algún test haya sido realizado por los alumnos, el profesor dispone de un conjunto de estadísticas y de informes sobre la actividad realizada en el *analizador de resultados*. Esta actividad está disponible tanto para TAI como para tests que siguen la TCT. Además, su presentación puede ser personalizada mediante diversos filtros, algunos de los cuales son: el informe por alumnos individuales, por test, por fecha, etc. Los datos de las sesiones realizadas pueden ser utilizados como fuente de información

en el *calibrador* para estimar las CCI de los ítems de un test.

Otra característica importante que puede realizar el profesor es utilizar la *componente de intercambio* para exportar e importar datos relacionados con los contenidos o con la evidencia recopilada de los alumnos mediante ficheros XML. En el propio sistema estos ficheros se pueden utilizar como mecanismo de copia de seguridad. También, pueden ser usados por sistemas externos para extender su banco de ítems o para utilizar nuevas evidencias sobre el conocimiento del alumno.

En cuanto al alumno, éste puede acceder al *aula virtual* en la que se le presentan los tests diseñados previamente por los profesores. Además de la realización de tests, de forma individual o colaborativa, los alumnos pueden consultar sus resultados en aquellos tests en los que el profesor ha dado permisos para ello. Para este fin, Siette está integrado con Ingrid, un sistema que utiliza las evidencias de un sistema concreto para presentarlas en un formato entendible por el alumno. En otras palabras, Siette, a través de Ingrid, abre su modelo del alumno con el fin de proporcionar una herramienta para que éste pueda gestionar su propio auto-aprendizaje.

En cuanto al uso de Siette por un sistema externo, éste puede servir tanto como para realizar tests mediante alguna interfaz Web que embeba el contenido de Siette, como para utilizar los mecanismos de evaluación de la TRI. Además, Siette puede vincularse de manera segura con Sistemas Gestores de Contenidos Educativos. Para ello proporciona un protocolo específico que hace uso de criptografía asimétrica para la autenticación de los sistemas externos. Un ejemplo de uso de este mecanismo es la incorporación de Siette en el campus virtual de la Universidad de Málaga, implementado con Moodle[1]. Esta integración permite a los alumnos de la universidad realizar toda la actividad disponible en Siette sin tener que abandonar el campus virtual.

### 6.3.3. Ítems compuestos

Como parte de esta tesis, se ha implementado el modelo teórico para tests explicado en la sección 5.1. Para ello se ha desarrollado en Siette los ya explicados *ítems compuestos*. Estos ítems, como su propio nombre indica, están formados a su vez por otros ítems. Además de para la aplicación del modelo de evaluación combinado de la TRI con el MBR, la utilización de este nuevo tipo de ítems sirve para definir ítems que puedan modelarse de acuerdo a esta estructura y que se usen íntegramente en sistemas de tests. En el primer caso se habla de ítems componente *virtuales*, ya que éstos modelan evidencia que no son preguntas; mientras que en el segundo caso se denominan *reales*, al ser ítems preguntados directamente.

A la hora de implementarse hay que tener en cuenta una pequeña consideración para poder aplicar los mecanismos de la TRI sobre los ítems componentes (equivalentes a las restricciones MBR). Puesto que una restricción pueden aparecer en diferentes problemas, usando la analogía presentada en la sección 5.1, las componentes podrán aparecer en diferentes ítems compuestos, siendo el mismo ítem. Tal y como está diseñado el sistema, la única forma de considerar dos ítems componente como si fuesen un mismo ítem es utilizar los mencionados ítems plantilla o generativos. Estos ítems puede aparecer repetidos en un test mediante diferentes instancias con valores diferentes y son tratadas como si fueran el mismo.

El desarrollo de este tipo de ítem ha supuesto modificar la parte del editor de tests para que el profesor pueda definirlos. Igualmente, para poder consultar los resultados,

[1] http://moodle.org/

el analizador ha debido ser extendido mediante los requisitos necesarios para considerar este nuevo tipo de ítems. A la vez, el calibrador debe tener en cuenta el modelo TRI para los ítems compuestos (las componentes se calibran como cualquier otro ítem).

El proceso de definición de ítems compuestos en Siette se realiza en el editor de tests y comienza por realizar las diversas componentes atómicas que posteriormente conformarán la respuesta compuesta. Posteriormente, en la interfaz se selecciona la creación de un ítem de tipo compuesto. La elección de este tipo de ítems permite seleccionar un título del ítem compuesto que será compartido por los ítems componente. Con esta información se creará el ítem compuesto vacío, siendo necesario seleccionar de entre los ítems disponibles cuáles serán sus componentes.

Como se mencionaba anteriormente, este tipo de ítems pueden ser utilizados no sólo para aplicar el modelo de evaluación propuesto en esta tesis, sino que también permiten la creación de ítems compuestos reales que sigan una estructura relacionada. Cuando los ítems se utilizan de esta forma, las componentes no tienen por qué ser ítems de verdadero / falso, como es requerido por las restricciones, sino que pueden emplearse cualquier otro tipo de ítem, ya sea de opción múltiple, de respuesta corta, etc. Un ejemplo de esto en el dominio de la Física se puede ver en la figura 6.15. Aquí, se muestra la parte del editor de tests en donde el profesor realiza una pre-visualización de cómo se presentaría el ítem a un alumno (la respuesta debajo de los ítems es mostrada sólo en este modo de visualización). El ítem trata sobre la velocidad de caída de un cuerpo en el vacío y cada componente es un ítem de respuesta corta donde el alumno debe introducir el valor de diversas variables involucradas en la situación que se presenta.

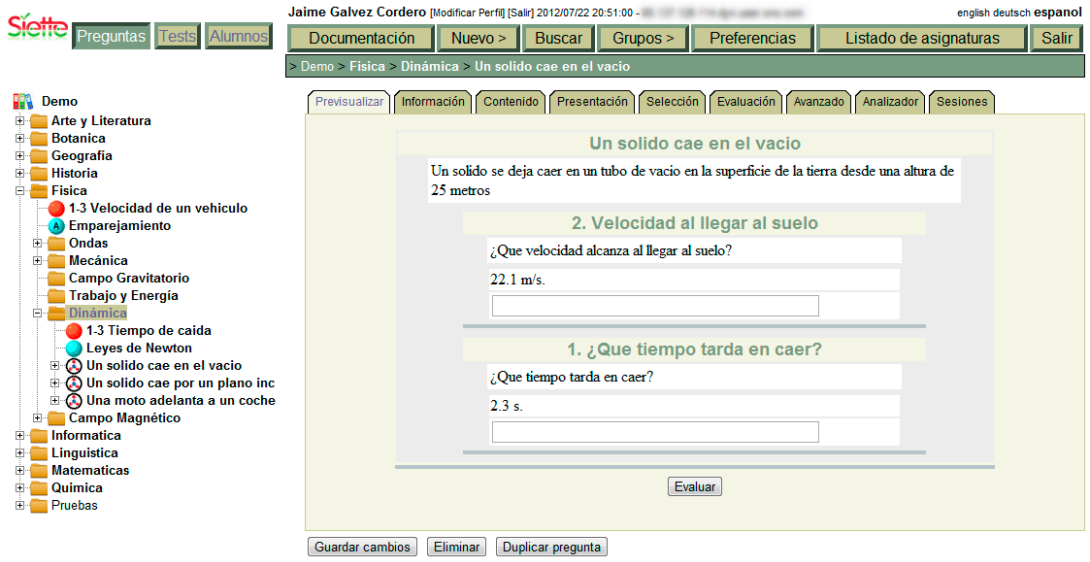



Figura 6.15: Ejemplo de ítem compuesto.

Dependiendo del uso que se le dé a los ítems compuestos, ya sea como parte de un test Siette, o como un problema en un sistema MBR, las evidencias del alumno son recopiladas y almacenadas en la base de datos interna. En el caso de sistemas MBR la evidencia es proporcionada utilizando la interfaz de comunicación externa mediante servicios Web. Esta evidencia es la fuente empleada por el calibrador para determinar las CCI que serán utilizadas posteriormente por los mecanismos de selección adaptativos

basados en la TRI.

## 6.4. SQL-Tutor y SQL-Tutor Processor

La herramienta SQL-Tutor ha sido desarrollada por investigadores de la Universidad de Canterbury (Nueva Zelanda) (Mitrovic y Ohlsson, 1999; Mitrovic, 2003a) y se ha utilizado para realizar algunos experimentos, los cuales se explican en detalle en el apartado 7.7. Concretamente, se utilizaron datos de la actividad de los estudiantes obtenidos durante diversos años. Estos datos fueron proporcionados por el grupo *Intelligent Computer Tutoring Group* (ICTG) de la mencionada Universidad y para su procesamiento se creó la herramienta *SQL-Tutor Processor*, todo ello en el marco de una colaboración iniciada tras la estancia del doctorando en dicha Universidad.

Puesto que SQL-Tutor fue explicado ampliamente en el apartado 2.3.7.2, aquí, sólo se explicará SQL-Tutor Processor, la cual ha sido implementada íntegramente por el doctorando dentro de la investigación asociada a esta tesis. SQL-Tutor Processor es una aplicación de escritorio desarrollada con Java cuyo objetivo consiste en procesar datos del sistema SQL-Tutor. Ésta permite usar los datos recopilados durante un periodo de interacción con el alumno, para transformarlos, poniéndolos en un formato adecuado que permita aplicar los modelos de evaluación teóricos explicados en capítulos anteriores. Aunque ha sido desarrollada específicamente para tratar datos de SQL-Tutor, ésta puede aplicarse para procesar datos de cualquier otro sistema MBR que almacene los datos en el mismo formato que el sistema anterior o que, con un formato diferente, cambie una de las componentes del sistema. La principal utilidad de esta herramienta se encuentra en la capacidad que ofrece para aplicar la evaluación a posteriori, la cual puede ser personalizada en numerosas opciones, como se podrá ver a continuación.

En particular, la herramienta es capaz de procesar ficheros que reflejan la actividad realizada por los estudiantes y que son almacenados por SQL-Tutor tras la utilización del sistema. Estos ficheros pueden ser de dos tipos: modelo del usuario y registro de actividad o de log. Los primeros tipos de ficheros almacenan para cada usuario el histórico de restricciones que se han violado o satisfecho, en qué problemas fueron relevantes, y una estimación del nivel del estudiante. El segundo tipo de ficheros guardan toda la actividad que un usuario tiene en el sistema, desde que inicia sesión, hasta que la cierra. Esto incluye toda la lógica de secuenciación que el sistema propone al estudiante (siguiente problema sugerido, nivel estimado del alumno tras una acción que cambie este valor, y otras medidas internas). El fichero también incluye la información de cada intento que el alumno realiza en el sistema, la cual engloba la solución enviada, y el resultado de la misma. Este último compuesto por las restricciones violadas y satisfechas, y el refuerzo recibido.

Los ficheros del modelo del estudiante y los de log han cambiado su formato durante la investigación realizada en esta tesis con los datos de SQL-Tutor. De esta forma, los datos generados hasta el año 2008 siguen un formato concreto, pero a partir del 2009, éstos cambiaron añadiendo nueva información y reestructurando la ya existente. La herramienta tiene capacidad para procesar los dos formatos, siempre y cuando se le indique explícitamente cuál es el que se debe usar (no detecta automáticamente cuál es el formato del fichero). Ya sea del modelo de estudiante o del fichero de log, la información procesada se debe transformar para aplicar las técnicas de evaluación de la TRI, las cuales utilizan otra herramienta independiente: *MULTILOG* (Thissen et

al., 2003). MULTILOG se encarga de ejecutar mecanismos de calibración de la TRI a partir de ficheros de datos, que contienen información de las evidencias del alumno, y de unos ficheros de instrucciones que contienen las órdenes y comandos necesarios para indicar qué mecanismos y de qué forma aplicarlos.

Así pues, la parte central de la herramienta radica en la generación de los ficheros de órdenes para MULTILOG y de los datos con las evidencias del estudiante, los cuales requieren de un formateo adecuado y de un procesamiento de los datos provenientes de las herramientas MBR. SQL-Tutor Processor tiene dos modos de funcionamiento, en primer lugar, la herramienta dispone de una interfaz gráfica que permite de una forma sencilla realizar el procesado. Esta forma de ejecución permite procesar un único conjunto de datos con todos los ficheros de cada alumno situados en un directorio. El conjunto de datos debe seguir un único formato de los dos mencionados anteriormente. El segundo modo de funcionamiento carece de interfaz y consiste en la ejecución de un script, el cual puede combinar diferentes formatos almacenados en conjuntos de datos diferentes, aplicando modelos de evaluación diferentes y con diversas combinaciones. El problema de este modo radica en que la personalización del script requiere de conocimientos de programación en el lenguaje Java.

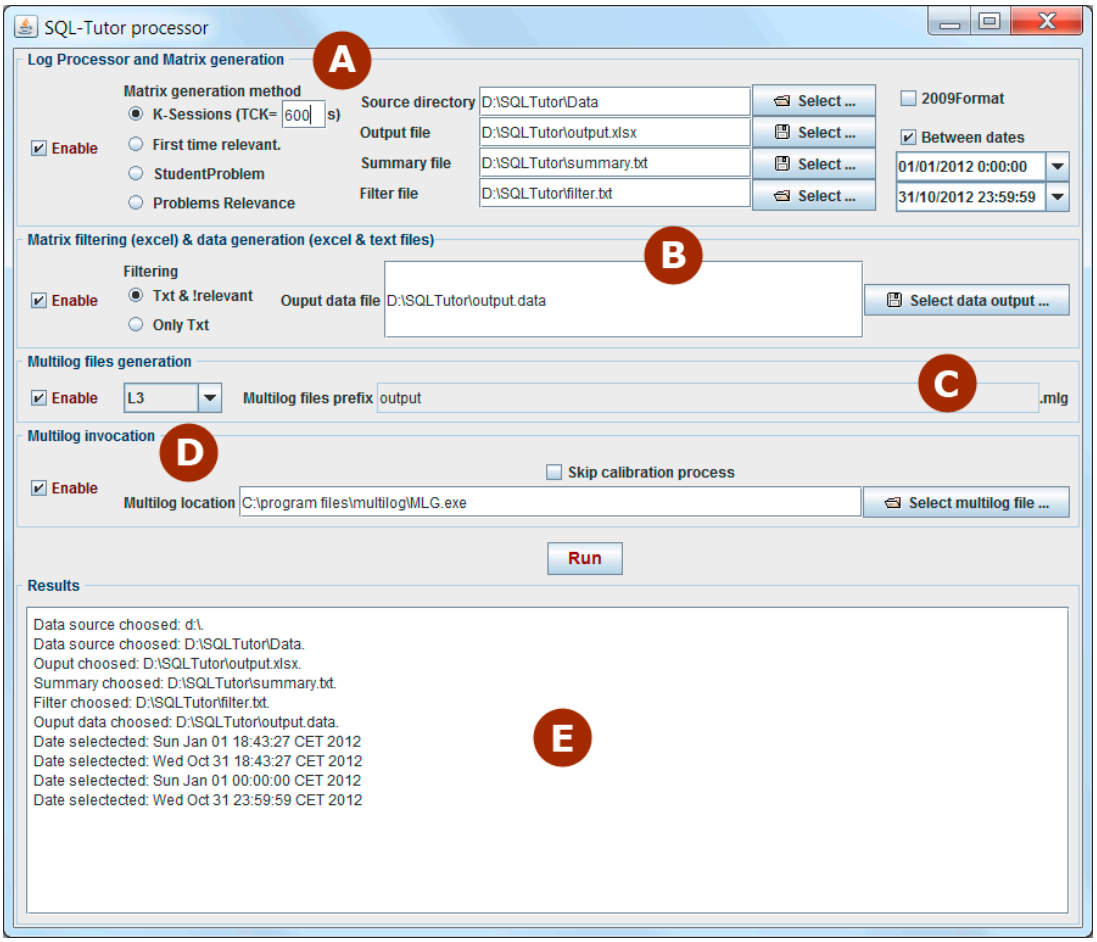



Figura 6.16: Interfaz de la herramienta SQL-Tutor Processor.

En el primer modo de ejecución se puede realizar el procesado completo de manera

automática, el cual abarca desde la calibración de las CCR hasta la evaluación usando estas mismas. También, es posible ejecutar el proceso por partes, seleccionando qué pasos se desean ejecutar. En cualquier caso, el usuario debe establecer una serie de parámetros en la interfaz gráfica, la cual puede observarse en la figura 6.16. La interfaz está compuesta de varios elementos, cada uno asociado a un paso del proceso completo y que se enumeran a continuación según la etiqueta que lo identifica en la interfaz:

(A). El primer paso se encarga de recopilar las evidencias de un conjunto de alumnos y agruparlas según varios de los métodos mencionados en la sección 5.2.2.3, generando la correspondiente matriz de rendimiento. Los diferentes métodos de generación de la matriz implementados son:

- Agrupación mediante umbral TCK fijo. Este método requiere de especificar el valor (en segundos) que se quiere usar para realizar la agrupación de evidencias.
- Agrupación de evidencias considerando sólo el valor de la primera vez que la restricción es relevante a lo largo de toda la actividad del alumno.
- Agrupación mediante problemas realizados por el estudiante, el cual es similar a la primera vez que una restricción es relevante, pero restringiendo esto a la aparición dentro de los diferentes intentos realizados para un problema.
- Adicionalmente, la herramienta tiene un método para procesar todas las evidencias y generar un resumen sobre las restricciones que son relevantes en cada problema. Si bien no es un método de recopilación de evidencias, es una funcionalidad que puede ser útil para realizar estudios en los que es necesario conocer la relevancia de cada restricción respecto de los problemas y no se dispone de esta relación de antemano o no es trivial calcularse.

La ejecución de este paso estará condicionada por el método de agrupación de evidencias seleccionado. Además, será necesario proporcionar los siguientes parámetros:

- Directorio fuente, en el que se encuentran los ficheros del registro de actividad realizada por cada usuario.
- Formato de los ficheros, que podrá tomar los dos valores mencionados previamente: ficheros con formato anterior al 2009, o con formato igual o superior a este año.
- Fichero de salida de resultados, en el que se escribirá la matriz de rendimiento. Este podrá ser un fichero de texto plano (*.txt*), o un fichero de hoja de cálculo según el formato de Microsoft Excel (*.xls* o *.xlsx*).
- Fichero resumen de resultados, que contendrá un resumen de la actividad realizada por cada estudiante virtual (ver sección 5.2.2 para más detalle). La información contiene la fecha / hora de la primera acción realizada, fecha / hora de la última acción realizada, número de intentos realizados y número de problemas intentados. Esta información es útil y más fácil de procesar que la matriz de rendimiento si se desean obtener estadísticas generales de cada alumno.
- Fichero filtro de restricciones, el cual contiene una lista de identificadores asociados a restricciones que serán filtradas durante la recolección de evidencias. Con esta opción se pueden descartar del estudio restricciones que

previamente se conocen son incorrectas, no proporcionan información útil, o se desean obviar por alguna otra razón.

Otra opción disponible es la de recopilar evidencias sólo durante un periodo de tiempo determinado. Para ello, se puede seleccionar la fecha y hora inicial y final de este periodo. Sólo las evidencias que han tenido lugar dentro del intervalo serán tenidas en cuenta para formar la matriz.

(B). El siguiente paso consiste en, a partir de la matriz de rendimiento del paso anterior, poner las evidencias en un formato adecuado para que puedan ser utilizados en los mecanismos de la TRI que proporciona la herramienta MULTILOG. Aquí, sólo es necesario especificar la localización del fichero que se va a generar. Dado que la matriz generada en la etapa anterior puede contener algunas restricciones que no sean relevantes, se proporciona una opción para filtrarlas y que no sean tenidas en cuenta en la generación los ficheros de datos.

(C). Como tercer paso, se generan los ficheros de instrucciones que determinan los mecanismos de la TRI que MULTILOG aplicará. En este sentido, se generan dos tipos de ficheros, los que establecen los parámetros necesarios para realizar la calibración de las restricciones, y los que se encargan de definir cómo realizar la evaluación. En estos ficheros se indica a MULTILOG la procedencia de los datos, número de alumnos, y el modelo a aplicar para las estimaciones. Este último puede ser seleccionado por el usuario de entre varios disponibles: 1PL, 2PL, o 3PL (funciones logísticas de 1, 2, y 3 parámetros); modelo nominal; de respuesta graduada; y de respuesta para ítems de opción múltiple (Thissen et al., 2003).

(D). El paso final del proceso realiza la invocación a MULTILOG a partir de todos los ficheros existentes y generados en las fases anteriores. El parámetro que hay que indicar en la interfaz es la ruta donde se encuentra el ejecutable de MULTILOG. Por defecto, se ejecutará el proceso de calibración y a continuación el de evaluación. No obstante, se puede seleccionar que se ejecute solamente el de evaluación a partir de una calibración existente previamente. De esta forma, se pueden calibrar las restricciones, que es el proceso más costoso computacionalmente, y ejecutar diversas evaluaciones a posteriori, sin necesidad de realizar el proceso de calibración cada vez.

(E). La última componente de la interfaz no se corresponde a ningún paso, sino que se encarga de mostrar mensajes sobre las acciones realizadas en la interfaz. Estos mensajes son informativos, advirtiendo del cambio de los parámetros de la interfaz o mostrando el resultado de cada paso. De esta forma, se va informando al usuario si cada paso se ha ejecutado correctamente o si ha habido algún error, y su descripción. La salida que MULTILOG genera, tras ser invocado, también es recogida y mostrada en este panel informativo.

La segunda forma de ejecución combina el proceso descrito anteriormente con diferentes conjuntos de datos sobre los que recopilar evidencias; diferentes métodos de formar la matriz de rendimiento; diferentes modelos de la TRI; diferentes filtros de restricciones asociados a diferentes experimentos; e incluso, métodos que utilizan la calibración de las restricciones para generar la función de información de cada restricción de acuerdo a los diferentes modelos. La aplicación está diseñada de manera modular

en diversas componentes, de tal forma que cada uno de los métodos descritos es reutilizable para combinarse de la manera deseada. Lógicamente, para poder realizar esta combinación, es necesario programar el script correspondiente en el lenguaje Java.

## 6.5. Marco de trabajo CBMEngine

Aunque, tal y como se explicaba en la sección anterior, la parte del modelo de evaluación genérico explicado en el capítulo 5 se implementó en el sistema SIETTE. Todavía, la parte correspondiente al MBR quedaba en los EIRP. Por este motivo, y partiendo de la experiencia adquirida durante el desarrollo de los tutores mencionados en las secciones anteriores de este mismo capítulo, se abstrajeron elementos comunes a sus arquitecturas y se pusieron en un marco de trabajo llamado *CBMEngine* (Gálvez et al., 2012). El nombre de esta plataforma viene del Inglés *Constraint-Based Modeling Engine*, que en español podría traducirse como Motor MBR.

El objetivo de CBMEngine es facilitar la tarea de construcción de EIRP que quieran hacer uso del modelo de evaluación MBR + TRI, sin distinción del dominio de aplicación. Para usar este modelo, no es necesario implementar cada componente del MBR en el sistema, sino que sólo es necesario implementar los métodos que permitan enviar la información recopilada a través de la interacción con el alumno al CBMEngine. Consiguientemente, CBMEngine es un componente reutilizable para la construcción de nuevos sistemas. Con el objetivo de garantizar la robustez del sistema, su desarrollo ha sido guiado por pruebas (en inglés *Test-driven development*). Aunque este tipo de metodología requiere mayor tiempo de desarrollo, aumenta la robustez, característica fundamental en una componente que va a dar servicio a otros sistemas.

La plataforma se basa en el mismo esquema que el utilizado en Simplex Tutor y que se abstrajo posteriormente en el marco de trabajo presentado en (Gálvez et al., 2008). Por tanto, CBMEngine se puede considerar como la evolución de estos trabajos y la primera versión funcional que posibilita la utilización del modelo combinado MBR + TRI. A diferencia de las ideas iniciales, CBMEngine carece de una interfaz propia, pues se encarga solamente de implementar la parte correspondiente al modelo del alumno y del dominio; a realizar las inferencias de evaluación; y a aplicar las estrategias adaptativas. Con este enfoque, CBMEngine consigue ser totalmente independiente del dominio donde se desee aplicar.

Al igual que sus predecesores, CBMEngine es una plataforma SOA desarrollada en Java que utiliza un motor de inferencia para evaluar las evidencias recogidas del alumno. La arquitectura del sistema, mostrada en la figura 6.17, comprende tres capas principales: a) una capa situada en el nivel más alto ofrece servicios Web a sistemas externos; b) una capa de lógica de negocio que comprende los elementos asociados con la funcionalidad del sistema; y c) una capa de persistencia que gestiona el almacenamiento de estructuras de datos asociadas a los modelos del alumno y del dominio. Aunque el sistema dispone de una interfaz Web, ésta sólo permite llevar a cabo las tareas básicas de administración y su carácter es principalmente informativo. Por este motivo, esta interfaz no se contempla como parte de la arquitectura principal. Más detalles sobre esta interfaz se pueden encontrar en el apéndice A.

En la capa superior se proporcionan los servicios Web que cualquier EIRP que quiera utilizar nuestro modelo debe invocar. Para realizar correctamente la tarea de un servicio, los sistemas externos deben proporcionar información, de alguna manera a

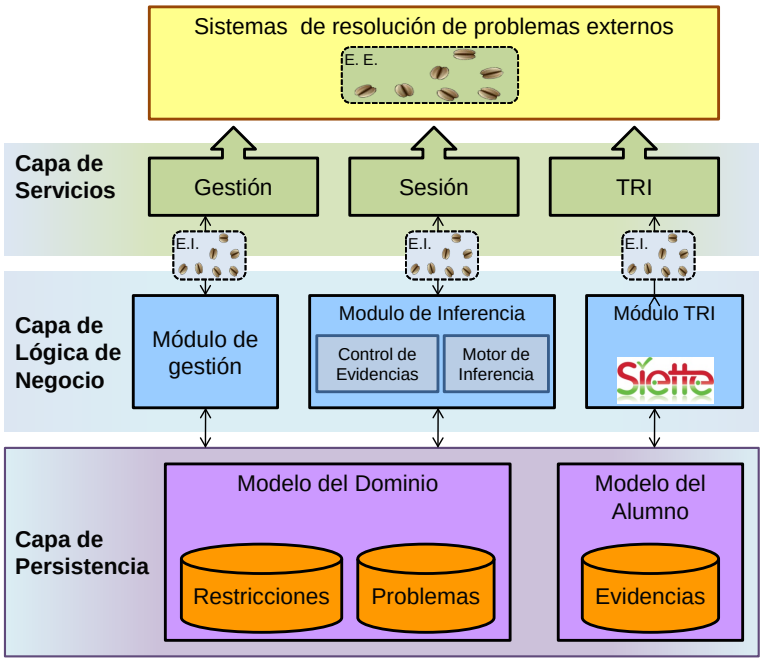



Figura 6.17: Arquitectura de la plataforma CBMEngine.

CBMEngine. Para esta tarea se deben utilizar lo que se ha denominado como **estructuras externas**, reflejando el origen de la información que contienen (en la figura, están señaladas con la etiqueta "E.E." y con un contorno discontinuo). Estas estructuras normalmente son una serie de Java Beans, como se hacía en Simplex Tutor, cuyo único objetivo es recoger sólo la información relevante. También, pueden ser XML con una estructura concreta que almacene la misma información. En cualquier caso, los sistemas externos pueden usar tres puntos de entrada SOA, a partir de los cuales se pueden acceder a los diferentes servicios. Estos tres puntos de entrada categorizan los servicios dependiendo de la funcionalidad a utilizar de CBMEngine:

- En primer lugar, el punto de entrada *de Gestión*, como su propio nombre indica, permite gestionar los usuarios y los problemas. A través de este punto de entrada se pueden usar diversos servicios para las tareas básicas de registro, modificación y borrado de los usuarios y los problemas asociados a un sistema externo. Es requisito fundamental para poder usar los servicios posteriores de evaluación, que el sistema externo registre en CBMEngine los usuarios y que registre los problemas que éstos pueden realizar.
- El siguiente punto de entrada se denomina *de Sesión*. Los servicios agrupados aquí tienen como finalidad la recopilación de evidencias de los sistemas externos durante una sesión de uso. A través de estos servicios se puede iniciar una sesión de resolución; seleccionar el problema que se está resolviendo o sobre el cual se van a proporcionar evidencias; añadir información correspondiente a la solución dada por un alumno en un intento; comprobar los errores existentes en la solución de acuerdo al proceso de inferencia del MBR y ordenarlos adaptativamente, de acuerdo al nivel estimado del alumno; informar sobre la finalización de un problema; o cerrar la sesión.

- El último punto de entrada se le ha dado el nombre de *TRI*. En este otro punto, los servicios que se pueden invocar permiten la aplicación de los mecanismos de la TRI desarrollados en esta tesis. Estos servicios incluyen la calibración de las restricciones, las estrategias básicas de selección de problemas, y la invocación para calcular el nivel estimado del alumno en base a las evidencias que tiene la plataforma.

Cada dominio tendrá sus tres puntos de entrada, los cuales se particularizan para las estructuras externas concretas requeridas por un sistema externo. Los puntos de entrada usan funciones de tres componentes internos comunes que, asociados a la funcionalidad descrita anteriormente, se encargan de pasar los datos a la capa siguiente.

En la capa intermedia se llevan a cabo las acciones requeridas por los servicios anteriores, agrupándose en módulos, de acuerdo al servicio que dan soporte. En cada uno de ellos se trata la información de las estructuras externas que llegan a través de los servicios y se transforma en estructuras internas, que extenderán la información básica externa con información propia de la plataforma.

- En el módulo de gestión, el control de la información que se realiza consiste básicamente en añadir, actualizar o borrar la información asociada a los usuarios / problemas correspondientes de la capa de persistencia.

- En el módulo de inferencia, se controlan las sesiones existentes, procesando las evidencias que se van recibiendo para cada intento de un alumno. Tales evidencias, además de pasarse a la capa de persistencia para registrar toda la actividad del usuario, se introducen en el motor de inferencia, para lo cual se añaden una serie de identificadores y manejadores requeridos por el mismo a las estructuras internas. El resultado de la comprobación de errores también corresponde a este módulo, el cual, actualizará el modelo del alumno de la capa de persistencia con las nuevas evidencias de violación de restricciones o de satisfacción. Además, estos errores estarán disponibles para los sistemas externos en caso que quieran utilizar el refuerzo que proporciona el MBR. En este caso, se ordenan por el criterio que se haya seleccionado al iniciar la sesión y ya el sistema externo lo usa como crea conveniente.

- El último módulo se encarga de dar servicio a las peticiones relacionadas con los mecanismos de la TRI, tales como la calibración, según los métodos presentados; las estrategias básicas de selección; y la determinación del conocimiento del alumno. Estas peticiones se pueden atender invocando los servicios desarrollados en la implementación del modelo realizada en SIETTE o bien usando la herramienta SQL-Tutor Processor (ver la sección 6.4 para ampliar detalles de las características de ésta).

.

Al igual que en Simplex Tutor, se sigue usando el motor de inferencia JBoss Rules. Sin embargo, merece la pena destacar la ventaja de usar este en un entorno como el del CBMEngine en el que pueden coexistir múltiples dominios funcionando. Aparte de las ventajas mencionadas en el apartado 6.2.1, su uso supone una mejora sobre los enfoques existentes para sistemas educativos bajo el MBR y que son multidominio. Para estos entornos, Mitrovic (1998a); Mitrovic et al. (2005) utilizan las llamadas *redes de restricciones* con el fin de mejorar la eficiencia. Esto no es más que agrupar elementos

comunes de las reglas con la carga del sistema para formar redes RETE que optimicen el posterior proceso de emparejado de patrones. Este mecanismo no es necesario realizarse con JBoss Rules ya que el motor automáticamente compila las reglas de entrada, una vez que son cargadas en el motor, como redes RETEOO (redes RETE extendidas y optimizadas para Orientación a Objetos).

La última capa es la de persistencia, la cual contiene el modelo de dominio y del alumno siguiendo con la formulación del MBR. Estos modelos son almacenados en una base de datos y actualizados por los módulos de la capa intermedia con la información de los sistemas externos. Respecto del modelo del alumno, al igual que en los sistemas anteriores, se siguen guardando tanto las evidencias de cada intento que es realizado, como el nivel estimado, obtenido a través del módulo de evaluación. Por otro lado, el modelo del dominio está comprendido por los datos que representan los problemas que se podrán realizar en los sistemas externos y por las restricciones del dominio.

### 6.5.1. Implementación de restricciones mejorada

Las restricciones en este marco de trabajo han evolucionado respecto de las anteriormente usadas en los sistemas OOPS o Simplex Tutor. Principalmente porque la condición de satisfacción asociada a la detección del estado solución incorrecto se ha movido al consecuente de la regla. Si bien esta condición es por naturaleza propia del antecedente, el cambio realizado no afecta al comportamiento de inferencia del MBR y tiene una razón de ser muy importante. Antes, con la estructura de las reglas que se utilizaban, era necesario fijar manualmente cuáles de las restricciones eran relevantes para un problema dado o duplicar la regla para registrar la relevancia, ya que no se podía realizar de manera automática durante el proceso de inferencia. El motivo de esta incapacidad radica en dos elementos:

- La detección automática de la relevancia de una regla como parte del proceso de inferencia requiere de registrar este hecho en la capa de persistencia. Esto se puede hacer en dos lugares: o bien en el consecuente de la regla, como un método más que realice esta acción, o como parte de la condición de satisfacción. Esta segunda opción no es coherente por dos motivos: el primero, que teóricamente, la condición de satisfacción sólo debería comprobar si el estado es incorrecto, sin realizar otras acciones que no tienen que ver con esta comprobación; y segundo, que en la práctica esto no es posible ya que, por el lenguaje utilizado para codificar las reglas, no en todas las restricciones se realizan llamadas a métodos para comprobar la condición de satisfacción. En restricciones simples, el propio lenguaje ya incluye mecanismos que hacen innecesario la llamada a métodos Java y, de reescribir estas restricciones para que usaran llamadas a métodos, se estaría añadiendo complejidad extra sin necesidad. Por ello, la única parte válida para registrar la relevancia de las restricciones sería el consecuente de las reglas.

- Puesto que la condición de satisfacción estaba unida a la de relevancia en el antecedente de una regla, para poder ejecutarse el consecuente, las condiciones de relevancia y de satisfacción tenían que ser ciertas a la vez, lo cual equivale a violar la restricción. Esto implica que el registro de la relevancia sólo se pueda hacer automáticamente para las restricciones violadas, dejando fuera aquellas que eran satisfechas por el alumno. Una opción que permite registrar la relevancia de restricciones satisfechas es el uso de reglas que modelan la satisfacción pero

requiere duplicar cada regla, añadiendo mucha más complejidad al modelo del dominio.

La solución tomada en CBMEngine supone situar la condición de satisfacción en el consecuente de la regla y justo después de haber realizado el registro de la relevancia. Esto se puede ver más claro en la regla mostrada como ejemplo en la figura 6.18. Aquí podemos ver que el antecedente de la regla está compuesto por la condición de relevancia únicamente. De esta forma, si la restricción es relevante, siempre se ejecuta el consecuente, el cual, primero ejecutará el registro de la restricción, etiquetado con "Tratamiento $Cr$"; a continuación comprobará la condición de satisfacción; y, sólo si la solución representa a un error y la condición de satisfacción es cierta, se realiza el tratamiento de la violación, que corresponde con lo que en los sistemas OOPS y Simplex Tutor era el consecuente (etiquetado con "Consec." en la figura). La regla corresponde al tutor Visual Nets que se explica en el apartado 6.5.4.2.

```
rule "Net Subnet NumIps inf"
    when
        p: DefNetProblemSubntBean(s1 : session, prob: id)
        s: NetInternalSolutionSubntBean(session == s1, problem == prob)        Cr
    then
        String nombreRegla = drools.getRule().getName();
        relevantes.newRelevance(nombreRegla);          Tratamiento Cr
        if (s.checkNumIpsInference()){                                          Cs
            String m = "Número de Ips es incorrecto para los nodos resaltados";
            logger.logInferenceInfo(m);                                  Consec.
            errors.newError(nombreRegla, m, s.getNumIpsInferenceErrorsIds());
        }
end
```

Figura 6.18: Restricción implementada en el motor CBMEngine.

Si bien el hecho de contemplar la relevancia de las restricciones es de considerable magnitud en sistemas MBR, ya sea para elaborar planes de instrucción, para calcular heurísticos que determinan el nivel del estudiante, o para discriminar las restricciones que son apropiadas para evaluación; en la literatura existente no se ha encontrado ningún ejemplo del tratamiento a llevar a cabo para determinar automáticamente la relevancia. Por el contrario, en los ejemplos de reglas encontrados, algunos de los cuales son (Mitrovic, 1998a; Martin y Mitrovic, 2000; Baghaei et al., 2006; Mitrovic y Ohlsson, 2006), sólo se muestran las condiciones de relevancia y satisfacción, sin indicar cómo se tratan, lo que nos lleva a pensar que si hay algún tratamiento de la relevancia, éste se realiza fuera de la regla. Esto se puede justificar por el uso de lenguajes que no permiten de manera directa el tratamiento y, dado que es algo que se debe hacer para todas las reglas, probablemente convenga hacerlo en otro nivel, invocando a las correspondientes funciones del lenguaje. Dado que en DROOLS el procesamiento, consistente en invocar un simple método de Java se puede conseguir de manera sencilla, se ha incorporado como parte del mecanismo natural de las reglas.

### 6.5.2. Evaluación en CBMEngine

La parte de CBMEngine que se encarga de aplicar los mecanismos de evaluación, así como también, de la calibración de las CCR, tiene dos componentes diferentes que llevan a cabo estas tareas. Dado que los mecanismos de la TRI ya están implementados

por otras herramientas, CBMEngine no los re-implementa, sino que hace uso de la funcionalidad proporcionada por estas herramientas en la medida de lo posible.

Por un lado, CBMEngine puede aplicar los mecanismos de la TRI para la calibración y la evaluación utilizando la herramienta MULTILOG, al igual que se hizo previamente en SQL-Tutor Processor. De hecho, CBMEngine invoca a MULTILOG a través de los componentes reutilizables de SQL-Tutor Processor. MULTILOG permite utilizar los modelos paramétricos más importantes explicados en la sección 3.2.1 como: las funciones logísticas de 1, 2 y 3 parámetros; el modelo de respuesta graduada; o el modelo de crédito parcial, entre varios otros. Para la calibración se utilizan los mecanismos típicos mencionados en la sección 3.3.1.2 como la máxima verosimilitud conjunta o la marginal, en conjunción con los métodos de agrupación en CK-sesiones mencionados en la sección 5.2.2.1. Para la evaluación se utiliza la máxima verosimilitud conjunta o el método bayesiano del Máximo a Posteriori. Las evidencias recopiladas en CBMEngine son puestas en un formato entendible por MULTILOG y el resultado de aplicar alguno de los mencionados métodos de calibración o evaluación es capturado por el sistema.

La otra opción es utilizar los servicios de Siette. Para ello es necesario definir un ítem simple por cada restricción del sistema y tantos ítems compuestos subyacentes como problemas tenga el sistema, de acuerdo con el modelo presentado en la sección 5.1. Hasta la fecha no existe un mecanismo automático para ello, por lo que esta labor debe realizarse a mano. Una vez definido el ítem, CBMEngine proporciona las evidencias necesarias a Siette, delegando la tarea de calibración o evaluación en esta plataforma. Para estas tareas, los métodos de calibración y de evaluación de los que Siette dispone utilizan un modelo no paramétrico, presentado por Guzmán (2005).

### 6.5.3. Uso de CBMEngine en un sistema externo

Se ha explicado cómo funciona y cuáles son las características de CBMEngine, pero ¿Cuáles son los pasos a seguir para poder utilizar CBMEngine como parte de un EIRP ya existente? Puesto que se trata de integrar dos sistemas, el uso de un lenguaje común de comunicación para poder entenderse es un requisito fundamental. Por ello, dado que los mecanismos usados para comunicarse son las estructuras externas, y dado que cada sistema puede dialogar un lenguaje propio y asociado a un dominio particular, en algún momento determinado se debe especificar en CBMEngine qué información contienen las estructuras externas para que, además de comunicación, haya un entendimiento común. Este es el primero de los pasos requeridos en el proceso de integración. El segundo, es un paso que inevitablemente siempre está presente en cualquier sistema educativo que quiera utilizar el MBR. Éste consiste en la elaboración de las reglas del dominio mediante la codificación de los principios / restricciones que permitirán detectar los errores del alumno. Por último, sólo se tendrán que invocar a los servicios necesarios pasando la información sobre las estructuras correspondientes creadas anteriormente.

En cuanto al primer paso, como se ha comentado anteriormente, las estructuras externas en CBMEngine son Java Beans, cuya única función es almacenar de la forma más simple posible la información a intercambiar. Pero ¿cómo se establece qué información deben contener? La respuesta a esta pregunta nos lleva a uno de los puntos débiles de CBMEngine en la actualidad. Estas estructuras están intrínsecamente ligadas a la información de un dominio particular, por lo que para cada dominio se deben definir nuevas estructuras. La especificación del contenido de estas estructuras para los dos sistemas que actualmente utilizan CBMEngine, y que se explicarán a continuación, se

ha realizado programando directamente las estructuras requeridas. Sin duda, ésta no es la mejor forma de hacerlo, pues requiere de conocimientos de programación y de cierto tiempo de dedicación. Para paliar este inconveniente y, a la vez, realizar el segundo paso, se ha desarrollado la herramienta CBM-DoME (sección 6.6), la cual permite la definición de las estructuras y reglas asociadas a un dominio de una forma sencilla y sin muchos conocimientos técnicos.

Aunque la herramienta CBM-DoME facilita la tarea de crear las estructuras y las reglas, una posibilidad que probablemente permitiría mejorar el sistema sería el descubrimiento semiautomático de las estructuras por parte de los sistemas externos usando una ontología de descubrimiento. De esta forma el sistema externo especificaría las estructuras y CBMEngine las generaría automáticamente, ahorrándonos el tener que crear las estructuras desde cero. No obstante, este mecanismo es sólo una idea que no se ha explorado durante la elaboración de esta tesis dado que ha surgido al final de la misma. Habría que evaluar si el proceso de descubrimiento haría más complejo el uso de CBMEngine desde un sistema externo en comparación con la creación a partir de CBM-DoME.

Una característica que puede resultar atractiva para los sistemas externos que utilicen CBMEngine, es la posibilidad de usar identificadores de interfaz en las estructuras externas. Esto consiste en usar un campo de una estructura externa con el identificador del elemento que, en la interfaz, se asocia a la información almacenada en esa estructura. Este campo se utiliza en el proceso de inferencia, en el momento en se viola una restricción debido a los valores de la estructura, para añadirse como parte del error cometido. De esta forma, entre la información de los errores que CBMEngine proporciona, se encuentran estos identificadores de la interfaz, los cuales pueden utilizarse para remarcar el elemento y así ayudar al alumno a identificar el error correspondiente.

Respecto al tercer paso de integración, se ha definido un protocolo que los sistemas externos deberían seguir para hacer uso adecuado de los servicios Web. Antes de poder usar el diagnóstico del MBR con la TRI proporcionado por la plataforma es fundamental que los sistemas externos registren en CBMEngine los estudiantes del mismo y los problemas del dominio. También, deberían seleccionar un modo de funcionamiento, de entre los tres posibles mencionados en el apartado 5.3.5; y elegir un método de agrupación de las restricciones. Posteriormente, deben iniciar una sesión de trabajo para el alumno y proporcionar la información sobre su actividad. CBMEngine informará al sistema externo de los errores cometidos y actualizará el modelo del alumno. Los mecanismos relacionados con la TRI, como la calibración, evaluación o selección del problema más adecuado, se pueden usar durante este proceso o de manera independiente. Este protocolo, junto con los servicios Web involucrados, y sus características, son detallados en el apéndice A.

### 6.5.4. Plataforma DEDALO

Un ejemplo palpable de la utilización de CBMEngine en sistemas externos se encuentra en la plataforma que surge del proyecto DEDALO (DEDALO, 2009) y que tiene este mismo nombre. El objetivo de este proyecto es la creación de un marco de trabajo para el modelado y el diagnóstico del estudiante en entornos procedimentales que también pueda servir para mejorar el proceso instructivo del alumno. Además, con este marco de trabajo se pretende favorecer la construcción de EIRP educativos de forma genérica mediante la independización de componentes y su reutilización.

El marco de trabajo busca ofrecer un lugar virtual donde diferentes herramientas puedan ofrecer y consumir servicios proporcionados por otras. Para ello, DEDALO basa su funcionalidad en servicios Web y se organiza en una arquitectura como la mostrada en la figura 6.19. De las seis componentes básicas la principal es la etiquetada como ESB (*Enterpise Service Bus*). ESB es un bus mediador de servicios Web que da acceso a la capa de servicios, encamina mensajes, balancea la carga, controla la escalabilidad, etc. La componente SCA (*Service Component Arquitecture*) es una capa sobre la que recaen los servicios Web de la plataforma. La capa etiquetada *Agentes* contiene servicios Web nativos proporcionados por la plataforma para modelar y evaluar al alumno. La componente *Corpus* contiene un conjunto de ontologías con una representación genérica de los diferentes conceptos y habilidades involucradas en un dominio de enseñanza particular. La consola es una interfaz para la gestión de la plataforma, sus servicios, y para realizar las tareas administrativas. El *Repositorio* contiene la base de datos asociada a la plataforma.

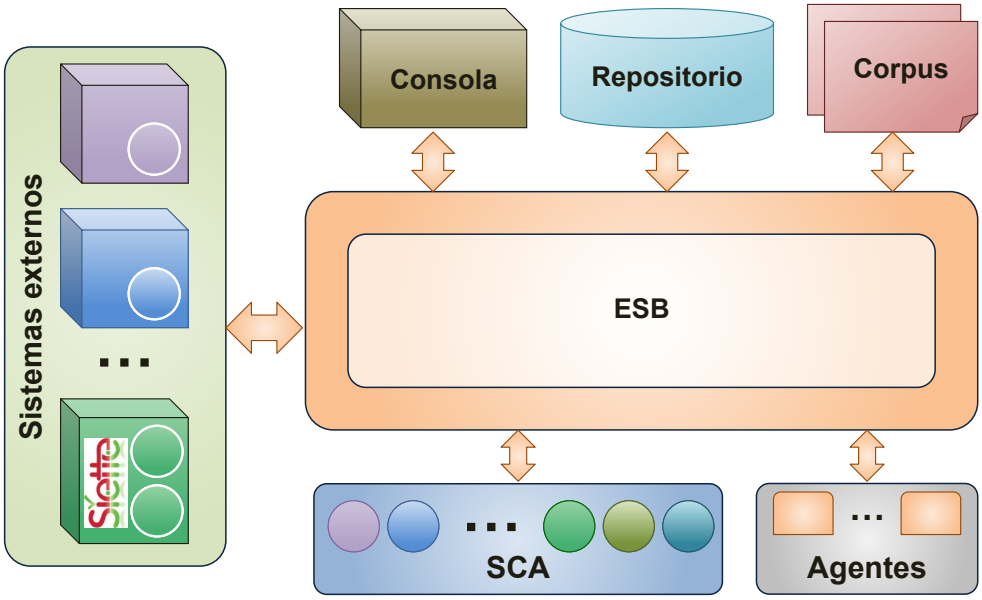



Figura 6.19: Arquitectura del marco de trabajo DEDALO.

En esta arquitectura, una de las componentes utilizadas para recopilar evidencias del alumno y realizar la evaluación del mismo es CBMEngine. Los puntos de acceso Web de CBMEngine son utilizados por la plataforma para proporcionar servicios de evaluación a los sistemas educativos que se asienten sobre DEDALO. Mediante la integración realizada se ha podido ajustar las características necesarias de CBMEngine para dar soporte a múltiples dominios en paralelo. Concretamente, se han implementado dos sistemas que hacen uso de estos servicios en la plataforma y que se explican a continuación.

#### 6.5.4.1. PIPSE

*PIPSE* es el primer sistema de DEDALO en el que se integró CBMEngine como parte del funcionamiento del mismo (Gálvez et al., 2012). Su nombre viene por las siglas en inglés de *Project Investment Problem Solving Environment*, que en español podría

traducirse como Entorno de Resolución de Problemas de Análisis de Inversiones. Tal y como su nombre indica, el dominio de aplicación se centra en la Gestión de proyectos y, dentro de ésta, en el ámbito del Análisis de Inversiones. El objetivo de los problemas que se plantean al alumno consiste en estudiar si un proyecto es beneficioso dada una serie de variables que especifican datos asociados a los costos y beneficios que éste tendrá. El dominio se puede clasificar como bien definido y las tareas a su vez son bien definidas, pues la resolución de un problema recae principalmente en la aplicación de índices económicos como el VAN (Valor Actual Neto) o el TIR (Tasa Interna de Retorno) (Khan y Jain, 1999).

Este entorno, es uno de los dos que no han sido desarrollados específicamente para esta tesis. Por el contrario, uno de los compañeros dentro del grupo de investigación ha sido el autor de la misma. El objetivo de esta aplicación, además de servir como herramienta de soporte para la asignatura de Gestión de Proyectos (asignatura de Ingeniería Informática), ha sido probar los modelos de evaluación que están siendo investigados dentro del grupo. En este sentido, una vez desarrollada, se aplicaron los pasos mencionados en la sección 6.5.3 para dotar de las capacidades de evaluación del MBR y la TRI en un entorno existente.

La herramienta, desarrollada en *.NET*, es una aplicación Web que intenta centrar el esfuerzo del alumno en la resolución de problemas. Como pasaba con Simplex Tutor, el tipo de problemas que se pueden resolver tiene inherente una sobrecarga cognitiva asociada a los cálculos necesarios (Sweller et al., 1998). Con el mismo objetivo, se intenta minimizar este obstáculo proporcionando a los estudiantes con mecanismos parecidos a los de una hoja de cálculo para utilizar referencias a las celdas de una tabla en la creación de fórmulas que serán interpretados automáticamente y calculadas por el sistema. Los estudiantes deben rellenar información del problema en una tabla y otros valores que, juntos, representan la solución al problema.

Puesto que el sistema no se ha desarrollado dentro de la tesis aquí presentada, no se entrará en detalles sobre su arquitectura. Tan sólo es relevante para esta tesis la parte que permite usar CBMEngine como una componente más del entorno. Respecto a esta parte, el sistema fue diseñado para que la información obtenida de la interacción con el estudiante pudiera ser evaluada de diferentes formas. Para lograr esto, el sistema dispone de una componente que permite el envío de información mediante servicios Web a los sistemas de evaluación contemplados en PIPSE. Estos sistemas son independientes y pueden ser reemplazados dinámicamente, agregándose o eliminándose. Aunque en la actualidad existen dos sistemas de evaluación funcionando, cada uno asociado con una metodología diferente, sólo el CBMEngine es relevante en este trabajo.

La componente encargada de la comunicación en PIPSE, iniciará sesión en CBMEngine y proporcionará la información sobre la solución dada por el alumno para que CBMEngine sea capaz de aplicar la evaluación. Los errores cometidos que detecta CBMEngine son recogidos por PIPSE y presentados en la interfaz del alumno. Esta característica hace que el sistema no sólo sea una herramienta de evaluación, sino también, un entorno de aprendizaje. La figura 6.20 muestra un ejemplo de esta situación en la que PIPSE muestra al alumno los errores detectados, resaltando en rojo los elementos de la interfaz que han producido el error. Además, se pueden ver las cuatro partes principales de PIPSE: etiquetado con "A", un panel de acciones relacionadas con la sesión actual y los intentos del estudiante; "B" contiene el título del problema y botones para ocultarlo y mostrarlo; "C" es la tabla con la solución del estudiante, la cual puede ser editada; y "D" contiene los controles para añadir componentes a la

solución, las variables solución, y un panel de área de trabajo donde todas las acciones llevadas a cabo por el estudiante están representados, ofreciendo también la posibilidad de introducir instrucciones a través de un intérprete de comandos.

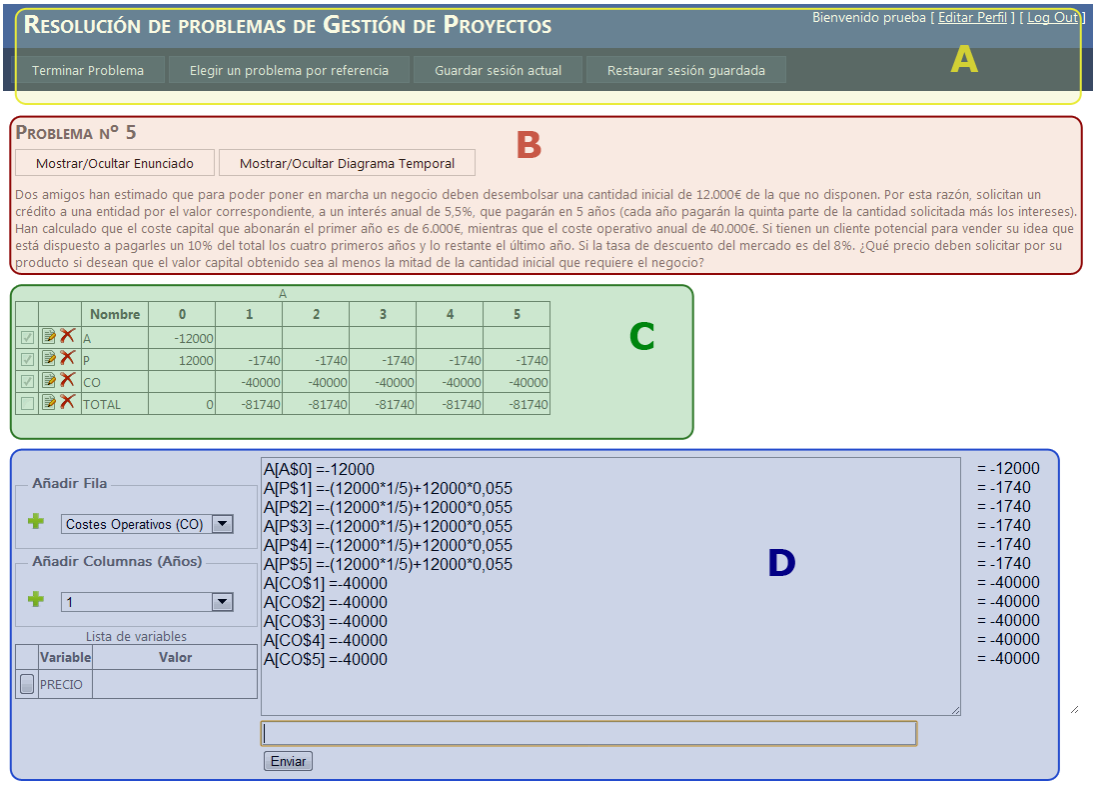



Figura 6.20: Informe de errores cometidos en la interfaz del sistema PIPSE.

Para lograr que PIPSE pudiese utilizar la metodología combinada presentada en esta tesis, y tal y como se mencionó en el aparatado 6.5.3, fue necesario crear en CBMEngine el tratamiento de las estructuras externas y de las restricciones asociadas al modelo de dominio. En este caso, ambas se programaron directamente en Java al no estar completamente integrado CBMEngine con la herramienta CBM-DoME (ver sección 6.6). Tras consultar con profesores expertos en la asignatura, un total de 17 restricciones fueron definidas para el dominio. Como se puede observar, es un dominio con un número reducido de restricciones, al igual que sucede en Simplex Tutor. No obstante, las reglas se pueden agrupar en tres subconjuntos: (a) cálculo e inferencia asociada con la solución, compuesto por 2 restricciones; (b) manipulación de datos de la tabla solución, con 7 restricciones; y (c) definición correcta de las variables relacionadas con el problema, que contiene 8 restricciones.

#### 6.5.4.2. Visual Nets

El segundo sistema de DEDALO que funciona con CBMEngine es *Visual Nets* (VN para abreviar de ahora en adelante). El nombre del sistema sería traducido al español como "Redes Visuales" y se centra en el dominio de las redes de ordenadores. Concretamente, en los conceptos relacionados con el manejo y conocimiento de las

direcciones IP. En este sistema se pueden resolver 5 tipos de problemas diferentes: 1) operaciones básicos sobre redes IP, en los que se pone en práctica el cálculo de la Máscara de red, dirección de difusión y de red, a partir de una dirección IP dada; 2) fragmentación de paquetes, relacionados con el modo en que se dividen los paquetes de datos al viajar por una red; 3) ensamblado de paquetes, que se asocian a la unión de los paquetes divididos al llegar a un destino dado; 4) asignación de IP, en los que se debe realizar una asignación de IP a un conjunto de equipos dentro de una red, dado un conjunto de restricciones; y 5) enrutamiento de paquetes, mediante los cuales, se practica la forma en que un paquete viaja a través de diversas redes hasta llegar a una entidad destino. Las actividades a realizar para proporcionar una solución en cada tipo de problema están bien definidas, por lo que VN se sitúa dentro de los dominios bien definidos con tareas bien definidas.

Esta herramienta es otra instancia particular del marco de trabajo DEDALO que se pretende usar como herramienta de apoyo en la asignatura de Redes, perteneciente al grado de Ingeniería de Telecomunicación. VN implementa una interfaz Web para recoger datos del alumno en la resolución de problemas. Para este sistema tampoco se explicará la arquitectura, ya que la parte relacionada con esta tesis es la componente del sistema que se encarga de pasar los datos recopilados a CBMEngine, la cual lo hace utilizando el conjunto de servicios explicado en la sección 6.5 y que se extiende en el apéndice A.

El comportamiento de VN para invocar a CBMEngine y recoger los errores es el mismo que en PIPSE, con la diferencia de que VN usa como estructuras externas documentos XML y Java Beans para almacenar la información (PIPSE sólo usa Java Beans). Este mecanismo será igual en cualquier sistema que integre la plataforma de evaluación CBMEngine. Lo único que será diferente en cada sistema externo será la forma de presentar el refuerzo que les llegue desde CBMEngine. Al igual que en PIPSE, aquí también se aprovecha la característica de identificadores de la interfaz para señalar los errores del alumno. Un ejemplo de esto se puede ver en la figura 6.21, la cual está asociada a los problemas de fragmentación de paquetes. Aquí, en la parte etiquetada como “C”, se muestran los errores cometidos y al pasar el ratón por encima de ellos se resaltará en rojo el elemento de la interfaz que tiene el error (en este caso el valor 700). Otros elementos de la interfaz son: etiquetado con “A”, un panel perteneciente a la componente de interfaz DEDALO que permite el control de la sesión, configuración de la aplicación, y navegación; bajo la etiqueta “B”, se muestra el enunciado del problema, aunque al estar mostrando errores éste queda casi oculto; la etiqueta “D” corresponde al área de trabajo donde el alumno puede ir dividiendo los paquetes e introduciendo sus valores.

Al igual que en el sistema PIPSE del apartado anterior, y pese a que éste se ha integrado recientemente, todavía no estaba disponible la integración entre CBMEngine y CBM-DoME, por lo que las restricciones y las estructuras externas tuvieron que ser codificadas e integradas manualmente en la plataforma. Aquí también se identificaron las restricciones en colaboración con los profesores de la asignatura para la que se desarrolló el sistema. En total, 37 restricciones se codificaron en este dominio. La agrupación realizada en las restricciones se corresponde a los diferentes tipos de problemas 6 restricciones para los problemas de operaciones básicas sobre IP; 13 en los problemas de fragmentación; 6 en los de ensamblado; 6 en los de asignación de IP; y 6 restricciones en los problemas de enrutamiento.

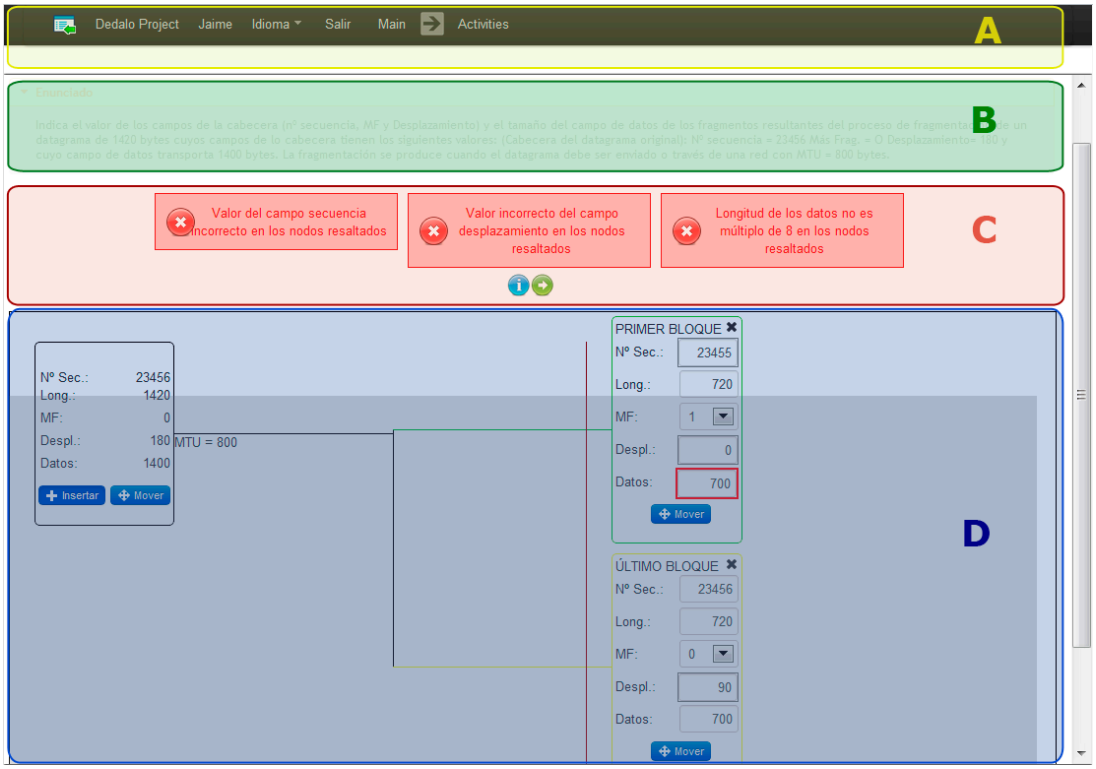



Figura 6.21: Tratamiento de los errores del alumno particular de Visual Nets.

## 6.6. Framework CBM-DOME

En los dos ejemplos de sistemas integrados con CBMEngine la elicitación de los elementos particulares del dominio se tuvo que hacer de forma manual codificando las estructuras directamente en Java y las reglas en el lenguaje DROOLS. Aunque es una tarea que no es muy complicada, si no se posee conocimiento sobre programación y sobre el lenguaje DROOLS, puede llegar a convertirse en una tarea complicada. Por ello, y con el fin de favorecer la utilización de CBMEngine en sistemas que quieran utilizar sus servicios, nació *CBM-DoME* (Fernández y Gálvez, 2011). El nombre proviene del Inglés *Constraint-Based Modeling Domain Model Editor* (editor de modelo de dominio para el MBR). Aunque esta tarea no es crítica para probar los modelos teóricos, sí que es atractiva de cara a la futura construcción de sistemas que los usen.

El sistema es una herramienta Web implementada con JSP y la biblioteca Smart-Client (Isomorphic Software, 2009). Esta última proporciona un conjunto de componentes gráficas mediante el lenguaje Javascript, que generan una interfaz fácil de manejar e intuitiva para el usuario. El objetivo de CBM-DoME es facilitar en la medida de lo posible el proceso de definición del dominio. La interfaz está compuesta por dos editores, asociados a cada elemento del dominio que se puede modelar.

Respecto al primero de los editores, asociado con las estructuras externas, para construir de forma sencilla los Beans de Java, simplemente hay que especificar qué campos van a tener éstos, su tipo, y un nombre. De esta forma, la construcción de un Bean se puede simplificar a la construcción de una lista de campos que formarán el Bean. En la interfaz (figura 6.22), la lista de campos actuales se puede ver en la

parte sombreada bajo la etiqueta "B", donde cada fila de la lista es un campo. Los campos se pueden editar, borrar, o añadir mediante botones o haciendo clic con el ratón. En el caso de añadir nuevos campos, se debe seleccionar un tipo asociado de entre los disponibles en el panel etiquetado con "A", y arrastrar este tipo a la lista de campos. Actualmente para modelar diferentes tipos de datos se da la posibilidad de usar tipos básicos asociados al lenguaje Java (entero, natural, carácter, booleano, cadena de caracteres) o tipos complejos (array, vector, u otro Bean). Además, dado que los Beans son objetos Java y que estos se utilizarán en las reglas, puede ser necesario determinadas funciones para las comprobaciones que no sean simples, por lo que también se da la posibilidad de añadir funciones Java en el panel "C" para usuarios avanzados. Al finalizar la edición de un Bean, se comprobarán que no hay errores y generará el Bean Java asociado.

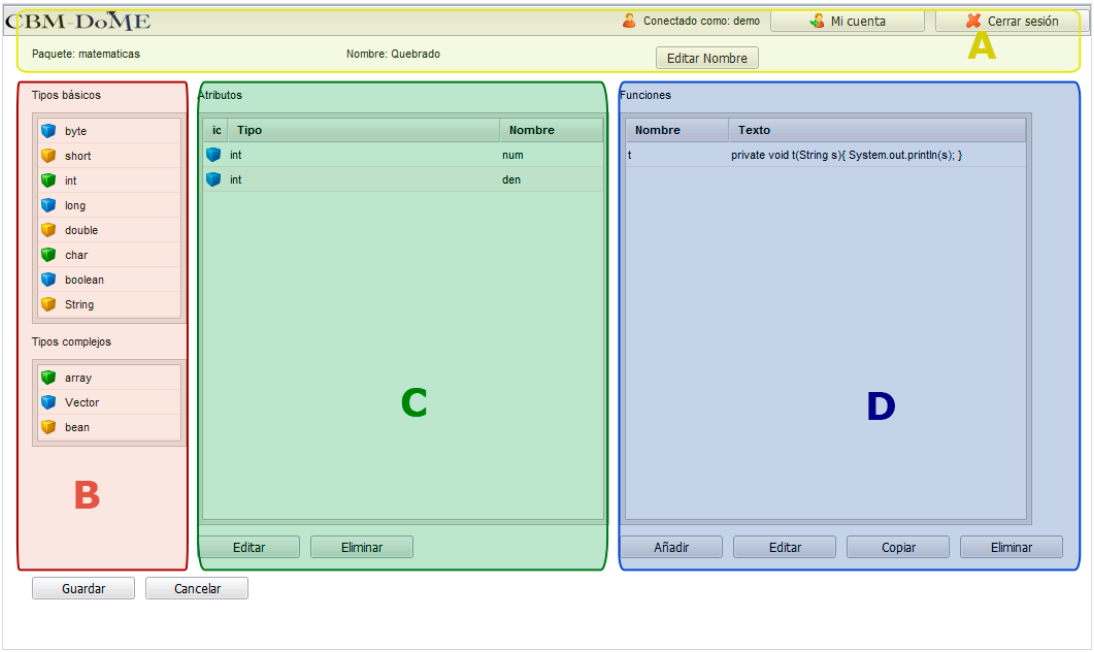



Figura 6.22: Interfaz de la edición de estructuras en CBM-DoME.

El segundo de los editores permite la creación de reglas a partir de los beans que se han creado anteriormente. Las reglas tendrán un antecedente y un consecuente. El antecedente estará formado por una combinación de expresiones booleanas que determinarán la condición de satisfacción y de relevancia. Estas expresiones se forman seleccionando los campos de la estructuras sobre las que se va a comprobar algo y añadiendo operaciones básicas de comparación (igual, distinto, mayor que, menor que, etc.). El consecuente tendrá el tratamiento de la restricción que normalmente seguirá la misma estructura en todas las reglas, tan sólo siendo necesario personalizar el refuerzo asociado. En este editor también se da la posibilidad a usuarios avanzados de personalizar el consecuente con alguna acción especial (normalmente no es necesario). Para ello, se permite introducir invocaciones a funciones Java, siempre dentro de las que estén definidas como parte de los Beans construidos anteriormente.

Esta primera versión del marco de trabajo cuenta con la posibilidad de crear las estructuras, hasta las más complejas, y de la creación de reglas, aunque de una forma

todavía por mejorar. En este sentido, existe una limitación en cuanto a las operaciones disponibles para construir las reglas, ya que permiten operandos muy básicos. En caso de que se quisiera realizar comprobaciones sobre una estructura del tipo tabla (por ejemplo implementada mediante un array de arrays), las comparaciones simples no servirían, siendo necesaria la utilización de funciones Java nativas. Esto es algo que se tiene previsto paliar en versiones futuras extendiendo el sistema con nuevas operaciones de comparación más complejas.

## 6.7. Conclusiones del capítulo

En este capítulo se han presentado las diferentes herramientas que se han utilizado para implementar los modelos de evaluación teóricos del MBR con la TRI. Siguiendo con la metodología de trabajo que se planificó como parte de los objetivos de esta tesis (ver la sección 1.2), se ha usado un enfoque de desarrollo desde lo más específico a lo más general. Así pues, primero se construyeron herramientas específicas, para posteriormente moverse a un nivel más general.

La primera herramienta desarrollada fue OOPS, un EIRP en el dominio de la POO, el cual es bien definido y con tareas débilmente definidas. El modelo del dominio está compuesto por un total de 86 restricciones que se comprueban en un motor de reglas JESS. El sistema incorporaba unos heurísticos iniciales para la actualización del modelo del alumno con el nivel estimado. La selección adaptativa se realizaba originalmente también mediante heurísticos y utilizando un método parecido al de máxima verosimilitud de la TRI.

Posteriormente, se desarrolló Simplex Tutor, el cual incorporaba diferencias significativas, principalmente por la tecnología usada, la cual hizo más fácil el desarrollo del sistema. Su dominio es la optimización lineal en el algoritmo Simplex y el método de las dos fases, el cual es bien definido con tareas bien definidas y consta de un conjunto reducido de 17 restricciones. Estas restricciones se comprueban en un motor de inferencia JBoss Rules, el cual permite la utilización directa de objetos Java, siendo más sencillo de utilizar, dando mayor potencia y haciéndolo más eficiente que el sistema OOPS. Si bien el sistema carece de selección adaptativa integrada de problemas mediante la TRI, el desarrollo del mismo permitió abstraer patrones y elementos comunes en la arquitectura de EIRP que darían lugar posteriormente a CBMEngine.

Respecto al modelo teórico para la evaluación en dominios procedimentales mediante sistemas de tests, expuesto en la sección 5.1, se implementaron los ítems compuestos en el sistema Siette. Como ya se mencionó, estos ítems agrupan un conjunto de evidencias, cada una modelada mediante ítems componente. Además de servir como modelo equivalente para las restricciones y los problemas MBR, los ítems componentes podrían implementar cualquier tipo de ítems como los de opción múltiple o de respuesta corta.

Fruto de los resultados iniciales de la investigación, se estableció un trabajo de colaboración con los autores de SQL-Tutor, una de las herramientas más prolíficas en el MBR. Aunque esta herramienta carece de un mecanismo de evaluación formal, sirvió como fuente de datos para experimentar y mejorar nuestra metodología (ver capítulo 7). La parte relevante de la herramienta que hemos usado se refiere al modelo del alumno que recoge la actividad del mismo en el sistema y que se puede usar como evidencia para aplicar nuestros modelos. Para procesar la información se implementó SQL-Tutor Processor, una herramienta que es capaz de aplicar las técnicas de evaluación a posteriori y

de manera independiente. La herramienta utiliza la información existente para formar la matriz de rendimiento, formatear los datos, e invocar a la herramienta MULTILOG que se encarga de aplicar los mecanismos de la TRI. Las opciones con las que cuenta la herramienta permiten personalizar cada etapa del proceso de calibración y evaluación. Aunque SQL-Tutor Processor ha sido diseñada para procesar la información procedente de SQL-Tutor, ésta puede procesar la información de cualquier herramienta MBR que mantengan un formato similar en su fichero de actividad.

A partir de la experiencia obtenida en los sistemas anteriores y generalizando la estructura de éstos y el modelo, se desarrolló CBMEngine, un marco de trabajo que proporciona nuestro modelo de evaluación mediante un conjunto de servicios Web a EIRP externos. Usando CBMEngine, un sistema externo sólo tiene que definir las estructuras y reglas involucradas en el modelo de dominio, el cual sigue usando el motor JBoss Rules para el funcionamiento. El modelo del alumno es parte del sistema y se actualiza a partir de las evidencias que los sistemas externos proporcionan al motor. La evaluación y adaptación mediante la TRI forman parte de los servicios proporcionados, los cuales, internamente, funcionan usando el sistema Siette o MULTILOG. La plataforma fue integrada con PIPSE y Visual Nets, dos sistemas del marco de trabajo DEDALO, cuyo objetivo es proporcionar servicios de evaluación y modelado del alumno para la construcción de EIRP. CBMEngine es equiparable a la herramienta de autor WETAS mencionado en el apartado 2.3.7.6, pues proporciona un entorno sobre el que ejecutar las técnicas de modelado y evaluación para sistemas tutores, con la diferencia de que CBMEngine utiliza mecanismos formales y además dispone de CBM-DoME para la autoría del dominio (WETAS no dispone de ninguna asistencia para esta tarea). Sin embargo, todavía necesita extender varias componentes para poder compararse con la madurez de ASPIRE, la herramienta de autor más importante en el MBR.

Por último, se desarrolló CBM-DoME para ayudar en la construcción de las estructuras y reglas necesarias que se utilizan en CBMEngine de forma automática. La herramienta es una extensión que, hasta la fecha, permite la creación de estructuras y reglas simples sobre éstas, siendo necesario extender su funcionalidad para alcanzar todo el potencial que se tendría de usar DROOLS en su forma nativa en la codificación de restricciones. Este marco de trabajo no se ha utilizado todavía para desarrollar ningún sistema, pero se pretende incorporar como componente de integración en la herramienta CBMEngine. CBM-DOME, como herramienta de autor, queda muy atrás todavía de la herramienta ASPIRE explicada en la sección 2.3.7.6, ya que la funcionalidad que proporciona sólo contempla la creación de dos de las componentes del dominio y la forma de construirse es todavía rudimentaria, en comparación con ASPIRE, que descubre restricciones mediante un proceso semi-automático y permite la autoría de un tutor al completo. No obstante, es una herramienta relativamente joven y el trabajo futuro pretende cubrir algunas de estas diferencias.

Las herramientas utilizadas, Siette, MULTILOG, CBMEngine, y CBM-DoME, funcionan como piezas de un puzle, que encajadas correctamente permiten, por un lado la incorporación de los elementos necesarios para utilizar los modelos definidos en capítulos anteriores, y por otro la utilización de la evaluación y selección adaptativa de contenidos.

# Parte V

# Evaluación

El contenido de esta parte se centra en explicar los diferentes estudios empíricos realizados para probar la validez y la efectividad de los modelos teóricos desarrollados.

# Capítulo 7

# Experimentación

*Son vanas y están plagadas de errores las ciencias que no han nacido del experimento, madre de toda certidumbre*

Leonardo da Vinci (1452 - 1519)

**Resumen:** En este capítulo se explican los detalles de cada experimento realizado para estudiar de forma empírica diferentes características sobre los modelos teóricos presentados en capítulos anteriores.

De forma general, la investigación implica la formulación de hipótesis y modelos que, mediante la aplicación de los mismos, permitan determinar si las hipótesis son ciertas. En este sentido, para estudiar los modelos desarrollados durante esta tesis es necesario recurrir al método empírico en el que el estudio se basa en experiencias, a partir de las cuales se recopilan evidencias sobre las hipótesis, con el fin de determinar si éstas son ciertas o no. Tal y como menciona Chin (2000), el estudio empírico en el modelado del alumno es un hecho que, aunque no todos los autores realizan, es fundamental para poder garantizar la aplicabilidad de los modelos.

Las hipótesis que se han planteado en esta tesis pasan por la aplicabilidad de un modelo de evaluación sumativa y la posibilidad de usarse para realizar una evaluación formativa que favorezca el proceso de aprendizaje del alumno. Es por ello que es necesario estudiar las características que hacen que el modelo sea aplicable y que su aplicación es efectiva. En relación con el modelo de evaluación sumativa desarrollado, además, de la aplicabilidad, sería conveniente estudiar otras propiedades deseables para que la metodología sea lo más objetiva y formal posible. Dos de las propiedades más importantes son, según se revisó en el apartado 1.1.3, la validez y la fiabilidad del mecanismo de evaluación.

Respecto a la validez, en el ámbito al que se refiere esta tesis, una evaluación será válida si es capaz de medir de manera precisa el conocimiento del estudiante para el que está diseñada. En el área de la Psicometría existe un amplio abanico de definiciones acerca de la validez (Moss et al., 2006). Sin embargo, dado que estudiarlas todas supondría un trabajo extensísimo que no hubiera permitido avanzar en otros aspectos, esta parte de la tesis se ha apoyado en una de las definiciones de validez. Concretamente, en la denominada *validez del constructo* que según Sampieri et al.

(2006) es una de las más importantes, sobre todo desde una perspectiva científica. Este tipo de validez usa el concepto de constructo como una variable medida que tiene lugar dentro de una teoría o esquema teórico, y juzga el instrumento de medida respecto del grado en que una medición se relaciona consistentemente con otras mediciones sobre conceptos que están midiéndose.

Con el fin de determinar la validez del método de evaluación sumativa, de acuerdo a la validez del constructo, se ha optado por comparar el resultado con el de otra metodología que produzca un juicio formal y objetivo del conocimiento. En nuestra experimentación hemos optado por usar como punto de referencia para la comparación la metodología propuesta que hasta ahora es, posiblemente, la forma más objetiva y formal de medir el conocimiento del alumno: los tests basados en la TRI. Como parte de la definición de la mencionada validez, y para garantizar la objetividad de la comparación, es importante que los dos instrumentos utilizados, tanto los ítems, como las restricciones, estén intentando medir el conocimiento sobre los mismos conceptos. Es por ello que los conceptos que intentan evaluar los ítems presentados en el test deben tener una correspondencia directa con los conceptos que modelan las restricciones.

Otro aspecto importante de la evaluación formal es la fiabilidad (ver apartado 3.2.2). Esta característica se refiere a la consistencia que un instrumento de medida proporciona. Es decir, debe proporcionar los mismos resultados ante las mismas circunstancias y resultados similares ante circunstancias similares. En este sentido, la metodología de evaluación diseñada se asienta en los mecanismos formales de la TRI.

Es necesario destacar una característica sobre los experimentos que se han realizado para probar los modelos teóricos. Todos y cada uno de ellos han basado sus conclusiones mediante la puesta en práctica y la recopilación de información sobre una población real. El uso de estudiantes reales para realizar cada prueba, en lugar de usar estudiantes virtuales, supone una dificultad adicional a la hora de realizar estos estudios por varios motivos.

- En primer lugar, es necesario contar con la aprobación de profesores que quieran colaborar en la experimentación, permitiéndonos aplicar las herramientas con sus alumnos.

- En segundo lugar, la aceptación del profesor no es suficiente para realizar una prueba, pues es necesario disponer de una herramienta sobre el dominio que abarca la asignatura. En caso de no disponer de ella, se requiere un esfuerzo adicional de desarrollo, que puede ir desde los tres personas-mes en sistemas simples (como es el ejemplo de Simplex Tutor), a un persona-año, en sistemas más complejos (ejemplo del sistema OOPS). Este problema se puede ver acentuado con los cambios docentes de la entidad en donde se realizan los estudios. Sin ir más lejos, los cambios de profesores en la asignatura sobre la que pretendíamos aplicar el tutor OOPS, supuso que sólo uno de ellos quisiera colaborar. Su posterior desaparición del plan de estudios la convirtió en una herramienta inaplicable, siendo necesario modificar el lenguaje que utiliza para poderse aplicar de nuevo en materias relacionadas existentes.

- Como tercer motivo, la experimentación está influida por el número de alumnos de la asignatura. Incluso contando con la colaboración del profesor y de la herramienta, si el número de alumnos es muy reducido, los resultados son muy proclives a no ser significativos.

Esta problemática para realizar la experimentación ha estado presente durante toda la tesis, por lo que la experimentación realizada ha estado limitada por la disponibilidad de los alumnos y no ha podido abarcar los aspectos que nos hubiera gustado.

La experimentación se centra principalmente en el estudio de la aplicación de los modelos teóricos de evaluación sumativa del alumno. En relación con el modelo teórico de evaluación formativa, el estudio se ha centrado sólo en la parte inicial, correspondiente a la traza del conocimiento. Esto es así porque su desarrollo se ha finalizado en la etapa final de esta tesis y, tal y como se comentaba anteriormente acerca de los problemas para poder realizar la experimentación sobre estudiantes reales, no hemos contado con la oportunidad realizar una experimentación adecuada para estudiar la efectividad de los modelos, por lo que se propone como parte de las líneas abiertas.

Los experimentos realizados para estudiar la validez del mecanismo de evaluación sumativa han intentado abarcar diferentes tipos de dominios. Usando como base la clasificación de los dominios en los que se puede aplicar el MBR, explicada al principio del capítulo 2, se han utilizado dos tipos: dominios procedimentales sencillos con tareas bien definidas, y dominios procedimentales complejos con tareas débilmente definidas. Siempre quedándonos dentro de la dimensión de dominios bien definidos.

Como se explicaba en la sección 4.5.1, el modelo teórico planteado, con el objetivo de ser genérico, proponía una abstracción sin tener en cuenta el modelo de la TRI utilizado. No obstante, debido a que para poder aplicar la metodología hay que usar algún modelo concreto, en las pruebas que se explican en este capítulo, se ha optado por utilizar modelos paramétricos. La decisión de usar los modelos paramétricos radica en su simplicidad de cálculo en comparación con los no paramétricos, además de que son deseables si el modelo se ajusta a los datos, ya que la reducción de información en su manejo es considerable (sólo sería necesario usar los parámetros de las CCR, en lugar de todos los valores). No obstante, esto no implica que los modelos no paramétricos sean peores, siendo necesario un futuro estudio sobre ellos.

Como ya se introdujo en la sección 1.1.1, la evaluación puede ser aplicada a diversos elementos. Aunque en esta tesis se han usado los términos evaluación formativa y sumativa para referirse de forma general a la evaluación sobre el estudiante, también pueden aplicarse a sistemas. En esta línea, autores como Littman y Soloway (1988); Shute y Regian (1993); Winne (1993) utilizan el significado de estos términos aplicados a sistemas de aprendizaje. No obstante, el significado tiene un elemento común y de naturaleza similar al de la evaluación aplicada a un alumno: la sumativa se aplica para conocer la efectividad del sistema, similar a la emisión del juicio sobre un alumno; y la formativa para conocer los elementos que se pueden mejorar, parecido al uso de la evaluación para mejorar el aprendizaje.

Los dos tipos de evaluación mencionados son inherentes al proceso de experimentación realizada con los sistemas implementados. Además de la evaluación sumativa, realizada para determinar la validez de los modelos desarrollados y su aplicabilidad, se ha realizado una evaluación formativa constante que ha tenido lugar tanto en el desarrollo de los mismos, como en la experimentación. En el desarrollo el refuerzo característico de la evaluación formativa ha sido proporcionado por expertos en los dominios de aplicación, mientras que en la experimentación se han utilizado encuestas de opinión al finalizar cada experiencia, lo que nos permitía obtener de mano de los alumnos otro tipo de refuerzo para la mejora de los sistemas. Dado el carácter subjetivo de esta evaluación, no será incluida como parte de la experimentación, la cual se centra en la evaluación sumativa. Tan sólo se menciona que el resultado más destacable obtenido

de las encuestas es la impresión positiva y generalizada de los alumnos ante el uso de cada sistema evaluado.

Este capítulo se estructura de la siguiente forma, en primer lugar, se presenta la evaluación realizada sobre una de las piezas claves para aplicar la metodología de evaluación: la efectividad del MBR como herramienta para modelar e instruir al alumno en EIRP. Posteriormente, se presenta la experimentación realizada para validar los modelos teóricos de evaluación en dos tipos de dominios de naturaleza diferente. Primeramente, en la sección 7.2, se estudia la aplicabilidad en dominios bien definidos y con tareas simples y bien definidas. Seguidamente, en la sección 7.3, se presenta el estudio realizado en dominios con tareas débilmente definidas y no acotadas. En la sección 7.4 se presentan varios estudios realizados sobre la propiedad de invariancia de las estimaciones que se pueden obtener con el modelo. A continuación, se presenta la experimentación realizada para estudiar el modelo más preciso que sería conveniente utilizar como parte de la evaluación formativa, en relación con la calibración de las restricciones (sección 7.5) y la evaluación (sección 7.6). En la sección 7.7 se detalla la experimentación realizada sobre la técnica que determina la calidad de las restricciones a la hora de ser usadas en la evaluación formal. Finalmente se exponen las conclusiones del capítulo.

## 7.1. Evaluación del MBR como herramienta de modelado

Los resultados de este experimento fueron publicados en una revista indexada en el ISI JCR (*Journal Citation Reports*): (Gálvez et al., 2009a). Hasta entonces sólo disponíamos de unos modelos iniciales, que fueron presentados en (Gálvez et al., 2007). Éstos eran modelos teóricos que se centraban en el MBR pero no habían sido implementados todavía. A continuación se detalla el trabajo publicado, haciendo hincapié en la evaluación empírica que se realizó. De cara a esta tesis, el objetivo principal era estudiar la validez del MBR como herramienta de modelado del alumno, como paso previo a la implementación de los modelos de evaluación. Pero además, existía una motivación adicional muy importante para las asignaturas que participaron en la experiencia. Esta motivación adicional se detalla primeramente y, a continuación, se explica el estudio realizado.

A pesar de que la estrategia de aprendizaje más común sigue siendo las clases presenciales, impartidas oralmente por un profesor, el número de sistemas alternativos de aprendizaje ha aumentado. El proceso de aprendizaje ideal es aquel donde los estudiantes pueden recibir clases, resolver ejercicios y obtener una respuesta inmediata por parte del profesor. Por desgracia, la masificación en las aulas hace que esta situación deseable no sea factible. Hoy en día, los profesores tienen que impartir docencia a docenas o incluso cientos de estudiantes, por lo que es difícil para ellos asimilar correctamente los conceptos que se enseñan. Mediante la adopción de sistemas de aprendizaje, tales como los STI, los profesores podrían abordar esta situación de masificación usando un *aprendizaje mixto* (conocido en inglés como *blended learning*). Esta es una estrategia de aprendizaje se basa en la incorporación de diferentes modos de enseñanza y estilos de aprendizaje. El objetivo es introducir múltiples medios de comunicación para facilitar el diálogo alumno-profesor (Heinze y Procter, 2006). Existen varios sistemas como Assistment (Razzaq et al., 2007), que se han utilizado con éxito en experiencias de aprendizaje mixto.

La problemática del escenario de masificación de los estudiantes, mencionado anteriormente, es una de las causas que motivó la realización de la experiencia descrita en esta sección. Varios profesores se encargaban de la instrucción de estudiantes universitarios, específicamente, en el estudio de programación avanzada en el segundo semestre de Ingeniería Técnica de Telecomunicación en la Universidad de Málaga (España). Alrededor de 300 personas estudiaban esta asignatura cada año, la cual dejó de impartirse recientemente al desaparecer el plan de estudios. Tres profesores se encargaban de introducir a los estudiantes los conceptos de POO. Hasta entonces, los alumnos sólo contaban con una asignatura sobre los conceptos básicos de programación imperativa. Cada profesor impartía docencia en dos grupos de alrededor 50 estudiantes, y con un programa de la asignatura muy denso. Por esta razón, el tiempo de clase disponible para resolver problemas de programación o para ayudar a los estudiantes a desarrollar programas era muy limitado. En consecuencia, los profesores de la asignatura decidieron introducir una estrategia de *aprendizaje mixto* para facilitar el proceso de aprendizaje de los estudiantes. Esta estrategia consistía en dejar utilizar a los estudiantes un STI para la resolución de problemas de POO, a la vez que se realizaban las clases presenciales, y se permitía el acceso al sistema Siette, explicado en el apartado 6.3. La herramienta proporcionada se corresponde a la primera versión de OOPS, explicada en la sección 6.1.

Esta experiencia de aprendizaje mixto se incorporó en 2008 a una asignatura de programación avanzada que recibía el nombre de *Elementos de Programación*, impartida durante el segundo semestre (aproximadamente 14 semanas, con sólo dos horas por semana) en la E.T.S. de Ingeniería de Telecomunicación de la Universidad de Málaga. La figura 7.1 ilustra el programa de la asignatura. Para aprobar la asignatura, los estudiantes debían asimilar correctamente el concepto de abstracción de datos y su implementación utilizando un lenguaje de POO. Estas nociones se introducen en el tema 1, como puede verse en la figura, y son vitales para los estudiantes, ya que el resto de la asignatura se basa en ellas. La aplicación de los *Tipos Abstractos de Datos* (abordado en los temas 2, 3 y 5) se realiza utilizando POO.

Las estrategias pedagógicas aplicadas en esta asignatura consistían en clases presenciales. Estas clases eran teóricas y los profesores explicaban los conceptos que el estudiante debía asimilar. Al final de cada lección, se les proponían a los estudiantes una serie de ejercicios que debían ser resueltos en casa. Como parte de la asignatura había también clases prácticas, en las que los profesores resolvían los problemas propuestos en pizarra, a petición de los estudiantes. Esta última estrategia demostró ser muy ineficaz, a juzgar por la tasa de fracaso escolar. Al final del semestre, los estudiantes tenían que aprobar un cuestionario y un examen consistente en un problema de programación. El cuestionario evaluaba los conceptos teóricos impartidos durante el semestre y representa el 30 % de la puntuación final (3 puntos), siendo necesaria la obtención de al menos 1,5 puntos. De lo contrario, el profesor no corregía el examen y el estudiante suspendía automáticamente. Los resultados obtenidos al final de cada curso, ilustraban que esta estrategia no era la adecuada. La tasa de fracaso era bastante alta y, de acuerdo con los profesores, los estudiantes tenían algunos errores conceptuales importantes acerca de la abstracción de datos desde el inicio del semestre, haciendo que les resultara muy difícil seguir la asignatura.

La solución ideal para resolver esta situación implicaba la corrección individual de cada una de las soluciones que los estudiantes construían para los problemas propuestos durante el semestre. De esta forma se podrían identificar conceptos erróneos y propor-

**1. Introducción a la abstracción de datos**
- 1.2 Abstracción de datos y Programación Orientada a Objetos
- 1.2 Conceptos básicos de Programación Orientada a Objetos
- 1.3 Conceptos avanzados de Programación Orientada a Objetos

**2. Tipos de datos abstractos lineales**
- 2.1 Concepto de Tipo Abstracto de Datos(TAD)
- 2.2 Pila: Definición, ejemplos e implementación
- 2.3 Cola: Definición, ejemplos e implementación
- 2.4 Lista Posicional: Definición, ejemplos e implementación

**3. Memoria dinámica**
- 3.1 Gestión física de la memoria dinámica
- 3.2 Punteros
- 3.3 Listas enlazadas
- 3.4 Clases y memoria dinámica
- 3.5 Implementaciones dinámicas de TADs

**4. Recursividad**
- 4.1 Concepto
- 4.2 Implementación física
- 4.3 Utilización y ejemplos

**5. TADs no Lineales**
- 5.1 Árboles binarios: Definición, ejemplos e implementación

Figura 7.1: Programa de la asignatura sobre la que se realizó la experimentación.

cionar la ayuda y refuerzo adecuados. Por desgracia, esto no era factible desde el punto de vista del profesor, debido a la gran cantidad de estudiantes matriculados. Para mejorar estos resultados negativos, se dio un primer paso orientado al uso de aprendizaje mixto. En el año anterior al experimento que trata esta sección (2007), se aplicó al final del semestre la técnica de poner a disposición de los alumnos cuestionarios abiertos. Durante un período limitado, se podían intentar los cuestionarios, pero no podían ver la corrección hasta el final de dicho período. Los resultados de esta experiencia previa fueron alentadores: el porcentaje de estudiantes que hicieron el examen final y aprobaron fue de 74 %, en comparación con el 49 % del 2006. Si en su lugar, se considera el porcentaje en relación con todos los alumnos matriculados, este porcentaje fue del 23 % frente al 14 % del 2006. Sin embargo, el porcentaje de estudiantes que se presentaron al examen final era sólo del 31 %, mientras que en el 2006 había sido del 30 %. Como puede verse, a pesar de este buen resultado, el porcentaje total de estudiantes que realizaron el examen era todavía muy pequeño. Los profesores de la asignatura consideraron que probablemente se debió a las dificultades de los estudiantes para asimilar el paradigma Orientado a Objetos desde el principio del semestre.

### 7.1.1. Diseño del estudio

Para solucionar este problema, en 2008, se decidió extender la estrategia de aprendizaje mixto mediante el uso de la herramienta OOPS, en su primera versión, explicada en el apartado 6.1. El fin de incorporar OOPS como parte de la asignatura era doble. Por un lado, nos permitiría evaluar la validez y efectividad del MBR como herramienta educativa, mediante sesiones controladas. Por otro lado, aprovechando la accesibilidad y disponibilidad características de un entorno Web, se dejaría disponible el uso de OOPS como entorno de aprendizaje que sirviera de soporte a las clases presenciales. De esta forma, los alumnos podrían practicar libremente en un entorno educativo cualquier

problema relacionado con los conceptos básicos de la POO. Puesto que la herramienta está basada en el MBR, al construir soluciones incorrectas, se advierte al alumno sobre éstos de forma que pueda actuar en consecuencia para corregir el conocimiento incorrecto.

Respecto del estudio controlado, después de la primera lección que introducía los conceptos básicos de POO, con una duración estimada de seis horas, se preparó una sesión especial. La hipótesis establecida para este experimento se puede enunciar como "el MBR es válido para modelar al alumno y su utilización es beneficiosa para el aprendizaje del mismo". Con esto queríamos probar la eficiencia que otros trabajos mencionaban sobre la técnica de modelado. Esta sesión tuvo lugar en los laboratorios de la E.T.S. de Ingeniería Informática y estaba compuesta de tres partes:

1. Se inició con un pre-test administrado a través de Siette. Este cuestionario estaba compuesto de 15 ítems de opción múltiple que evaluaban todos los conceptos relacionados con la abstracción de datos. Cada ítem tenía tres opciones y sólo una de ellas era correcta. Los estudiantes podían dejar cualquier ítem en blanco y tenían un límite de 15 minutos para completar la prueba.

2. Una vez que habían terminado la prueba, los estudiantes tenían que resolver problemas usando OOPS. Estos problemas eran similares a los incluidos en la lista de problemas propuestos por los profesores como parte de la asignatura. Cada problema consistía en la construcción de las partes pública y privada de una clase con ciertas características. Los estudiantes tenían que determinar y definir los atributos de clase y los métodos asociados a la misma, diferenciando si éstos eran públicos, para situarlos en la interfaz o si eran privados. OOPS proporcionaba refuerzo a los estudiantes bajo demanda. En este sentido, mientras los estudiantes estaban construyendo sus clases, podían hacer clic en un botón y el sistema indicaba el número de errores que tenían y las causas.

3. Después de interactuar con OOPS, se les administró un post-test a los estudiantes, similar al pre-test, que también se presentó utilizando la herramienta Siette. Nuestro objetivo fue explorar la mejora en el aprendizaje experimentada por los estudiantes después de su interacción con OOPS. El formato del test fue el mismo que en el pre-test. Es decir, 15 ítems de elección múltiple de tres opciones cada uno, y un límite de tiempo de 15 minutos.

El grupo de los estudiantes que realizaron la prueba era nuestro grupo experimental. Además, teníamos un grupo de control que asistió a las 6 horas de clase presenciales pero no trabajó con OOPS, ni tampoco realizó el pre-test. A este grupo de la muestra sólo se le administró el post-test. Los criterios utilizados para dividir a los estudiantes en estos dos grupos se establecieron de acuerdo a los grupos académicos a los que cada individuo pertenecía. Por último, señalar que la puntuación de ambas pruebas se expresó en términos de porcentajes, es decir, entre 0 y 100. La nota de corte fue del 50 %, es decir, los estudiantes que hubieran obtenido una puntuación igual o superior al 50 % superaron la prueba.

### 7.1.2. Análisis de los datos y resultados

Un total de 47 estudiantes participaron en este experimento, el cual tuvo lugar en marzo de 2008. 29 individuos formaron el grupo experimental, es decir, el grupo que

realiza los tres pasos del experimento descrito anteriormente. Al grupo de control (de 18 estudiantes) se le administró sólo el post-test. La tabla 7.1 resume los resultados de ambos grupos. Las dos primeras filas representan los datos del grupo experimental en el pre-test (primera fila) y en el post-test (segunda fila). La última fila contiene la información relativa al grupo de control. La tercera columna muestra el número de personas que participaron en cada test y la cuarta el porcentaje de ellos que aprobaron el test (puntuación igual o superior al 50 %). Las dos últimas columnas contienen la puntuación media de cada muestra y su desviación típica, respectivamente. Como puede verse, los resultados sugieren que a pesar de que el aumento de la media de la muestra para el grupo experimental no fue muy alta (12,70 %), el porcentaje de estudiantes que pasaron cada test se incrementó de 48 % al 81 %.

| | | Num. Est. | Aprobados ( %) | Media | Desv. Típ. |
|---|---|---|---|---|---|
| Grupo Experimental | Pre-test | 29 | 48 % | 43,79 | 21,94 |
| | Post-test | | 81 % | 55,24 | 25,22 |
| Grupo de Control | Post-test | 18 | 61 % | 44,32 | 26,68 |

Tabla 7.1: Comparación de los resultados del pre-test y post-test entre el grupo experimental y el de control.

También se realizó un análisis estadístico para ver si los resultados entre el pre-test del grupo experimental y el test del grupo de control eran similares o no. Para ello, se utilizó el clásico test estadístico de hipótesis (es decir, el test $t$ de Student o también conocido como t-test). Los resultados sugieren que no podemos rechazar la hipótesis nula que establece que las medias de las dos muestras son diferentes, (p-valor $>$0,8223) al 95 % de confianza. Por otra parte, se hizo una comparación por pares para determinar si los estudiantes del grupo experimental aumentaron su rendimiento después de trabajar con OOPS o no. Con este fin se realizó una t-test por pares comparando los resultados del pre-test y del post-test para cada individuo. El objetivo era averiguar si la diferencia entre el rendimiento de cada estudiante en ambas pruebas fue estadísticamente significativa. La evidencia sugiere que podemos rechazar la hipótesis nula (que establece que la media de cada persona antes y después de usar OOPS es similar) con p-valor $<$0,009115 al 95 % de confianza. La figura 7.2 ilustra la relación por pares mediante un diagrama de dispersión. Se puede observar que la mayoría de los estudiantes mejoraron su puntuación entre el pre y el post-test.

## 7.2. Validez de la propuesta en entornos bien definidos

El segundo experimento, fue publicado en el artículo (Gálvez et al., 2009c). Hasta entonces habíamos demostrado la validez del MBR como técnica de modelado del alumno y su efectividad como herramienta educativa. Entonces, habíamos desarrollado los modelos teóricos presentados en (Gálvez et al., 2007), pero éstos carecían de una demostración sobre su validez como metodología de evaluación. El objetivo principal de este experimento fue probar la validez práctica en un dominio procedimental reducido y con tareas que requieren una serie de pasos bien definidos y específicos para su resolución. Las características del experimento y los resultados son detallados a continuación.

En este experimento, para explorar la validez de los modelos de evaluación que

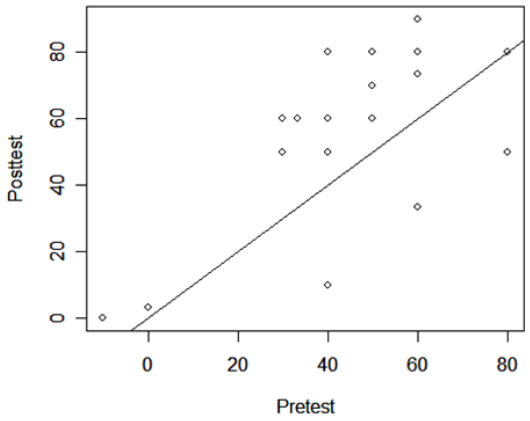



Figura 7.2: Diagrama de dispersión comparando los resultados de los individuos del grupo experimental entre el pre-test y el post-test.

combinan el MBR con la TRI, se estudió si la evaluación del conocimiento, mediante un EIRP que implementara los modelos teóricos, era similar a la obtenida cuando se utiliza un test basado en la TRI. Para asegurarse de que los resultados de ambos experimentos fueran comparables, se trató de diseñarlos de tal manera que ambos sistemas realizaran la evaluación sobre los mismos conceptos. Por esta razón, se optó por un dominio procedimental bien definido y acotado: la optimización lineal mediante el algoritmo Simplex y el de las dos fases (este último es una variante de la primera) (Dantzig, 1940). Para realizar el diagnóstico, se utilizaron dos herramientas diferentes basadas en la Web. Por un lado, el sistema de tests Siette, explicado en el apartado 6.3, para diagnosticar el conocimiento de los estudiantes mediante tests. Por otro lado, el tutor Simplex, un EIRP en el dominio de programación lineal explicado en el apartado 6.2.

### 7.2.1. Diseño del experimento

El experimento se llevó a cabo en noviembre de 2008, con estudiantes de Informática de la Universidad de Málaga (España). La experiencia fue incorporada como parte de una asignatura semestral que recibía el nombre de *Investigación Operativa*. Esta asignatura consiste en técnicas de programación lineal como el algoritmo Simplex y el de las dos fases. La profesora de la asignatura colaboró en la realización del artículo que surgió a partir de los resultados obtenidos en esta experiencia.

En primer lugar, los estudiantes asistieron a una clase de dos horas sobre la aplicación de los dos algoritmos mencionados. A continuación, la semana siguiente, fueron evaluados durante dos sesiones diferentes en los laboratorios de la E.T.S de Ingeniería Informática. En la primera de las sesiones, la cual tuvo una duración de media hora, se les administró a los estudiantes tres problemas en la herramienta Simplex Tutor. Inmediatamente después, los estudiantes realizaron un test de 56 ítems con Siette, el cual tenía una duración máxima de una hora. La hipótesis de la que partimos es que la evaluación en el EIRP sería equiparable a un test en el que se evaluasen los mismos conocimientos.

Los problemas que se presentaron a los estudiantes en la primera sesión se diseñaron con el objetivo de cubrir tantas restricciones del modelo de dominio como fuera

posible. Para ello, la profesora de la asignatura, como experta en el dominio, sugirió la realización de tres problemas, cada uno asociado a un tipo de solución de entre los diferentes posibles (Dantzig, 1940): uno sobre la aplicación del algoritmo Simplex con una única solución; otro que requería de aplicar el mismo algoritmo pero con una solución al problema de tipo infinita; y, finalmente, uno para la aplicación del algoritmo de dos fases. Los estudiantes podían construir la solución y comprobar su validez con el sistema antes de concluir el problema. No obstante, el Tutor Simplex fue limitado para que, en lugar de mostrar refuerzo sobre los errores cometidos por los alumnos, presentara solamente un mensaje avisando de si la solución enviada tenía errores, o no. La principal razón de no presentar información sobre las restricciones violadas, está en intentar minimizar el aprendizaje proporcionado por la herramienta a los alumnos. Este es un requisito indispensable para poder realizar una calibración de las restricciones adecuada (más detalle sobre esta necesidad se da en la sección 4.5.1).

El test de Siette estaba compuesto por 56 ítems. A la hora de definir este test, cada uno de los ítems fue diseñado teniendo en cuenta que éstos fueran apropiados para evaluar el mismo conocimiento que el asociado a las restricciones del problema. La mayoría de ellos (52 ítems) eran de opción múltiple con cuatro opciones, de las que sólo una era correcta. El resto eran ítems de respuesta libre que eran corregidas de forma automática por un mecanismo de patrones y expresiones regulares. Por cada restricción definida en el modelo de dominio de Simplex (14 en el momento de la experiencia), había 4 ítems de test evaluando el mismo conocimiento.

### 7.2.2. Análisis de los datos y resultados

Inicialmente, 23 estudiantes participaron en el experimento. Sin embargo, debido a varios problemas (por ejemplo, algunos estudiantes abandonaron la sesión antes de terminarla), los datos de siete alumnos tuvieron que ser descartados. La actividad del resto de estudiantes, se analizó a posteriori mediante los siguientes pasos: En primer lugar, se calibraron las CCI del test. Para este proceso se usaron los resultados obtenidos en el test. Concretamente, para cada estudiante, se usó un valor que indicaba si el ítem había sido respondido correctamente o incorrectamente. Esta información se utilizó para realizar el proceso de calibración con la herramienta MULTILOG. La calibración de las curvas se realizó usando el modelo 3PL. Una vez calibradas las curvas, obtuvimos los parámetros representativos, de acuerdo con el modelo 3PL usado, los cuales fueron usados posteriormente para inferir el nivel de conocimiento de los alumnos en el test.

A continuación, se realizó un proceso análogo con los datos obtenidos a partir del EIRP. En primer lugar, se calibraron las CCR del modelo del domino. Como se mencionó en la sección 4.3, en nuestro modelo, se asume que cada restricción es equivalente a un ítem en el que el resultado de violación o satisfacción equivale a responder correctamente o no el ítem subyacente. Por lo tanto, para realizar la calibración, lo que necesitábamos saber, para cada estudiante, era la lista de restricciones que violó y las que satisfizo. Una vez más, utilizamos MULTILOG para este propósito. Como resultado del proceso de calibración, se obtuvieron los parámetros que representaban las CCR. Estos parámetros fueron usados de nuevo en MULTILOG para calcular el conocimiento de los estudiantes aplicando las ecuaciones vistas en el apartado 4.5.2.

El objetivo de este experimento fue comparar el diagnóstico del conocimiento del estudiante, proporcionado por Siette, con el nivel de conocimiento del estudiante inferido por nuestro modelo. Como consecuencia, se compararon los dos valores utilizando el

test estadístico t-test por pares al 95 % de confianza, que es una técnica apropiada para casos como el presente donde el tamaño de la muestra es pequeño. Este test compara dos conjuntos por pares para determinar si se diferencian el uno del otro de manera significativa. La hipótesis nula del t-test por pares es que la media de las diferencias es igual a cero. El resultado, p-valor = 0,2091, sugiere claramente que no podemos rechazar la hipótesis de que las estimaciones de los estudiantes de conocimientos realizadas por Siette son similares a las realizadas por nuestro modelo.

## 7.3. Validez de la propuesta en entornos no acotados

El siguiente experimento fue publicado en el mismo año que el anterior, en el artículo (Gálvez et al., 2009b), en la conferencia que organiza la Asociación Española para la Inteligencia Artificial. La validez de la metodología de evaluación combinada había sido estudiada, en el experimento anterior, sobre un entorno bien definido, acotado, y con unas tareas en las que los pasos de resolución estaban bien definidos. Con el experimento que se explica en esta sección, se buscaba evaluar la idoneidad de la metodología, pero esta vez en un dominio en el que el espacio de soluciones de las actividades no está acotado y es débilmente definido.

Con el fin mencionado, para evaluar la validez de nuestras técnicas de diagnóstico en dominios con tareas débilmente definidas, llevamos a cabo un experimento con alumnos universitarios en el dominio de la POO. Para este experimento usamos la herramienta OOPS, en la versión extendida, cuya interfaz puede verse en la figura 6.2 dentro del apartado 6.1.1. El objetivo de la sesión era que los estudiantes probasen sus conocimientos de POO a través de una sesión práctica llevada a cabo en un laboratorio de docencia. Ésta fue estructurada, a grandes rasgos, en dos fases: Primeramente, se administró un test usando el sistema Siette. Posteriormente, se propusieron dos problemas a los alumnos para que los resolviesen en el EIRP OOPS.

La hipótesis de la que partimos es la misma que en el experimento anteriormente explicado en la sección 7.2, pero con la particularidad mencionada anteriormente sobre la naturaleza de las tareas del dominio. Por tanto, el objetivo de esta sesión era recopilar datos, a través de un test por una parte y de problemas de programación por otra, y usarlos para evaluar el conocimiento del alumno. Los resultados en los dos sistemas serían comparados para verificar si la inferencia del conocimiento de cada alumno, empleando las dos alternativas, proporcionaba como resultado datos similares.

### 7.3.1. Diseño del experimento

Con el fin de conseguir que los resultados de ambos sistemas fuesen comparables, el experimento se diseñó de forma que los mismos conocimientos que estaban siendo puestos a prueba en el test, fuesen presentados en los problemas prácticos a desarrollar. Para ello, en el sistema OOPS se eligieron dos problemas tipo que evaluasen los conceptos básicos de la POO, o lo que es lo mismo, dos problemas con los cuales se pudiesen evaluar las restricciones fundamentales asociadas a los principios del dominio. Usamos un subconjunto de 15 restricciones que fuese significativo en cuanto a la importancia de los conceptos contenidos, el cual fue definido por varios expertos en el dominio. En cuanto al test, el conjunto de ítems a mostrar fueron diseñados a partir del conjunto de restricciones previamente seleccionado. Cada restricción seleccionada tenía dos ítems en el test que evaluaban el mismo concepto, lo que supone un total de 30 ítems.

El experimento fue llevado a cabo en mayo del 2009 con estudiantes de Ingeniería Técnica de Telecomunicación de la Universidad de Málaga. Dichos alumnos habían recibido previamente clases en pizarra sobre los conceptos que se evaluarían en el experimento. Un total de 20 alumnos asistieron a la sesión práctica. Tras la realización del test (en el que en ningún momento se mostraban las soluciones a los ítems), los alumnos comenzaron a utilizar el sistema OOPS. Inicialmente se les propuso a todos un primer problema que les sirvió de entrenamiento en el manejo del sistema y que no se tuvo en cuenta para el análisis. Posteriormente, resolvieron dos problemas de programación a través de OOPS. Durante la utilización del sistema se tomó una fotografía de los alumnos, que se puede ver en la figura 7.3.

Figura 7.3: Alumnos de la E.T.S.I. de Telecomunicación usando OOPS.

### 7.3.2. Análisis de los datos

Una vez obtenidos los datos de ambos sistemas, incorporamos las características de la TRI a estos resultados preliminares utilizando la herramienta MULTILOG, la cual es una de las más populares para este proceso. Primeramente calibramos las CCI del test, de acuerdo al modelo 3PL de la TRI. Para ello, usamos los resultados del test en una matriz de valores booleanos, idéntica a la matriz de rendimiento que se presentó en la sección 4.5.1.2. En dicha matriz, requerida como entrada a MULTILOG, para cada estudiante e ítem, se representaba si el concepto asociado se sabía (ítem correctamente respondido) o no (respuesta incorrecta). Posteriormente, la calibración obtenida fue usada en MULTILOG en conjunción con los resultados del test para obtener la estimación del conocimiento de cada estudiante.

De forma análoga, los datos obtenidos en OOPS fueron usados para calibrar las CCR. En este caso, la matriz de rendimiento utilizada en MULTILOG representaba, para cada estudiante y restricción, si el concepto asociado se sabía (había sido satis-

fecho) o no (restricción violada) durante la resolución de los problemas. Las curvas obtenidas se utilizaron de nuevo con los datos de OOPS para generar una valoración del conocimiento del alumno.

Este proceso de evaluación fue cuidadosamente diseñado para que los resultados obtenidos fuesen homogéneos en cuanto a la fuente del conocimiento evaluada y en la naturaleza de los datos usados para estimar el conocimiento. Lo primero, mediante la ya mencionada correspondencia entre restricciones e ítems; y lo segundo, usando la matriz de rendimiento con el mismo significado de los valores: un valor booleano verdadero indicando el conocimiento del concepto relacionado y otro falso para conceptos no aplicados correctamente. Finalmente, es necesario mencionar que se utilizó MULTILOG en ambos casos, para asegurar que las técnicas de calibración e inferencia que se empleaban eran las mismas.

### 7.3.3. Resultados obtenidos

Para llevar a cabo la comparación de ambas estimaciones, realizamos una prueba $t$ de Student por pares, al 95 % de confianza, sobre las dos evaluaciones. Este estadístico es comúnmente utilizado para comparar la diferencia entre dos poblaciones de tamaño reducido. La hipótesis nula del mismo es que la diferencia de medias de las poblaciones es cero. El análisis aportó un p-valor de 0,7972, que claramente sugiere que no podemos rechazar la hipótesis nula anterior y que, por tanto, no existe una diferencia significativa entre las evaluaciones obtenidas con OOPS, que aplican nuestras técnicas de inferencia mediante la TRI y el MBR; y las obtenidas con Siette, aplicando el modelo 3PL. La importancia de estos resultados radica en que la evaluación proporcionada por las restricciones puede obtenerse más rápidamente y con menor número de restricciones que de preguntas. Esto es así porque el responder a cada ítem individualmente requiere mucho más tiempo que resolver un problema donde las restricciones están implícitas.

## 7.4. Estudio de la invariancia del modelo

Para la comprobación de la invariancia del modelo se han llevado a cabo varios experimentos sobre los diversos sistemas para los cuales disponíamos de datos experimentales. A continuación se comentan resumidamente las características más importantes de esta experimentación. Los experimentos realizados han buscado comprobar la invariancia de las estimaciones de los parámetros. Para ello se ha dividido la población de evidencias en dos grupos y se ha realizado la calibración usando distintas fuentes de evidencia. Esto se ha realizado para los sistemas OOPS, Simplex, PIPSE y para los datos disponibles de SQLTutor.

- En OOPS, se realizó una partición del grupo de 20 alumnos que proporcionaron evidencia sobre 15 restricciones en dos grupos de 10 alumnos cada uno. Esta división buscaba realizar una división homogénea de las evidencias. El coeficiente de correlación entre b y c está es de 0,95 y 0,9, respectivamente, mientras que el parámetro a es de 0,42.

- En Simplex Tutor, se dividió el grupo de 16 estudiantes en dos de 8 y 8 que proporcionaron evidencia sobre 18 restricciones. El resultado mostraba una correlación de 0,81 para el parámetro a; 0,68 para el parámetro b; y 0,91 para el parámetro c. No obstante, el resultado de este sistema es poco fiable dado que el conjunto

de estudiantes para realizar la calibración es muy reducido, proporcionando un conjunto de evidencias muy pobre para realizar este estudio.

- Para el estudio del que disponemos datos sobre el tutor PIPSE, el cual se explicará en la sección 7.5, se usaron datos de 24 estudiantes, sobre un conjunto de 6 restricciones. Aunque el modelo de dominio posee 17 restricciones, sólo 6 proporcionaban evidencia suficiente, como se verá en el estudio posterior sobre este sistema. El conjunto de 24 estudiantes fue dividido en dos grupos de doce alumnos. El resultado es una correlación alta en los tres parámetros en torno a 0,999. No obstante, pese a este resultado, el resultado no es significativo al realizarse sobre muy pocas evidencias.

- En los datos provenientes de SLQ-Tutor se utilizaron datos del año 2009, filtrando aquellas restricciones que no tenían suficiente evidencia. Este filtrado es el mismo que el realizado en el experimento 3b, el cual se explicará en el apartado 7.5. Un total de 234 restricciones formaron parte del estudio, que realizaba una división en dos grupos de 41 y 43 estudiantes, buscando tener una cantidad homogénea de evidencias. El resultado mostró un nivel alto de correlación en los parámetros de dificultad y adivinanza: 0,77 y 0,74, respectivamente; mientras que el parámetro c sólo tenía 0,3. El problema en este sistema es que, aunque se dispone de más alumnos que en los sistemas anteriores, el número de restricciones del sistema es considerablemente mayor, por lo que es necesario un grupo mucho mayor para obtener resultados significativos.

Como puede observarse, como norma general, parece existir una correlación alta entre los parámetros de dificultad y adivinanza (b y c), mientras que el parámetro de discriminación, a, suele ser más bajo. Aunque los resultados son prometedores, éstos no son significativos, puesto que parten de un volumen de datos muy reducido. Con los datos disponibles no tiene mucho sentido estudiar la invariancia a nivel de evaluación, mencionada en la sección 3.2, pues también se pueden anticipar resultados poco significativos. Por estos motivos, es necesario seguir explorando estas características una vez que se disponga de un conjunto de datos más apropiado. Entonces, si la correlación sigue siendo menor en el parámetro a, se debería estudiar la causa de esta anomalía.

## 7.5. Traza del conocimiento en MBR (calibración)

En los estudios explicados previamente, se había experimentado con grupos relativamente reducidos de estudiantes y en sesiones cortas de uso. A la hora de escalar la metodología a sistemas con una muestra de estudiantes más amplia, con un periodo de utilización mucho más prolongado, y donde los estudiantes están aprendiendo, surgen ciertas interrogantes que nos hicieron extender la metodología con la técnica de las CK-sesiones explicada en el apartado 5.2.2. En este experimento se estudiaron y compararon un subconjunto de las técnicas propuestas para agrupar las restricciones y se utilizaron para determinar cuál de las formas permite generar una mejor calibración. Además de utilizar diversas formas de agrupar, como se podrá ver, se estudian también diversos modelos paramétricos de la TRI.

Para tratar la escalabilidad y la extensibilidad en sistemas tutores con un uso elevado y donde el alumno recibe constante aprendizaje, se contó con la posibilidad de usar datos procedentes de uno de los sistemas tutores más importantes en el paradigma del

MBR: el sistema SQL-Tutor, descrito en detalle en la sección 2.3.7.2. La envergadura del sistema se puede encontrar en los dos factores mencionados anteriormente. En primer lugar, el número de estudiantes no se limita a los estudiantes de la Universidad de Canterbury, en los que SQL-Tutor es una herramienta usada habitualmente. Por el contrario, esta población está abierta al mundo a través del portal Database Place (Mitrovic, 2006), mencionado en la descripción de SQL-Tutor. En segundo lugar, al estar integrado el sistema como parte de las asignaturas de bases de datos en la universidad neozelandesa, o disponibles en todo momento a través de Internet, el periodo de uso es considerablemente mayor que el de experimentos anteriores. Además, el número de restricciones que componen el modelo de dominio, superior a las 700, es mucho mayor que el de los sistemas usados anteriormente. Por último, puesto que el sistema es un STI cuyo objetivo es proporcionar un refuerzo ante cada solución errónea en lugar de realizar evaluación, los datos incluían el aprendizaje recibido por éstos. Esto requiere del uso de una técnica como las CK-sesiones para poder aplicar los mecanismos de la TRI.

Los datos usados para este experimento provienen de ficheros de registro de actividad que SQL-Tutor genera paralelamente con las acciones realizadas en la interfaz del sistema. Para cada estudiante, se almacena un fichero en el que se van registrando las decisiones pedagógicas del sistema, las soluciones enviadas por el estudiante, las restricciones violadas, las restricciones satisfechas, el nivel de refuerzo mostrado, las peticiones de ayuda realizadas por el estudiante, así como otras medidas internas. La información sobre la violación y satisfacción de restricciones, correspondientes al modelo del alumno, también se almacenan en otro fichero para cada estudiante. Sin embargo, descartamos su utilización puesto que éstos carecían de información temporal requerida para aplicar la agrupación de restricciones en CK-sesiones. Por este motivo, la fuente de datos proviene de los ficheros de registro de actividades de cada alumno, los cuales fueron recopilados durante tres años: 2008, 2009 y 2010.

El primer paso de la técnica de las CK-sesiones modifica la forma en que se realiza la calibración, en comparación con los experimentos anteriores, mediante la utilización del umbral TCK. El estudio se centra en analizar el valor para el cual el umbral TCK puede producir una calibración más precisa de las restricciones. En este análisis, utilizamos los datos de los tres años mencionados y realizamos una calibración a posteriori, generando la matriz de rendimiento tal y como se explica en la sección 5.2.2.1. Inicialmente, los conjuntos de datos estaban compuestos por 39 estudiantes en 2008, 98 estudiantes en 2009, y 60 en 2010. Un primer filtro dejó fuera del experimento 15 alumnos del 2009 y 6 estudiantes del 2010 debido a su inactividad en el sistema. Los datos de los estudiantes que al menos realizaron un intento fueron usados para calibrar las restricciones utilizando diferentes valores del umbral TCK para generar los estudiantes virtuales. En concreto, TCK se determinó que tomara los valores 10, 5, 3 y 1 minutos. La razón principal para elegir esos valores bajos es que el aprendizaje tiene lugar cuando el estudiante está resolviendo un problema, y por lo tanto, el conocimiento no se mantiene constante por mucho tiempo.

Además, dos criterios diferentes se consideraron para construir la matriz de rendimiento: *la primera vez relevante* y el criterio de *agrupación por problemas*. El primero tiene como única evidencia el valor de la primera vez que una restricción que es relevante para un estudiante, que es el enfoque original de los experimentos anteriores. Este criterio es equivalente a establecer el umbral TCK a un valor mayor que la duración del período completo en el que se toman las evidencias. El segundo criterio consiste

en agrupar las evidencias por problemas, lo que significa que los intentos consecutivos de un estudiante dentro de un problema se consideran pertenecientes a la misma CK-sesión y, por tanto, asociados a un estudiante virtual. Aunque este último criterio tiene un valor variable de TCK, ya que entre dos problemas diferentes realizados por un estudiante no hay cantidad fija de tiempo, pensamos que sería interesante hacer esta distinción para enfatizar el cambio de conocimiento sólo entre la resolución de problemas diferentes.

### 7.5.1. Diseño del experimento

Dada la forma de las restricciones, explicada detalladamente en el apartado 2.3.2, éstas son sólo relevantes en algunos problemas. Esto supone que el número de evidencias obtenidas para una restricción dependerá del número de veces que los problemas asociados han sido intentados por los estudiantes. Por este motivo, en algunos casos, ciertas restricciones tendrán menos evidencias que otras, reflejando en menor grado el rendimiento de los estudiantes que otras. Esto puede verse acentuado en SQL-Tutor, pues el modelo de dominio tiene un número muy elevado de restricciones, llegando a haber restricciones sobre las que el número de evidencias es nulo o próximo a cero. Estas restricciones no se deberían usar en la calibración, pues producirían unos resultados menos precisos. Teniendo esto en cuenta, se programaron varios experimentos para producir conclusiones más fiables. En concreto, programamos 6 experimentos diferentes a realizar:

- Experimento 1a: Este es el experimento básico donde las restricciones que no eran relevantes durante un año determinado fueron descartadas previamente al proceso de calibración de ese año.
- experimento 1b: Sigue el mismo criterio que el experimento 1, pero en este caso, los años 2009 y 2010 fueron considerados como única fuente de evidencia (como si los dos años fueran uno sólo). Puesto que el conjunto de restricciones del modelo de dominio era el mismo en 2009 y 2010, se añadió esta variación a cada experimento con el fin de contemplar diferentes posibilidades de agrupación, para ampliar la validez de las conclusiones. El año 2008 se dejó fuera de esta consideración, ya que SQL-Tutor sufrió un cambio radical entre el 2008 y el 2009 que supuso la reestructuración y modificación de las restricciones, entre otras cosas. Por este motivo, para el año 2008 el conjunto de restricciones consideradas en este experimento es el mismo que en el experimento 1.
- Experimento 2a: Las restricciones que no habían sido relevantes más un 10 % del número máximo de veces que ésta podría haberlo sido, fueron descartadas en cada año.
- Experimento 2b: Al igual que en el experimento 1b, éste es una variación del experimento 2, pero considerando los años 2009 y 2010 como si fueran un único año. En este caso, para filtrar las restricciones de ambos años se utilizó la unión de los conjuntos de filtrado de cada año, procedentes del experimento 2.
- Experimento 3a: Se consideró también que sería interesante explorar el efecto de descartar no sólo las limitaciones con menor cantidad de evidencia, sino también aquellas con demasiada evidencia. En este experimento el filtrado de las restricciones se fijó en un 5 % como límite inferior y superior sobre la distribución del

número de veces relevantes en cada restricción. Esto quiere decir que las restricciones que no fueran relevantes un mínimo del 5 % o como máximo, un 95 %, de las veces que éstas podían ser, serían filtradas.

- 3b Experimento: Una vez más, los años 2009 y 2010 fueron considerados como una única fuente de evidencia.

Después de la combinación de las consideraciones mencionadas anteriormente, lo que supone la recopilación de datos correspondientes a 3 años; filtrando las restricciones de acuerdo a cada uno de los 6 criterios de cada experimento; y formando la matriz de rendimiento de las 6 maneras diferentes mencionadas anteriormente; el proceso de calibración se realizó utilizando la herramienta *SQL-Tutor Processor*, la cual recae sobre MULTILOG para realizar esta tarea (ver sección 6.4 para una explicación detallada). Adicionalmente, se consideraron tres modelos diferentes para la calibración de las curvas características de las restricciones, correspondientes a los modelos logísticos 1PL, 2PL y 3PL. En total, son 324 conjuntos sobre los que realizar la calibración. Lógicamente, de los dos modos de operación descritos sobre *SQL-Tutor Processor*, la interfaz gráfica no se pudo usar al sólo dar soporte para una calibración por ejecución, lo que supondría un coste temporal enorme, teniendo en cuenta que serían 324 ejecuciones con sus correspondientes ajustes de los parámetros. En su lugar, se utilizó el modo programático en el que, mediante un script, se realizó el proceso de calibración de cada elemento del conjunto, en un tiempo razonablemente reducido (inferior a 10 minutos de ejecución).

Con el fin de evaluar la calidad de la calibración realizada sobre cada conjunto de datos, se tomó uno de los valores de salida que MULTILOG proporciona como medida de la bondad del ajuste de los parámetros asociados a las CCR. Este valor, cuyo nombre original en inglés es *negative-twice-the-loglikelihood*), puede traducirse como *menos dos veces el logaritmo de la función verosimilitud*. El valor es el proporcional a la función de verosimilitud, y cuanto más pequeño este valor, mejor es el ajuste del conjunto de datos (Hambleton et al., 1991).

### 7.5.2. Resultados

En cuanto a los tres modelos de calibración, se concluyó que el modelo 1PL produjo una calibración con una calidad significativamente menos importante que las otras dos (t-test por pares con un p-valor $<1{,}767 \times 10^{-14}$ al comparar 1PL con el modelo 2PL y p-valor $= 1{,}852 \times 10^{-12}$ en la comparación de 1PL con 3PL). Sin embargo, no pudimos encontrar ninguna diferencia significativa entre los modelos 2PL y 3PL (p-valor = 0,1839) y lo que es más: a veces, siguiendo un patrón aleatorio, el modelo 3PL fue mejor y otras veces 2PL. Esto sugiere que para la calibración de las restricciones, un modelo 3PL o 2PL funcionan de manera similar, pero 1PL no sería apropiado.

Si tenemos en cuenta sólo los resultados de cada experimento, que se resumen en la tabla 7.2, podemos observar que los experimentos de tipo b, es decir, aquellos que están considerando 2009 y 2010 como una única fuente de evidencia, producen significativamente mejores resultados (p-valores de 0,002411; $6{,}037 \times 10^{-9}$; y 0,0002249, para las comparaciones de los experimentos 1a con 1b; 2a con 2b; y 3a con 3b, respectivamente). Esto podría explicarse por el hecho de que se añadieron nuevos problemas al modelo de dominio de SQL-Tutor entre 2009 y 2010, lo que haría algunas restricciones particulares fueran relevantes más veces que en el año anterior. Por otra parte, la situación

contraria ocurrió también. Es decir, algunos problemas se suprimieron, haciendo que las restricciones asociadas fueran relevantes menos veces a nivel general. El filtrado de esas restricciones en nuestros estudios puede haber contribuido a una exploración más precisa de la mejor manera de construir la matriz de rendimiento, ya que sólo tienen en cuenta las restricciones que sean relevantes en ambos años.

| Año / Experimento | 2008 | 2009 | 2010 |
|---|---|---|---|
| Experimento 1a | 5300,38 | 8105,63 | 3768,99 |
| Experimento 1b | 8079,88 | 3766,21 | - |
| Experimento 2a | 5280,26 | 7950,18 | 3635,91 |
| Experimento 2b | 7614,62 | 3564,85 | - |
| Experimento 3a | 395,94 | 5884,48 | 2604,74 |
| Experimento 3b | 5558,42 | 2561,12 | - |

Tabla 7.2: Calidad media de la calibración por experimento / año (la unidad de medida es menos dos veces el logaritmo de la función verosimilitud).

Al comparar los diferentes experimentos, el último da un mejor resultado, lo que sugiere que el filtrado de algunas restricciones, que normalmente son relevante un alto número de veces, es también un buen criterio. Este problema es especialmente notable en el conjunto de datos de 2008, donde algunas de las restricciones eran siempre relevantes, haciéndolas inadecuadas para la calibración (alguna evidencia correcta o incorrecta debe ocurrir para producir un resultado de la calibración adecuada). El filtrado de esas restricciones mejoró drásticamente la calidad de la calibración.

Se puede pensar que la calidad de los conjuntos de datos resultantes en cada experimento está relacionada con el número de restricciones que intervienen en ella. Siguiendo esta idea, la calidad de las restricciones debe ser mayor en conjuntos de datos grandes puesto que el error del ajuste sería menor al tener más evidencias. Sin embargo, como podemos ver en la tabla 7.3 esto no es cierto: el experimento 3 tiene menor restricciones que los experimentos 2 y 2b, pero la calidad es mayor (véase la tabla 7.2), lo que sugiere que los criterios de filtrado realmente eliminan del estudio las restricciones que no están proporcionando información importante.

| Año / Experimento | 2008 | 2009 | 2010 |
|---|---|---|---|
| Experimento 1a | 493 | 502 | 480 |
| Experimento 1b | 493 | 468 | 468 |
| Experimento 2a | 429 | 386 | 357 |
| Experimento 2b | 429 | 346 | 346 |
| Experimento 3a | 300 | 478 | 346 |
| Experimento 3b | 300 | 331 | 331 |

Tabla 7.3: Número de restricciones involucradas en cada experimento / año.

En cuanto a la mejor manera de construir la matriz de rendimiento, que era el objetivo principal de esta experimentación, encontramos resultados muy interesantes. En cuanto a los valores medios de todos los experimentos, que se resumen en la figura 7.4, podemos observar que cuanto mayor sea el valor de TCK, mejor será la calidad

de calibración. Esto también es consistente con el hecho de que el método de agrupación por *primera vez relevante* se comporta mejor que cualquier otro valor del umbral TCK (diferencias significativas comprobada con un t-test por pares con el resultado de p-valor $<2{,}2 \times 10^{-16}$ para todas las comparaciones). No obstante, el resultado más sorprendente encontrado es que uno de los métodos destacaba sobremanera en todos los experimentos, incluso con un conjuntos de restricciones diferentes en cada uno: la *agrupación por problemas* (diferencia significativa con un p-valor = $1{,}152 \times 10^{-15}$ cuando se compara con la TCK = 1; p-valor = $4{,}057 \times 10^{-16}$ en la comparación con TCK = 3; p-valor $<2{,}2 \times 10^{-16}$ para el resto de los métodos con TCK; y p-valor = $3{,}924 \times 10^{-7}$ para la comparación con el método de agrupación por *primera vez relevante*).

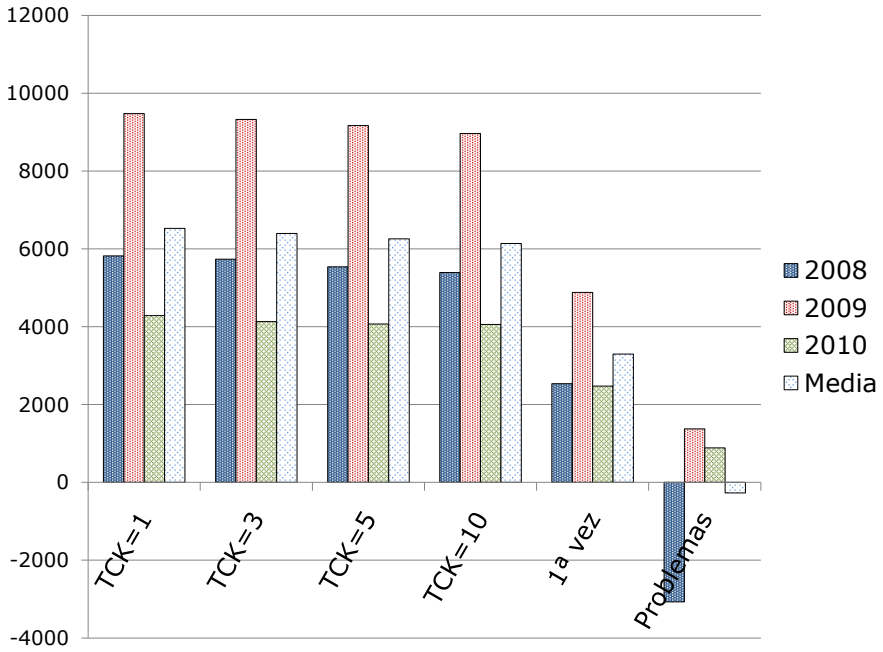



Figura 7.4: Gráfica con el resultado de la calidad de las restricciones para diferentes modelos (la unidad de medida es menos dos veces el logaritmo de la función verosimilitud).

Estos resultados podrían ser explicados por el hecho de que el método de agrupación de restricciones *por problemas* produce un número mayor de estudiantes virtuales. Esto implica que nuestro método de las CK-sesiones no es apropiado para la calibración, independientemente del valor del umbral TCK. En cambio, el enfoque original de *la primera vez que es relevante* es una mejor opción. La idea de la CK-sesión es una técnica de granularidad gruesa para ser utilizada en la calibración del CBM + TRI y, por tanto, un método de granularidad fina, como la agrupación de las pruebas por problemas produce una mejor calidad de calibración.

## 7.6. Traza del conocimiento en MBR (Evaluación)

Al igual que se usó la traza del conocimiento para la calibración, se realizó un análisis de los datos obtenidos del sistema SQL-Tutor para la evaluación del conocimiento. En este estudio contábamos con una sesión particular en la que se les administró un pre-test

a los estudiantes, seguidamente utilizaron el sistema y, por último, realizaron un post-test. Sin embargo los resultados encontrados son totalmente dispares y con resultados poco significativos. La razón de esta disparidad se encuentra en varios aspectos que como se verá no cabe esperar unos resultados mejores que los encontrados. Los aspectos que motivan esta situación son: en primer lugar la sesión sobre la que se realizó el estudio fue muy corta (2 horas de uso del sistema), lo cual puede no ser suficiente para recopilar evidencia suficiente. En segundo lugar, el pre-test y el post-test no estaban diseñados para tratar los mismos conceptos que las restricciones, sino que trataban el tema en general, por lo que es muy probable que el resultado no fuese el mismo. En tercer lugar, no contamos con los ítems del test y el resultado de cada estudiante en ellos, lo cual imposibilita la calibración de los mismos y la emisión de un juicio usando la TRI. De esta forma tenemos que usar para comparar la evaluación del sistema, emitida mediante la TRI, con criterios de la TCT como el porcentual o la calificación por puntos. Por este motivo, los resultados de la comparación, aunque sean normalizados en la misma escala, son muy proclives a ser diferentes.

En la actualidad estamos a la espera de nuevos datos que permitan realizar un estudio más controlado y orientado a probar la mejora en el rendimiento del sistema en relación con el aprendizaje del alumno. Los datos se obtendrán también mediante SQL-Tutor en un futuro próximo y buscarán proporcionar la información que en los experimentos realizados ha sido una muy probable fuente de error.

## 7.7. Medición de la calidad de las restricciones para evaluación

El último experimento publicado en un artículo es (Gálvez et al., 2012), el cual consistió en un estudio en el que se ponían en práctica diversas herramientas elaboradas. Por un lado se utilizó CBMEngine, que agrupaba los mecanismos de evaluación desarrollados anteriormente como parte de su funcionalidad (ver sección 6.5). Por otro lado, se utilizó PIPSE, que se asienta sobre CBMEngine para proporcionar la evaluación de los estudiantes (ver apartado 6.5.4.1). Sin embargo, el uso de las herramientas no era el principal objetivo del experimento, sino que era un medio para estudiar la factibilidad y efectividad de un mecanismo que determinara la calidad de las restricciones como instrumento de medida para la evaluación.

El mecanismo mencionado, el cual se explica detalladamente en la sección 5.4, está motivado por la problemática que surge en dos aspectos que pueden hacer que las restricciones no sean adecuadas de cara a usarse para evaluar el conocimiento del alumno. El primer aspecto se encuentra en el número de evidencias que se tienen para una determinada restricción, el cual, si es muy reducido puede hacer que la restricción no sea adecuada para evaluar. Este aspecto era originalmente el eje central del estudio que se explica en esta sección. Sin embargo, con los resultados obtenidos, se descubrió también el segundo de los aspectos: la posibilidad de que las restricciones no codificaran adecuadamente el principio para el que éstas están diseñadas, o si están codificadas adecuadamente, puede que el nivel de generalidad no sea adecuado.

La hipótesis principal consistía en que la FII de la TRI podría ser aplicada a las restricciones, de la misma manera que se utiliza normalmente en entornos de tests, para detectar las restricciones no adecuadas para evaluación. Con el experimento realizado, además de probar la validez de esta hipótesis, nos planteamos una meta secundaria.

Puesto que el experimento hacía uso del sistema PIPSE, el cual se asienta en la evaluación mediante los modelos propuestos en esta tesis, queríamos verificar que los resultados de la evaluación eran consistentes con los de otros experimentos realizados. En este sentido se trató de estudiar si la evaluación proporcionada era similar a la obtenida a partir de un test que evaluaba los mismos conocimientos.

### 7.7.1. Diseño del experimento

A fin de evaluar el método mencionado, se diseñó un experimento que iba a desarrollarse con estudiantes de último curso de la titulación de Ingeniería Informática de la Universidad de Málaga. Un total de 24 estudiantes participaron en el estudio que se realizó en diciembre de 2011 y que constó de varias etapas. En primer lugar, los estudiantes recibieron varias clases sobre los diferentes índices que permiten resolver los problemas de Análisis de Inversiones. A continuación, durante una sesión de una hora de duración se les dejó utilizar el sistema PIPSE para resolver dos problemas vistos previamente en clase. Una semana más tarde se realizó un examen basado en papel en el que se les propusieron dos problemas y se les planteó un test.

Para probar la hipótesis del experimento, los problemas propuestos en el examen no abarcaban todo el conjunto de restricciones, una característica que se utilizará posteriormente en el análisis de la calidad de las restricciones con la FIR. En cuanto al test, fue diseñado siguiendo las mismas pautas que en estudios anteriores (secciones 7.3 y 7.2), con el fin de evaluar los mismos conceptos asociados a las restricciones relevantes en los problemas planteados. Para este fin, por cada restricción, se escribió un ítem, produciendo un total de 15 ítems en el examen. Dos de las restricciones se quedaron fuera del test, ya que no estaban asociadas a principios concretos, sino a verificaciones matemáticas sobre la solución.

A diferencia de los primeros trabajos con esta técnica, el examen se realizó en papel con el objetivo de obtener solamente la violación de las restricciones y prevenir que los estudiantes recibieran cualquier tipo de refuerzo. El examen debía ser resuelto proporcionando una solución de la misma forma que se realiza con la interfaz del tutor PIPSE. Con esta omisión de información acerca de los errores cometidos en la solución, el factor de aprendizaje asociado al refuerzo fue aislado y dejado fuera del experimento, hecho que, de acuerdo a los requisitos de TRI, es importante para generar una buena calibración de las restricciones y para aplicar los mecanismos de la TRI. Una vez que todos los estudiantes terminaron el examen, las soluciones proporcionadas fueron introducidas en el EIRP y se comprobaron las restricciones violadas y satisfechas.

La nota de los alumnos en el examen a papel figuraba como parte de la nota final de la asignatura, por lo que los 24 alumnos matriculados en la asignatura participaron en él. Además, el test de Siette también se administró a los estudiantes después en la misma sesión evaluativa. Después de que todos los datos de los estudiantes hubieran sido recopilados, se realizó el análisis de las restricciones aplicando el mecanismo de la FIR, explicado en el apartado 5.4. En el resto del análisis usaremos un valor numérico para caracterizar la FIR de una restricción. Sin embargo, esto no es totalmente correcto ya que la FIR es una función de densidad y el valor que se usará es simplemente el área de la curva. No obstante, por simplicidad, asumimos que usar el término FIR asociado a un valor numérico se refiere al área mencionada. Fruto de este análisis se filtraron algunas de las restricciones y se utilizaron las restantes para aplicar la evaluación sumativa de los estudiantes, lo que nos lleva a los resultados descritos en la sección siguiente.

### 7.7.2. Resultados

La solución proporcionada por cada estudiante en el papel se introdujo en el sistema PIPSE, que envió cada evidencia a la herramienta CBMEngine, registrando todos los datos y realizando la calibración de las restricciones usando el modelo logístico 3PL. La salida de calibración, es decir, los parámetros que representan la CCR, fueron usados para determinar la función de información de cada restricción, aplicando la fórmula de la ecuación 3.9 sobre los parámetros de la restricción. Como resultado, se obtuvo un valor promedio de 14,81 de la FIR y una desviación típica de 2,18 para el conjunto de las 17 restricciones.

Antes de examinar los resultados, se agruparon las restricciones en aquellas que no eran relevantes en los problemas realizados en el examen y las que sí. En cuanto a esta agrupación, el primer hallazgo que apoyaba nuestra hipótesis fue que el grupo de las restricciones relevantes, compuesta por siete de ellas, tenían una FIR media mayor respecto al grupo de no relevantes (16,29 en comparación con 13,76). Aunque después de aplicar un t-test al 95 % de confianza no pudimos encontrar diferencias significativas en sus medias (p-valor de 0,68), descubrimos que una de las restricciones del análisis tenía un valor extraño que estaba afectando a los resultados mediante la introducción de ruido. Cuando descartamos esta restricción, la diferencia fue significativa (p-valor de 0,012). La restricción a la que nos referimos tenía un valor muy superior al resto y su explicación se comenta a continuación.

Para analizar la validez de la hipótesis principal, se ordenaron las restricciones de mayor a menor, de acuerdo al valor obtenido en su función de información. El resultado fue que 5 de las 7 restricciones que habían sido relevantes estaban en la parte superior de la lista. En este caso particular, la división de los datos usando como corte el valor $\bar{x} + 0{,}5\sigma$, se tradujo en la división de las restricciones relevantes de las que no lo eran. Esto sugiere que la mayoría de ellas podrían ser detectadas mediante la FIR (de acuerdo a la segunda fuente de error de las posibles explicadas en la sección 5.4.1).

En cuanto a las otras dos restricciones relevantes que no se encontraban en la parte superior, ambas estaban en la parte inferior por debajo de la media con un orden de 1,67 veces la desviación típica, lo cual era una diferencia significativa. De las dos restricciones, la que tenía una menor FIR se observó que estaba representando un principio del dominio que estaba implícito en otras restricciones y, además éste estaba incorrectamente codificado, por lo que, la restricción no estaba proporcionando mucha información sobre el conocimiento del estudiante. Para la otra restricción, en la parte inferior de la lista, no se encontró diferencia significativa respecto del resto y los expertos en el dominio no encontraron ninguna otra restricción con la que ésta pudiera fusionarse. Esto probablemente se explica por el tamaño tan reducido de la población de estudiantes, el cual no proporcionó evidencia suficiente para obtener una buena calibración de la restricción. En cualquier caso, las irregularidades de ambas restricciones fueron detectadas con esta herramienta (conforme a los errores que motivan la primera fuente de error, explicadas en la sección 5.4.1).

Adicionalmente, durante el análisis se encontró una restricción con un valor extremadamente destacado de su función de información sobre el resto. Tenía un valor de 20,07, que la sitúa por encima de la media en un orden de 2,4 veces la desviación típica. Puesto que esta restricción no se había diseñado a propósito para que fuese diferente de las demás, se examinó la causa de este valor desproporcionado. Nos dimos cuenta de que esto era debido a la agrupación de varios conceptos juntos, lo que llevó a que

los errores de los estudiantes fueran mucho más pronunciados aquí. Esto significa que el mecanismo fue capaz de detectar una restricción candidata a ser dividida en otras asociadas a principios más específicos del dominio (según otro de los posibles errores que se pueden cometer en la etapa de codificación de las restricciones).

A continuación, El conjunto de las restricciones filtradas fue utilizado en la herramienta CBMEngine para obtener la evaluación de cada estudiante. Esta evaluación se comparó con la obtenida en el test de Siette, usando un t-test por pares al 95 % de confianza. Como resultado de la prueba se obtuvo un p-valor de 0,8155. Esto indica claramente que en el caso de pares de puntuaciones pertenecientes a un estudiante, no hay ninguna diferencia significativa entre ellas. Además, se realizó un análisis de correlación entre ambos resultados, obteniendo un coeficiente de correlación de 0,06. Este es un valor muy pequeño que creemos podría ser explicado por dos factores: a) el número de datos de los estudiantes y restricciones no es lo suficientemente grande, o b) los ítems de la prueba no fueron diseñados para evaluar correctamente los mismos conceptos que el examen práctico.

## 7.8. Conclusiones del capítulo

Siguiendo con el enfoque desde lo más específico a lo más genérico que se planteó originalmente como metodología de trabajo, los experimentos iniciales han estudiado aspectos específicos de la metodología propuesta y han intentado ir abarcando aspectos más generales. El primer experimento realiza un estudio de viabilidad y eficiencia del MBR como técnica de modelado del alumno y herramienta educativa. La experimentación realizada compara en un grupo experimental el uso de un sistema MBR de POO con un grupo de control que no lo utiliza. Los resultados sugieren que la técnica mejorar el aprendizaje considerablemente, llegando a pasar del 48 % de los aprobados al 81 %.

Los experimentos posteriores extienden el estudio mediante la evaluación sumativa en dominios de diferente naturaleza. En dominios simples, como el que sirve de base para Simplex Tutor, así como también en dominios complejos, como el de OOPS, la aplicabilidad de la metodología se puede ver claramente exitosa, pues los sistemas pueden utilizarla como parte de su funcionamiento normal. La experiencia adquirida con los sistemas desarrollados y su experimentación sirvió para revisar y refinar los modelos, que fueron extendidos en el marco de trabajo CBMEngine, donde la aplicabilidad es incluso más clara. Dada la naturaleza de las restricciones, la evaluación que se realiza es más rápida ya que éstas no se preguntan directamente y una por una, sino que están implícitas en la resolución de los problemas.

Al comparar la evaluación entre los sistemas estudiados con la evaluación formal proporcionada por un test evaluando conceptos asociados a las restricciones, no hay indicios de una diferencia significativa. No obstante, el no encontrar diferencia no es tampoco indicativo de que son iguales. Para ello, se estudió la correlación entre los sistemas, la cual no era excesivamente grande. Esto no quiere decir que los sistemas no estén evaluando correctamente. Lo que creemos que puede estar pasando es una combinación de factores que influye en estos resultados. Primeramente, los estudios realizados cuentan con muy pocos alumnos, lo que puede hacer que la evaluación se vea afectada fácilmente por elementos externos que introduzcan ruido en los resultados. En segundo lugar, un test requiere una respuesta declarativa, sobre conceptos teóricos mayormente, mientras que en los sistemas, el proceso de resolución es de naturaleza

procedimental. De esta forma, mientras que en un test se pregunta explícitamente el concepto, en un EIRP el concepto es implícito en la resolución y es el conocimiento procedimental el que actúa, lo cual podría ser un factor influyente.

En la experimentación realizada sobre la evaluación formativa, se ha estudiado el uso de la agrupación en CK-sesiones para la calibración, probándose varias formas de agrupación de las sesiones. Los resultados muestran que la agrupación por umbral TCK no es una buena opción. En su lugar, la más beneficiosa es la agrupación de restricciones por problemas, la cual destacó sobre el resto de métodos estudiados. No obstante, sería necesario estudiar todavía otras formas de agrupación propuestas en la sección 5.2.2.3.

Respecto a la invariancia del modelo se han realizado varios estudios. Los resultados, aunque son prometedores, no son significativos ni concluyentes debido a la evidencia disponible. En este sentido, la evidencia obtenida a partir de estudiantes reales no es adecuada, bien porque el número de alumnos es muy reducido o porque las evidencias recopiladas no son suficientes. Por este motivo es necesario ampliar este estudio cuando se disponga del conjunto de datos adecuado.

En relación con el estudio realizado sobre la función de información como herramienta para la determinación de la calidad de las restricciones, el objetivo era usar un mecanismo automático para la filtración de restricciones que no hubiesen sido relevantes o que tuvieran pocas evidencias. No obstante, además de verse su utilidad en este sentido, se descubrieron otros usos de la misma para estudiar la calidad del modelo de dominio de un sistema MBR. Así pues, el estudio, aunque con tan sólo 7 restricciones, tiene datos muy importantes que reflejan la utilidad para detectar restricciones que están reflejando principios incorrectos del dominio.

Pese a que se ha probado la aplicabilidad y viabilidad de los mecanismos de evaluación, muchas otras características han quedado pendientes, principalmente motivado por la falta de estudiantes reales sobre los que poder experimentar. Aunque la validez estudiada se ha centrado en un tipo concreto de la rama de la Psicometría, existen muchos otros tipos de validez que podrían ser estudiados (Moss et al., 2006); otras formas de comprobar la fiabilidad (Cook y Beckman, 2006); y diversos índices para estudiar la precisión y la consistencia proporcionada por la TRI (Wyse y Hao, 2012). El objetivo del estudio empírico realizado en esta tesis ha buscado intentar abarcar tanto la evaluación sumativa como la sumativa, lo que ha provocado que no se haya profundizado en ninguna de estas ramas. Dado que esto es interesante desde el punto de vista psicométrico para estudiar las propiedades de la metodología de evaluación, se plantean como líneas abiertas de investigación.

# Parte VI

# Conclusiones

Esta última parte contiene las conclusiones del trabajo de tesis, las aportaciones, limitaciones, y las líneas de investigación abiertas.

# Capítulo 8

# Conclusiones

*Lo que sabemos es una gota de agua;*
*lo que ignoramos es el océano*

Isaac Newton (1642 - 1727)

**Resumen:** En este capítulo se presentan las conclusiones del trabajo de tesis. Aquí se incluyen las aportaciones de la investigación realizada, se mencionan las limitaciones existentes, y se detallan las líneas de investigación que quedan abiertas.

El doctor Stellan Ohlsson, cuyos trabajos han servido de referencia en la investigación realizada en esta tesis, expuso en su discurso como conferenciante invitado, en la *15th International Conference on Artificial Intelligence in Education* (Auckland, Nueva Zelanda, junio-julio del 2011), una analogía entre la investigación actual en el campo de la IA en la Educación y la realizada en el pasado en el campo de la Aeronáutica. La analogía se presenta en los esfuerzos que se realizaron en el pasado para conseguir alcanzar la meta de volar. Muchos fueron los aviones construidos, y muchos los intentos de mantenerse en el aire. No obstante, hasta la puesta en común de diversos principios físicos y de la aerodinámica, no se consiguió superar esta barrera. De la misma forma, las investigaciones actuales en la educación artificial, tienen como fruto muchos aviones, o sistemas tutores, que buscan la mejora del aprendizaje centrándose normalmente en un principio concreto. Al igual que pasó con la Aeronáutica, el futuro de los sistemas tutores pasaría por aunar principios de aprendizaje en un mismo sistema, maximizando la efectividad como herramienta educativa.

Una idea parecida defendió otro de los autores más importantes para esta tesis, la doctora Antonija Mitrovic, en la *11th International Conference on Intelligent Tutoring Systems* (Chania, Grecia, Junio del 2012). En una charla especial sobre el futuro de los STI, en la que participaron los mayores exponentes de la comunidad científica en este campo, Mitrovic asemejaba la investigación actual con islas de conocimiento. Cada isla se corresponde a los diferentes campos de investigación que cada miembro de la comunidad científica realiza. Así, cada campo, normalmente está aislado en el sentido de que los estudios se realizan sobre una materia específica, sin tener en cuenta lo que se realiza en otras islas. Para el futuro, Mitrovic expuso la necesidad de aunar esfuerzos y combinar principios existentes en cada campo con el fin de realizar una mejora significativa en los STI.

La investigación realizada sigue la filosofía de las dos ideas anteriores, puesto que ha acercado dos campos que hasta ahora estaban aislados entre ellos: una gran isla representada por el paradigma MBR, y otra representada por las técnicas de evaluación formales de la TRI. Las dos técnicas mencionadas han sido combinadas en un modelo que intenta resolver un problema doble:

- Primero, las metodologías formales de evaluación como la TRI, las cuales se aplican mediante tests, tienen el problema de que no están en principio diseñadas para evaluar el conocimiento en tareas donde hay un proceso de resolución complejo. Es decir, en dominios procedimentales. Las interacciones existentes en esta línea están limitadas, por la forma de los ítems, a tareas simples como ordenación o emparejamiento de elementos, por lo que es necesario usar EIRP que permitan al alumno llevar a cabo la interacción compleja requerida.
- De los paradigmas existentes sobre EIRP no existen mecanismos que permitan determinar de manera formal el conocimiento del alumno. Entre éstos, el MBR se presenta como una de las alternativas más fáciles y eficientes. Sin embargo, este paradigma carece de un mecanismo de evaluación bien fundamentado, ya que las estimaciones sobre lo que el alumno sabe se basan en heurísticos.

La propuesta presentada busca resolver los dos problemas anteriores combinando los dos paradigmas de forma que se aprovechen las características complementarias para formar un modelo beneficioso para ambos campos. Así pues, la evaluación formal busca ser extendida para aplicarse a dominios procedimentales donde la complejidad de las tareas no esté limitada por la forma de los ítems. Paralelamente, los EIRP en general y el MBR en particular se ven enriquecidos con una nueva forma de diagnosticar y de modelar al alumno.

El contenido de este capítulo realiza una recopilación del trabajo presentado de la siguiente forma: en la sección siguiente se resumen las contribuciones de esta tesis a las diferentes disciplinas sobre las que se enmarca. Seguidamente, en la sección 8.2, se mencionan las limitaciones identificadas en el modelo propuesto. Finalmente, se exponen las líneas de investigación que se abren con este trabajo, que a su vez suponen el plan de trabajo futuro.

## 8.1. Aportaciones

La investigación desarrollada en esta tesis contribuye al desarrollo del campo de la IA en la educación y de la evaluación educativa de acuerdo a cinco grandes grupos que se desglosan seguidamente:

1. *Aportaciones a la evaluación formal*: Se ha definido una metodología para realizar la evaluación en dominios de tipo procedimental a partir de las evidencias recopiladas de la interacción con el alumno. Sus características más importantes son las siguientes:
    - La metodología es sistemática y bien fundamentada. Esto es así puesto que la evaluación que se realiza sobre el conocimiento del alumno lo proporciona un método probabilístico y estadístico. El uso de los mecanismos de evaluación de esta técnica, los cuales destacan por su objetividad y formalidad, hacen que la emisión de un juicio sea, a su vez, de este tipo.

- Es un modelo de evaluación cuantitativo, que a diferencia de los cualitativos, no tiene en cuenta los elementos subjetivos que rodean el proceso de evaluación para emitir el juicio del alumno. De este modo se persigue la objetividad del proceso.
- La metodología de evaluación es sumativa por naturaleza, al proporcionar un juicio sobre el conocimiento del alumno. Ésta se presenta como el elemento básico previo de cara a proporcionar una evaluación formativa del alumno, la cual se contempla en el siguiente grupo de aportaciones.
- Aunque la base de la metodología es el uso de la evidencia para realizar la evaluación de forma general, el modelo se ha particularizado desarrollando un modelo de respuesta basado en la TRI que se aplica a EIRP mediante el MBR. Usando una analogía identificada entre los ítems de los sistemas de tests y las restricciones del mencionado paradigma, se ha formalizado un modelo que utiliza las restricciones como instrumento de medida. Los mecanismos de la TRI pueden ser aplicados sobre las restricciones como si éstas fueran ítems para estimar el conocimiento del alumno usando las evidencias recopiladas en dominios procedimentales.
- La metodología planteada intenta mantener la genericidad en la medida de lo posible. Con ese objetivo, el modelo de respuesta es independiente del modelo de la TRI usado y se centra en el uso de las CCR, análogas a las CCI en sistemas de tests. El modelo propuesto es independiente del dominio donde se aplica, característica que es heredada del MBR. También, el modelo es generalizable a otros tipos de EIRP en los que las curvas características puedan usarse sobre evidencias. Para ello las fuentes de evidencia deben cumplir con los supuestos necesarios por la TRI.
- El uso de EIRP, al ser un tipo de STI, tiene como objetivo principal el mejorar el proceso de aprendizaje del alumno. Esto entra en conflicto con algunos de los supuestos que permiten aplicar los mecanismos de la TRI. Para solventar esto, como parte de la metodología, se proponen dos *métodos de recolección de evidencias* que descartan el efecto del aprendizaje en estos sistemas, permitiendo la aplicación de la TRI.
- El modelo de evaluación puede ser aplicado adaptativamente en EIRP de forma parecida a como se realiza en los TAI. Sin embargo, en lugar de ítems se seleccionan adaptativamente problemas en base al conocimiento del alumno. Para ello, se ha definido un nuevo tipo de ítems en sistemas de tests que se han denominado *ítems compuestos*. Éstos son similares a los testlets, puesto que agrupan evidencias, pero están diseñados para modelar fuentes de evidencias más complejas. Por ejemplo, los ítems compuestos permiten modelar los problemas del MBR y sus preguntas componente las restricciones. Como cualquier otro ítem, éstos tienen una curva característica, que se ha denominado CCIC, que se obtiene a partir de las CCI de los ítems componente. En base a las CCIC se han formalizado los mecanismos básicos de selección adaptativa de la TRI para seleccionar ítems compuestos (sección 5.1.2).
- La forma de realizar la evaluación es mucho más corta que en sistemas de tests. Esto es así puesto que en la resolución de un problema, los conceptos involucrados no se preguntan tal cual, sino que son inherentes al proceso de resolución. De esta forma, la evidencia puede obtenerse en un menor

tiempo y mediante pocos problemas en lugar de usar un test compuesto por multitud de ítems, correspondientes a cada concepto involucrado.

La utilización de EIRP como base de la metodología propuesta permite cubrir la principal limitación de los mecanismos formales de evaluación en sistemas de tests. En este sentido, la novedad de la metodología es que extiende la evaluación del conocimiento en los sistemas anteriores a tareas complejas en las que el alumno debe realizar un proceso de resolución. La naturaleza de las tareas sobre las que se puede aplicar dista mucho de las típicas preguntas de respuesta simple o de opción múltiple y va mucho más allá de las interacciones más complejas existentes hasta la fecha en los sistemas de test que tratan tareas sencillas como la ordenación de elementos.

2. *Aportaciones al diagnóstico del alumno en EIRP*: La metodología anterior además de extender el ámbito de aplicación en los sistemas de tests, permite determinar el conocimiento del alumno en sistemas MBR y, de forma general, en EIRP. Las características particulares en relación con estos sistemas son:

   - La estimación del conocimiento del alumno se realiza usando una metodología bien fundamentada. Esto supone un avance en el modelado del alumno dentro del paradigma MBR pues, hasta ahora, la mayoría de los mecanismos existentes para esta tarea se basan de heurísticos. Para ello se propone una extensión de la estructura básica del MBR en: el modelo de dominio, con las curvas de las restricciones y de los problemas; el modelo del estudiante, con la estimación del conocimiento bien fundamentada; y el módulo pedagógico, con la lógica necesaria para manejar los modelos anteriores.
   - Aunque existen algunos mecanismos bien fundamentados para el modelado del alumno en el MBR que usan redes bayesianas, éstos tienen la limitación de que sólo son aplicables a sistemas con un modelo de dominio reducido. La metodología de evaluación propuesta, sin embargo, es aplicable a sistemas con un número elevado de restricciones. Como ejemplo de la escalabilidad del sistema, en las pruebas realizadas con uno de los sistemas más extensos e importantes, como es el SQL-Tutor, con un modelo de dominio de más de 700 restricciones, no se detectó efecto en la eficiencia, o si este existía fue insignificante.
   - A diferencia de cualquier método existente en el MBR, y de la mayoría de técnicas existentes dentro del campo de los EIRP, la propuesta de esta tesis se basa en la inferencia del conocimiento del alumno en lugar de la estimación del aprendizaje. Esta característica le confiere la capacidad de usarse como herramienta de evaluación, que otras no poseen. Más que una ventaja sobre el resto de técnicas, esto es un valor añadido, puesto que la razón de ser de los EIRP es favorecer el aprendizaje, no realizar evaluación.
   - Como característica inherente al uso del MBR, la metodología permite aplicarse a dominios procedimentales en los que las tareas son de naturaleza débilmente definidas. Esto es, aquellas donde, para resolver un problema, existen infinidad de caminos que llevan a la solución correcta. En el otro paradigma importante dentro de los STI, los tutores cognitivos, el modelado del dominio se torna prácticamente imposible en estos dominios al requerir modelar cada paso que se puede realizar para resolver un problema.

3. *Aportaciones a la instrucción del alumno en el MBR*: La metodología de evaluación sumativa anterior se ha extendido para definir un modelo de evaluación formativa del alumno en sistemas MBR. De esta forma, la metodología instructiva se caracteriza por:

   - De acuerdo con el trabajo de los autores más importantes en la Psicometría (ver sección 1.1.1), la evaluación formativa se ha planteado como una evaluación sumativa más un refuerzo que guía al alumno en la instrucción de cara a evaluar unos objetivos de aprendizaje que serán evaluados posteriormente en una evaluación sumativa final. Con esta idea, e inspirado en los test relativos al criterio, se extiende el modelo de dominio del MBR usando una agrupación conceptual sobre la que se pueden establecer objetivos de aprendizaje que serán evaluados en una evaluación sumativa final. De la misma forma, se propone la extensión del MBR con un modelo del alumno que refleja lo que el alumno sabe acerca de la abstracción anterior. Esta agrupación se propone tanto sobre problemas, agrupándolos en tipos o categorías; como en restricciones, agrupándolas en conceptos del dominio.
   - La metodología sumativa, sin más, sólo permite la evaluación en momentos concretos o considerando elementos en los que todavía no ha ocurrido aprendizaje. Esto deja fuera información importante sobre la evolución del conocimiento que puede usar el sistema para dirigir la instrucción. Con el objetivo de paliar este problema se extiende el modelo de evaluación sumativo mediante un concepto nuevo denominado *CK-sesión*. Éste consiste en agrupar evidencias de un alumno en periodos de tiempo para realizar la traza del conocimiento en sistemas MBR. Esta traza es equivalente a la TC de los tutores cognitivos. Para separar las CK-sesiones en donde se realiza la evaluación del alumno, se proponen cinco criterios diferentes, algunos orientados a un uso evaluativo del sistema y otros tienen carácter instructivo.
   - En base a la agrupación conceptual anterior, se formaliza un algoritmo de evaluación que combina los mecanismos de recolección de evidencias de la metodología de evaluación sumativa con la traza del conocimiento. Con este algoritmo se puede determinar en cada momento el conocimiento del alumno sobre cada componente de la agrupación, lo cual se asocia al grado de consecución de objetivos y permite dirigir la estrategia instructiva.
   - Las capacidades instructivas de la propuesta se basan en el instrumento principal que en el MBR guía la enseñanza del alumno: la adaptación. La metodología propuesta redefine la forma en que ésta se lleva a cabo, sustituyendo los heurísticos por los mecanismos bien fundamentados basados en la TRI. De esta forma, la adaptación se aplica de dos maneras:
     - Mediante la selección del siguiente problema: Para esta tarea se pueden usar los mecanismos de selección de la TRI gracias a las CCP (equivalentes a las CCIC). No obstante, el aprendizaje del alumno no está contemplado como objetivo en la TRI por lo que es necesario definir otras estrategias que tengan esta meta. En este sentido se proponen tres grupos de estrategias:
       - Estrategias básicas, que no tienen en cuenta los objetivos de aprendizaje. En este grupo se proponen 8 estrategias diferentes, algunas orientadas a aprendizaje y otras orientadas a la evaluación.

- Estrategias que están guiadas por uno o múltiples objetivos concretos que se deben intentar cubrir. Éstas usan alguna de las estrategias básicas para seleccionar el problema pero restringiendo su ámbito a los objetivos fijados.
- Estrategias generales, en las que no hay un objetivo predefinido y buscan hacer que el alumno mejore de manera general. Se proponen dos criterios asociados a estrategias diferentes: el primero utiliza el grado de cumplimiento de los objetivos y el segundo usa un nuevo tipo de curva característica que se define sobre las componentes de la agrupación conceptual. En cualquier caso, primero se selecciona un objetivo usando alguno de estos dos criterios y, posteriormente, se aplica alguna de las estrategias guiadas por este objetivo.

- Mediante el refuerzo mostrado: En el MBR a la hora de mostrar varios refuerzos es necesario determinar cuál mostrar primero. El orden de este refuerzo es donde se realiza la adaptación. Para sustituir la forma original de llevarse a cabo, también basada en heurísticos, se proponen tres técnicas que usan el modelo de evaluación para determinar el refuerzo más apropiado para el alumno. La primera utiliza las CCR para ello; la segunda se basa en el uso de la agrupación de conceptos y los objetivos de aprendizaje; y la tercera, usa refuerzos asociados directamente a conceptos en lugar de a restricciones. Ésta última está inspirada en mecanismos parecidos existentes en el MBR pero con la diferencia de la base proporcionada por la TRI.

Estas formas de adaptación, junto con un modelo abierto del alumno, que se muestra usando las estimaciones del conocimiento de la agrupación conceptual, sirven como refuerzo que complementa la evaluación sumativa para dar lugar a la evaluación formativa del alumno.

- Se proponen tres modos de funcionamiento recomendados que cualquier sistema MBR que use el modelo de evaluación formativa presentado debería implementar. Cada modo implica una forma apropiada de recolectar evidencias, un mecanismo de agrupación en CK-sesiones, y una estrategia de adaptación acorde con el modo. Los tres modos tienen un propósito diferente: la calibración de las curvas, que busca la obtención de las CCR requeridas posteriormente por el proceso de evaluación sumativo; la evaluación sumativa final, en la que el objetivo es evaluar principalmente; y evaluación formativa, cuyo objetivo se centra en hacer que el alumno aprenda.

4. *Aportaciones a la construcción de EIRP*: En base a la metodología de evaluación que combina el MBR con la TRI se proporcionan las siguientes facilidades para la construcción de EIRP:

   - Generalización del modelo: Partiendo de la metodología de evaluación sumativa se dan las pautas y los requisitos que, de manera muy general, permitirían aplicar la evaluación sumativa de la TRI a otros EIRP que no estén basados en el MBR pero que disponen de evidencias que pueden ser utilizadas como base para aplicar la TRI.
   - Utilizando la analogía entre los sistemas de tests y los sistemas MBR se ha propuesto una técnica que permite estudiar la calidad de las restricciones

para realizar la evaluación. Esta técnica consiste en usar las herramientas existentes de la TRI para analizar la calidad de las preguntas, pero aplicado al MBR. El método propuesto usa la función de información para detectar diferentes situaciones de error sobre un modelo del dominio existente: restricciones incorrectamente codificadas, restricciones correctas pero que modelan los principios del dominio con un nivel de generalidad no apropiado, y restricciones correctas pero que no tienen suficiente evidencia. Respecto al nivel de generalidad, las restricciones pueden estar agrupando diversos principios del dominio o ser demasiado específicas, siendo recomendable dividirlas en varias o agruparlas en una, respectivamente. De esta forma, la técnica puede ser utilizada en la construcción de EIRP bajo la metodología MBR + TRI para mejorar los elementos que componen el modelo de dominio del sistema.

- Con el objetivo de facilitar la construcción de futuros EIRP que deseen usar la metodología propuesta, se ha desarrollado el marco de trabajo *CBMEngine*. Éste ofrece servicios de evaluación para sistemas tutores externos. De esta forma, el marco de trabajo es una componente reutilizable que cualquier EIRP bajo el paradigma MBR puede usar, siempre que registre su modelo de dominio y pase la información necesaria al marco de trabajo. Para la construcción del modelo de dominio en el marco de trabajo se ha implementado otra herramienta de edición visual llamada CBM-DoME, la cual todavía está siendo desarrollada pero que ya permite la edición de restricciones y las estructuras necesarias para realizar el mecanismo de detección de errores del MBR. CBMEngine se ha usado para proporcionar evaluación a dos sistemas tutores que caen fuera del trabajo de esta tesis: Visual Nets, que trata los conceptos de redes de comunicaciones; y PIPSE, que trata sobre estudio de inversiones.

5. *Aportaciones desde el punto de vista de la implementación*: Además de los sistemas implementados ya mencionados (CBMEngine y CBM-DoME) y su integración con sistemas externos, los modelos de evaluación anteriores han sido implementados en varios sistemas tutores que pueden ser utilizados para evaluar / enseñar en los siguientes dominios:

   - El tutor OOPS permite enseñar a los alumnos los fundamentos de programación orientada a objetos.
   - El tutor Simplex se centra en el dominio de la optimización lineal mediante el algoritmo Simplex y el de las dos fases.
   - Otra herramienta que se ha implementado es SQLTutor Log Processor. Aunque ésta es específica para procesar datos del tutor SQLTutor, puede ser fácilmente reutilizable prácticamente en su totalidad para tratar los datos de otros sistemas tutores y generar la evaluación sumativa a posteriori.

## 8.2. Limitaciones

Aunque la combinación propuesta entre la TRI y el MBR proporciona una solución a las limitaciones detectadas en ambos paradigmas, el modelo propuesto posee también sus limitaciones. Éstas se enumeran a continuación:

- Uno de los inconvenientes de esta propuesta es heredado de la TRI y se encuentra en la calibración que es necesaria realizar para poder aplicarse. Dado que la fiabilidad de la evaluación depende de la correcta estimación del modelo, es necesario tener una población de estudiantes elevada para que la muestra sea lo suficientemente representativa. Esta población debe ser mayor conforme mayor es el número de restricciones, para garantizar que hay suficiente evidencia sobre el conjunto completo de las mismas.

- El grado de formalidad y objetividad de la metodología depende de la calidad de cada una de las restricciones en el modelo de dominio. Es por ello que, si se quiere usar una evaluación cuya objetividad y formalidad sea alta, es requerido un estudio exhaustivo sobre cada restricción para garantizar que ésta refleja el principio adecuado y que no posee anomalías. Este proceso implica un estudio que no queda sólo en la aplicación de la metodología propuesta en esta tesis para ver la calidad de las restricciones tras la calibración, sino que se extiende a todo el proceso de uso, aplicando otras herramientas como las propuestas en el apartado 5.4.3 con el fin de asegurar la calidad del modelo del dominio. Por este motivo, la aplicación de esta metodología para certificar un nivel de conocimiento de un estudiante de una manera rigurosa requiere de un alto esfuerzo de análisis.

- Otra limitación, heredada del MBR, se encuentra en la aplicación de la metodología propuesta a dominios donde la solución a los problemas no es altamente informativa diagnósticamente. Esto quiere decir que la solución por sí misma no proporciona información suficiente que permita realizar el diagnóstico del alumno. En este tipo de dominios es necesario el uso de las restricciones camino, introducidas por Mitrovic y Ohlsson (2006) y explicadas en la sección 2.3.2, las cuales son equivalentes a las reglas de producción de los tutores cognitivos. En ese caso, podría ser que el supuesto de independencia local de las restricciones no se cumpliera ya que dos reglas asociadas a un camino de resolución común pueden no ser sucesos independientes. Una investigación más profunda es necesaria en este aspecto.

- Si se utiliza el marco de trabajo CBMEngine para proporcionar la evaluación y éste usa el mecanismo de evaluación proporcionado por sistemas de tests, es necesario mantener un modelo del dominio y del alumno redundante en los dos sistemas. Primeramente, en el marco CBMEngine para recopilar las evidencias y aplicar las restricciones y, paralelamente, en el sistema de tests para poder aplicar los mecanismos de inferencia de la TRI. Actualmente no existe un mecanismo que realice el proceso de creación del dominio de manera automática en el sistema de test a partir del modelo de dominio existente en CBMEngine, por lo que el proceso de creación se debe realizar manual, resultando en una tarea tediosa.

- Otra de las limitaciones de la propuesta es que para incrementar la seguridad del sistema en relación con los problemas que se usan en la evaluación sumativa final, sería necesario mantener un elevado número de problemas. Esto es así porque habría que garantizar que durante la etapa de formación no son comprometidos, siendo necesario mantener un conjunto de problemas para la evaluación y otros para formación.

- Es necesario realizar un estudio empírico para determinar la efectividad de la

propuesta como herramienta instructiva. Esta característica se ubica fuera del alcance de esta tesis, siendo contemplado dentro de las líneas abiertas. No obstante se menciona aquí para hacer notar que, en el estado actual de la propuesta, se desconoce su eficiencia educativa.

- Aunque las herramientas tutores implementadas, OOPS y Simplex Tutor, permiten realizar la evaluación del alumno sirviéndose de herramientas como MULTILOG, el mecanismo de evaluación formativa no está integrado. Esto es así porque estas herramientas se han usado principalmente para el estudio de la evaluación sumativa. Otra característica que supone una limitación es el tipo de refuerzo que proporcionan los sistemas, el cual sólo abarca dos de los tipos propuestos en el MBR, explicados en la sección 2.3.4. Además, el sistema OOPS está definido sobre un lenguaje que ya no se utiliza, por lo que no se puede emplear actualmente para la enseñanza.

## 8.3. Líneas de investigación abiertas

De acuerdo con la metáfora de la doctora Mitrovic que abría este capítulo, la combinación propuesta abre un puente de comunicación que se torna beneficioso para ambas islas de conocimiento implicadas en esta tesis. Dada la dimensión de lo que todavía queda por investigar, podría decirse que este trabajo no hecho más que construir el puente. Sin embargo, el tránsito que se espera con esta comunicación, permitirá extender, a través de un enorme abanico de posibilidades, la eficiencia de la IA en la educación.

Las líneas que quedan abiertas y hacia las que se dirige el trabajo futuro se han dejado ver a lo largo de este documento y, especialmente, en el capítulo 5, donde se presenta el modelo formativo para mejorar el aprendizaje del estudiante. Estas líneas son las siguientes:

- Los diferentes métodos de calibración han surgido como una idea en la parte final de la tesis, cuando ya se habían realizado los experimentos y como parte de la unión de los diferentes criterios que dan lugar a los modos de uso presentados en la sección 5.3.5. Es por ello que también es requerido estudiar la efectividad de cada una de las tres estrategias en relación con la calidad del ajuste del modelo.

- Con el fin de contemplar el conocimiento cambiante del estudiante y realizar la traza del mismo se ha propuesto el mecanismo de las CK-sesiones, el cual puede realizarse de acuerdo a diversos criterios (ver sección 5.2.2.3). En la experimentación realizada se ha estudiado uno de estos criterios, siendo necesario contemplar los restantes y determinar cuándo es más adecuado cada uno, de acuerdo a las necesidades de la evaluación que se realice, ya sea sumativa final o formativa.

- En todos los estudios realizados se han utilizado modelos paramétricos de la TRI por su simplicidad de implementación y porque las restricciones equivalen a ítems con sólo dos respuestas, por lo que con estos modelos bastaba para un estudio inicial. No obstante, se deberían comparar diversos aspectos de los modelos paramétricos respecto de los no paramétricos para su uso en la evaluación del conocimiento en dominios procedimentales, tales como la calidad del ajuste

en la calibración, eficiencia, efecto en la implementación, etc. Además sería interesante estudiar el uso de modelos multidimensionales y extender la agrupación conceptual del modelo de dominio en una estructura jerárquica con varios niveles de agrupaciones.

- Teóricamente, el resultado de la primera vez que la restricción es relevante refleja el conocimiento del alumno evitando el aprendizaje que se produce con el refuerzo presentado. Sin embargo, también es razonable pensar que otros métodos pueden ser iguales o más efectivos si tienen en cuenta las evidencias que este método descarta. Por ejemplo, si el alumno satisface una restricción la primera vez que ésta es relevante y, seguidamente, la viola tres veces consecutivas, es más lógico considerar la violación que la satisfacción. En este sentido, los métodos propuestos no usan esta información adicional que podría hacer más efectivo el modelo. Este tipo de mecanismos, como se vio en la sección 5.2.2, son inviables para un uso formativo. Por este motivo se deberían estudiar sólo como alternativa en la evaluación sumativa o la calibración de las restricciones.

- Un pequeño matiz que puede influir en la emisión del juicio en la fase de evaluación sumativa final es la comprobación de la hipótesis que establece que en esta fase sería recomendable el uso del nivel más bajo de refuerzo. Este tipo de refuerzo debería compararse con la opción de no ofrecer ningún refuerzo y se debería mirar cuál es la más efectiva para esta tarea.

- Una de las grandes tareas que quedan pendientes por realizar es la investigación de la parte formativa de la propuesta: en primer lugar, del uso de modelos de la TRI que permitan manejar el aprendizaje para proporcionar adaptación mientras el estudiante está usando el sistema; y, segundo, de la eficiencia de las diferentes estrategias instructivas propuestas en la sección 5.3.3. Respecto a esta última es necesario investigar varios aspectos sobre las diferentes combinaciones que se pueden realizar. En relación con el refuerzo, dado que éste está relacionado solamente con el proceso de evaluación formativa del alumno, es necesario determinar que mecanismo tiene mayor influencia positiva en el aprendizaje. Respecto a la selección de problemas, hay que estudiar las múltiples combinaciones posibles para ver cuáles tienen mayor influencia de acuerdo dos metas diferentes: evaluación y aprendizaje. Cuando se aplica la evaluación sumativa final, lo importante es la emisión del juicio, por lo que se debe estudiar cuáles hacen que el proceso de evaluación sea más efectivo. Cuando se aplica la evaluación formativa, lo deseable es que el alumno aprenda y domine los objetivos de aprendizaje, por lo que se debe estudiar cuál de las estrategias es mejor para esta meta. Se prevé que esta tarea, dado el número de estrategias posibles y de los diferentes aspectos involucrados, puede requerir un esfuerzo considerable. No obstante, es una de las que más importancia tiene para el desarrollo futuro de la metodología. Como parte de la misma, sería necesario considerar otros mecanismos con control de exposición de los problemas, como los propuestos por Barrada (2012).

- Tras cumplir la línea de trabajo anterior, es necesario comprobar que la propuesta mejora la instrucción del alumno en comparación con los sistemas MBR tradicionales. Aunque la base del modelo es bien fundamentada, en comparación con los heurísticos del MBR utilizados para guiar la instrucción, es necesario confirmar que la propuesta realmente mejora la eficiencia instructiva del sistema. Para ello

sería necesario comparar el efecto en el aprendizaje entre un sistema MBR tradicional y el mismo sistema pero implementado la metodología propuesta. En un nivel mucho más general y lejano en el tiempo, sería deseable comparar también la efectividad de la metodología MBR + TRI con la obtenida mediante tutores cognitivos.

- Durante el desarrollo de esta tesis, uno de los problemas con los que se ha tenido que lidiar es la poca disponibilidad de datos reales sobre los que realizar los estudios. Cuando hemos dispuesto de datos, como los de SQL-Tutor, éstos no aplicaban nuestra metodología y había que descartar muchos datos. Por estos motivos sería deseable investigar el efecto de la metodología en una población considerable, en la que las conclusiones que se infieran sean significativas, y con un periodo de uso lo suficientemente extenso como para determinar la efectividad del enfoque propuesto en un entorno más realista.

- Para estudiar la calidad de las restricciones como instrumento de evaluación, se ha utilizado la función de información aplicada a las restricciones. Concretamente, se ha usado el área de la curva determinada por la mencionada función como indicativo de la calidad. También existen otras propiedades de la curva como la curtosis o el valor máximo que pueden servir para detectar anomalías en el modelo de dominio. Sería interesante estudiar estas propiedades en la función de información y compararlas con la utilizada en esta tesis. Además, también existen otras herramientas en los sistemas de test que podrían proporcionar diferentes utilidades para la mejora del modelo de dominio, las cuales habría que investigar.

- El marco de trabajo CBMEngine todavía requiere extender su funcionalidad para proporcionar una componente completa y autosuficiente que pueda ser utilizada por sistemas MBR externos. Principalmente, es necesario ampliar la herramienta CBM-DoME para la edición del modelo de dominio y su integración como parte del mencionado marco de trabajo. Respecto al uso del sistema Siette para realizar la evaluación, también es necesario completar la integración entre los dos sistemas y desarrollar un mecanismo que permita la generación automática del modelo subyacente en el sistema de tests, evitando la tediosa tarea de creación manual de las preguntas compuestas. Además, también sería recomendable estudiar la eficiencia de uso del marco de trabajo en sistemas externos, puesto que el protocolo de comunicación recae en servicios Web y cabe la posibilidad de que éstos impusieran algún tipo de restricción o mejora necesaria relacionada con la eficiencia.

- En los sistemas tutores implementados es necesaria la extensión de los mismos para incluir la evaluación formativa como partes de los mismos. Para ello es necesario comunicar estos sistemas con el marco de trabajo CBMEngine, lo que supone reestructurarlos para contemplar los diversos tipos de refuerzo y los modos de funcionamiento (calibración, formación y evaluación final); extender la lógica de funcionamiento; abrir su modelo del alumno, por ejemplo usando la herramienta Ingrid (Cruces et al., 2010; Conejo et al., 2011) desarrollada en el grupo de investigación al que pertenece el doctorando; y pasar su modelo de dominio a CBMEngine. Además, el sistema OOPS utiliza un lenguaje obsoleto que, aunque su cambio por uno más moderno como Java no supone una gran complicación, es necesario cambiar para poder volver a usarse de nuevo en el futuro.

- Otra característica que sería deseable y que no se ha contemplado en este trabajo es la posibilidad de añadir nuevas restricciones una vez que el sistema ha comenzado su utilización por parte de un alumnado. Por este motivo habría que investigar mecanismos que permitan realizar la calibración de las nuevas restricciones sin necesidad de tener que volver a calibrar el conjunto completo y su efecto en la eficiencia. También, sería recomendable estudiar la utilización de métodos de anclaje y equiparación en sistemas con un número muy grande de restricciones donde es necesario dividir el conjunto total en varias sesiones de problemas para la calibración.

- Aparte de la aplicación del modelo propuesto para suplir los heurísticos del MBR existen numerosos estudios que extienden la utilidad de este paradigma a diversas áreas de los STI, como los diálogos tutoriales, la colaboración, el campo afectivo, etc. Sería deseable estudiar si en esos campos, la propuesta puede mejorar la eficiencia del sistema.

- Por último, como se mencionaba en la sección 7.8, el estudio realizado sobre las diversas propiedades psicométricas ha sido muy reducido, centrándose en un tipo concreto de validez. No obstante, existen muchos otros tipos de validez que podrían ser estudiados (Moss et al., 2006). En este campo también existen otras formas de comprobar la fiabilidad del instrumento de evaluación (Cook y Beckman, 2006), así como también, diversos índices para estudiar la precisión y la consistencia proporcionada por la TRI (Wyse y Hao, 2012). Estas medidas y métodos para estudiar otras propiedades de la metodología de evaluación serían muy interesantes de estudiar de cara a afianzar la eficiencia de la metodología.

# Parte VII

# Apéndices

Esta parte contiene dos apéndices. En primer lugar, el apéndice A detalla información sobre los servicios Web que un EIRP externo puede utilizar para aplicar el modelo desarrollado en esta tesis. Seguidamente, el apéndice B presenta en inglés un resumen de la tesis, así como la traducción literal a esta lengua del capítulo de conclusiones.

# Apéndice A

# Servicios Web de CBMEngine

*El que no vive para servir,*
*no sirve para vivir.*

Madre Teresa de Calcuta (1910 - 1997)

**Resumen:** En este apéndice se incluye información más técnica sobre cómo usar CBMEngine por un EIRP externo.

Como ya se mencionó en la sección 6.5, CBMEngine es un marco de trabajo que permite a sistemas externos aplicar los modelos de evaluación sumativa y formativa desarrollados en esta tesis. Para ello la plataforma cuenta con una serie de servicios Web que son el punto de acceso a esta funcionalidad. El objetivo de este apéndice es ampliar la información relacionada con estos servicios a modo de guía de integración entre CBMEngine y cualquier EIRP externo, que en este apéndice será referido como *Sistema Externo* (SE) para abreviar.

Además de los servicios Web, la plataforma CBMEngine dispone de una interfaz Web accesible públicamente en la dirección http://cbm.iaia.lcc.uma.es/CBMEngine. Esta interfaz tiene dos objetivos principales:

- En primer lugar, busca ser un punto de acceso informativo para los sistemas que desean hacer uso de los servicios de la plataforma. De esta forma, la documentación existente es centralizada en esta interfaz, así como cualquier otra información útil para sistemas externos.
- En segundo lugar, pretende proporcionar los medios necesarios para realizar las tareas de administración básicas tales como el alta de nuevos dominios, gestión de usuarios y gestión de las opciones generales de configuración de la plataforma.

La apariencia actual de la interfaz puede verse en la figura A.1, donde el menú principal contiene cuatro apartados: la página inicial que da acceso a los documentos descriptivos de cada servicio Web; una sección de administración; un apartado de documentación; y un enlace para contacto con el autor de esta plataforma. La parte más importante desde el punto de vista de uso de la interfaz es, quizás, la elicitación de nuevos dominios dentro del apartado de administración. No obstante, esa parte está todavía por completarse e integrarse, lo cual está previsto una vez se finalice la herramienta de elicitación CBMDoME, explicada en la sección 6.6.


Figura A.1: Interfaz de configuración de CBMEngine.

Antes de que un SE pueda hacer uso de los servicios Web que se describen en este apéndice, es requisito necesario que éste haya sido dado de alta en CBMEngine. Esto implica que su modelo de dominio haya sido elicitado mediante la codificación de las restricciones y la definición de las estructuras involucradas en la comunicación. Aunque actualmente este proceso se realiza manualmente programando cada elemento, en un futuro se usará la herramienta CBMDoME para esta tarea.

El contenido de este apéndice se estructura en dos secciones. Primeramente se explican las características de los diferentes servicios Web de la plataforma. Por último, en la sección A.2, se describe el protocolo de uso de los diferentes servicios Web para que los mecanismos de evaluación puedan ser aplicados de la manera adecuada.

## A.1. Servicios Web

La plataforma CBMEngine proporciona servicio a sistemas externos a través de tres puntos de entrada SOA que agrupan servicios Web en base a su funcionalidad: gestión propiamente de los elementos del modelo del dominio y del alumno; servicios relacionados con la actividad realizada durante una sesión de uso; y manejo de los mecanismos de la TRI. Cada punto de entrada dispone de un documento *Web Service Definition Language* (WSDL) donde se describe el conjunto de servicios Web en términos de los parámetros que necesitan para ser invocados, su tipo, y el tipo de resultado que devuelve.

Cada dominio definido en CBMEngine tendrá los tres puntos de entrada mencionados anteriormente. La URL del documento que define cada punto de entrada se obtiene combinando la dirección de acceso principal de CBMEngine con un identificador del dominio y un texto concreto. A continuación se menciona, para los tres tipos de servicio, la URL de su documento WSDL y se describen las características de los servicios Web disponibles actualmente. Aunque estos servicios pueden cambiar en un futuro, de acuerdo a nuevos elementos incorporados o a la modificación de los ya exis-

tentes, la documentación actualizada de los mismos podrá ser consultada en la sección *documentación* de la página Web de CBMEngine.

### A.1.1. Servicios de gestión

La URL del documento WSDL asociado a este punto de entrada se obtiene usando el texto "ManagementService?wsdl". Así pues, para el ejemplo del sistema Visual Nets, cuyo identificador del dominio es "Net", el WSDL resultante estaría accesible en la URL http://cbm.iaia.lcc.uma.es/CBMEngine/NetManagementService?wsdl.

| **Servicio** | `addUser` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos la creación de usuarios. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `newUser`: Datos del nuevo usuario, como el identificador, la clave, el email, etc. |
| Devuelve | El identificador numérico asociado al nuevo usuario. |

Tabla A.1: Descripción del servicio `addUser`.

| **Servicio** | `getUser` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos obtener información sobre un usuario. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `userId`: Identificador del usuario del que se desea obtener información. |
| Devuelve | Datos del usuario consultado. |

Tabla A.2: Descripción del servicio `getUser`.

| **Servicio** | `updateUser` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos actualizar información de otro usuario. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `user`: Datos nuevos del usuario a modificar. |
| Devuelve | - |

Tabla A.3: Descripción del servicio `updateUser`.

| **Servicio** | `deleteUser` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos la eliminación de usuarios. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `userId`: Identificador del usuario a borrar. |
| Devuelve | - |

Tabla A.4: Descripción del servicio `deleteUser`.

| **Servicio** | `existsUser` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos comprobar si existe un usuario. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `userId`: Identificador del usuario a comprobar. |
| Devuelve | Valor lógico que indica si el usuario existe. |

Tabla A.5: Descripción del servicio `existsUser`.

| **Servicio** | `addProblemXXX` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos añadir un problema. El sufijo XXX en el nombre se asocia a un tipo de problema del dominio, existiendo un servicio por cada tipo de problema. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `newProblem`: Datos del nuevo problema (esta estructura externa es diferente en cada tipo de problema). |
| Devuelve | Identificador numérico del nuevo problema. |

Tabla A.6: Descripción del servicio `addProblemXXX`.

| **Servicio** | `getProblemXXX` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos obtener información de un problema. El sufijo XXX en el nombre se asocia a un tipo de problema del dominio, existiendo un servicio por cada tipo de problema. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `problemId`: Identificador del problema que se consulta. |
| Devuelve | Datos del problema consultado. |

Tabla A.7: Descripción del servicio `getProblemXXX`.

| **Servicio** | `updateProblemXXX` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos actualizar la información de un problema. El sufijo XXX en el nombre se asocia a un tipo de problema del dominio, existiendo un servicio por cada tipo de problema. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `problem`: Datos del problema (esta estructura externa es diferente en cada tipo de problema). |
| Devuelve | - |

Tabla A.8: Descripción del servicio `updateProblemXXX`.

| **Servicio** | `deleteProblem` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos borrar un problema. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `problemId`: Identificador del problema que se desea borrar. |
| Devuelve | - |

Tabla A.9: Descripción del servicio `deleteProblem`.

| **Servicio** | `existsProblem` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos consultar si existe un problema. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `problemId`: Identificador externo del problema a comprobar. |
| Devuelve | Valor lógico que indica si el problema existe. |

Tabla A.10: Descripción del servicio `existsProblem`.

| **Servicio** | `changeWorkingMode` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos cambiar el modo de funcionamiento de CBMEngine en cuanto al tratamiento de las evidencias. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `mode`: Nuevo modo de operación, el cual puede tomar los valores:<br>• 0 (*modo Calibración*): Para realizar la calibración de las curvas características.<br>• 1 (*modo Formativo*): Para realizar la evaluación formativa.<br>• 2 (*modo Sumativo*): Para realizar la evaluación sumativa. |
| Devuelve | - |

Tabla A.11: Descripción del servicio `changeWorkingMode`.

| **Servicio** | `getWorkingMode` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos consultar el modo de funcionamiento actual de CBMEngine en cuanto al tratamiento de las evidencias. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos. |
| Devuelve | El modo actual, que será uno de los tres valores mencionados en el servicio anterior. |

Tabla A.12: Descripción del servicio `getWorkingMode`.

### A.1.2. Servicios Web asociados a la TRI

Para este punto de entrada la URL de acceso al documento descriptivo se obtiene con el texto "IRTService?wsdl". De la misma forma que antes, con el sistema Visual Nets, esta URL sería http://cbm.iaia.lcc.uma.es/CBMEngine/NetIRTService?wsdl.

| **Servicio** | `calibrate` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos calibrar las restricciones de un SE. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario. |
| Devuelve | - |

Tabla A.13: Descripción del servicio `calibrate`.

| **Servicio** | `getFormativeAssessment` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos obtener la evaluación formativa de otro alumno. Tanto este, como el resto de servicios de evaluación formativa, no se deberían usar al ser necesario todavía su extensión con modelos de la TRI que incluyan el aprendizaje. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `userId`: Identificador del usuario sobre el que se realiza la consulta.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera (entre 0 y 100) o la propia de la TRI (perteneciente al intervalo $(-\infty, \infty)$). |
| Devuelve | Lista de estimaciones del conocimiento, cada uno corresponde a una CK-sesión. |

Tabla A.14: Descripción del servicio `getFormativeAssessment`.

| **Servicio** | `getFormativeAssessment` |
|---|---|
| Función | Permite a un usuario obtener la evaluación formativa propia. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera o de la TRI. |
| Devuelve | Lista de estimaciones del conocimiento, cada uno corresponde a una CK-sesión. |

Tabla A.15: Descripción del servicio `getFormativeAssessment` (2).

| **Servicio** | `getBoundedFormativeAssessment` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos obtener la evaluación formativa de otro alumno en un intervalo acotado de tiempo. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `userId`: Identificador del usuario sobre el que se realiza la consulta.<br>▪ `from`: Fecha límite inferior del intervalo.<br>▪ `to`: Fecha límite superior del intervalo.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera o de la TRI. |
| Devuelve | Lista de estimaciones del conocimiento, cada uno corresponde a una CK-sesión. |

Tabla A.16: Descripción del servicio `getBoundedFormativeAssessment`.

| **Servicio** | `getBoundedFormativeAssessment` |
|---|---|
| Función | Permite a un usuario obtener la evaluación formativa propia en un intervalo acotado de tiempo. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `from`: Fecha límite inferior del intervalo.<br>▪ `to`: Fecha límite superior del intervalo.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera o de la TRI. |
| Devuelve | Lista de estimaciones del conocimiento, cada uno corresponde a una CK-sesión. |

Tabla A.17: Descripción del servicio `getBoundedFormativeAssessment` (2).

| **Servicio** | `getSumativeAssessment` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos obtener la evaluación sumativa de otro alumno. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `userId`: Identificador del usuario sobre el que se realiza la consulta.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera o de la TRI. |
| Devuelve | Estimación del conocimiento. |

Tabla A.18: Descripción del servicio `getSumativeAssessment`.

| **Servicio** | `getSumativeAssessment` |
|---|---|
| Función | Permite a un usuario obtener la evaluación formativa propia. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera o de la TRI. |
| Devuelve | Estimación del conocimiento. |

Tabla A.19: Descripción del servicio `getSumativeAssessment` (2).

| **Servicio** | `getBoundedSumativeAssessment` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos obtener la evaluación sumativa de otro alumno en un intervalo acotado de tiempo. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `userId`: Identificador del usuario sobre el que se realiza la consulta.<br>▪ `from`: Fecha límite inferior del intervalo.<br>▪ `to`: Fecha límite superior del intervalo.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera o de la TRI. |
| Devuelve | Estimación del conocimiento. |

Tabla A.20: Descripción del servicio `getBoundedSumativeAssessment`.

| **Servicio** | `getBoundedSumativeAssessment` |
|---|---|
| Función | Permite a un usuario autenticado obtener la evaluación sumativa propia en un intervalo acotado de tiempo. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `from`: Fecha límite inferior del intervalo.<br>▪ `to`: Fecha límite superior del intervalo.<br>▪ `trueScore`: Valor lógico para especificar si se desea la estimación en la escala de puntuación verdadera o de la TRI. |
| Devuelve | Estimación del conocimiento. |

Tabla A.21: Descripción del servicio `getBoundedSumativeAssessment` (2).

| **Servicio** | `requestNextProblem` |
|---|---|
| Función | Permite a un usuario con permisos obtener el siguiente problema a presentar a otro usuario en base a una estrategia de selección particular |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos.<br>▪ `userId`: Identificador del usuario sobre el que se realiza la consulta.<br>▪ `criterion`: Criterio de selección del siguiente problema, el cual será alguno de los criterios básicos explicados en la sección 5.3.3.1:<br>• 1: Selección por máxima información.<br>• 2: Selección por dificultad.<br>• 3: Selección bayesiana (por completarse).<br>• 4: Selección por problema problemático (SPP).<br>• 5: Selección por restricción problemática (SRP).<br>• 6: Selección por máxima información y SRP.<br>• 7: Selección por dificultad y SRP.<br>• 8: Selección bayesiana y SRP (por completarse). |
| Devuelve | - |

Tabla A.22: Descripción del servicio `requestNextProblem`.

| **Servicio** | `requestNextProblem` |
|---|---|
| Función | Permite a un usuario autenticado obtener el siguiente problema más adecuado para él en base a una estrategia de selección particular. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario que realiza la consulta.<br>▪ `criterion`: Criterio de selección del siguiente problema (mismos valores que el servicio anterior). |
| Devuelve | - |

Tabla A.23: Descripción del servicio `requestNextProblem` (2).

| **Servicio** | `changeGroupingMode` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos cambiar el modo de agrupación de las evidencias. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `groupingMode`: Modo de agrupación nuevo, el cual puede tomar alguno de los valores explicados en la sección 5.2.2.3:<br>• 1: Agrupación mediante un umbral TCK.<br>• 2: Agrupación mediante problemas intentados.<br>• 3: Sin agrupación (considera el conjunto completo de evidencias).<br>• 4: Agrupación por intervalos.<br>• 5: Agrupación por cambio significativo del conocimiento.<br>• 6: Agrupación por ratio de uso. |
| Devuelve | - |

Tabla A.24: Descripción del servicio `changeGroupingMode`.

| **Servicio** | `getGroupingMode` |
|---|---|
| Función | Permite a un usuario autenticado y con permisos consultar el modo de agrupación actual de las evidencias en CBMEngine. |
| Parámetros | ▪ `sessionId`: Identificador de sesión del usuario con permisos. |
| Devuelve | El modo actual, que será uno de los valores mencionados en el servicio anterior. |

Tabla A.25: Descripción del servicio `getGroupingMode`.

### A.1.3. Servicios de control de sesiones

En este caso, la dirección de acceso al documento WSDL se obtiene con el texto "SessionService?wsdl". Siguiendo con el ejemplo del sistema Visual Nets, esta dirección sería http://cbm.iaia.lcc.uma.es/CBMEngine/NetSessionService?wsdl.

| **Servicio** | `createNewAdminSession` |
|---|---|
| Función | Permite crear una sesión de trabajo para tareas de gestión de usuario o problemas. Con esta sesión no se pueden resolver problemas. |
| Parámetros | ▪ `userId`: Identificador del usuario.<br>▪ `password`: Clave del usuario. |
| Devuelve | El identificador de la sesión creada. |

Tabla A.26: Descripción del servicio `createNewAdminSession`.

| **Servicio** | `createNewSession` |
|---|---|
| Función | Permite crear una sesión de trabajo para proporcionar las evidencias a CBMEngine. |
| Parámetros | ▪ `userId`: Identificador del usuario.<br>▪ `password`: Clave del usuario. |
| Devuelve | El identificador de la sesión creada. |

Tabla A.27: Descripción del servicio `createNewSession`.

| **Servicio** | `createNewSessionWithProblem` |
|---|---|
| Función | Permite crear una sesión de trabajo y, a la vez, indicar a CBMEngine el problema sobre el que se van a proporcionar las evidencias. |
| Parámetros | ▪ `userId`: Identificador del usuario.<br>▪ `password`: Clave del usuario.<br>▪ `externalProblemId`: Identificador del problema en el SE. |
| Devuelve | El identificador de la sesión creada. |

Tabla A.28: Descripción del servicio `createNewSessionWithProblem`.

| **Servicio** | `setSessionProblem` |
|---|---|
| Función | Permite indicar a CBMEngine el problema sobre el que se van a proporcionar las evidencias. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `externalProblemId`: Identificador del problema en el SE. |
| Devuelve | - |

Tabla A.29: Descripción del servicio `setSessionProblem`.

| **Servicio** | `getSessionProblem` |
|---|---|
| Función | Permite consultar el problema actual del que se está proporcionando evidencias. |
| Parámetros | ▪ `sessionId`: Identificador de sesión. |
| Devuelve | Identificador externo del problema actual. |

Tabla A.30: Descripción del servicio `getSessionProblem`.

| **Servicio** | `setSolutionXXX` |
|---|---|
| Función | Permite establecer la solución del alumno para el problema sobre el que actualmente se está proporcionando evidencias. El sufijo XXX en el nombre se asocia a un tipo de problema del dominio, existiendo un servicio por cada tipo de problema al que pertenece la solución. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `solution`: Datos de la solución (esta estructura externa es diferente en cada tipo de problema). |
| Devuelve | - |

Tabla A.31: Descripción del servicio `setSolutionXXX`.

| **Servicio** | `getSolutionXXX` |
|---|---|
| Función | Permite consultar la solución actual del alumno para el problema del que se está proporcionando evidencias. El sufijo XXX en el nombre se asocia a un tipo de problema del dominio, existiendo un servicio por cada tipo de problema. |
| Parámetros | ▪ `sessionId`: Identificador de sesión. |
| Devuelve | Datos de la solución, de acuerdo al tipo de problema correspondiente. |

Tabla A.32: Descripción del servicio `getSolutionXXX`.

| **Servicio** | `setSolution` |
|---|---|
| Función | Permite establecer la solución del alumno para el problema sobre el que actualmente se está proporcionando evidencias. En este servicio la solución se representa mediante XML en un formato acordado entre CBMEngine y el SE, de forma que el XML puede ser procesado por CBMEngine. Es un método alternativo al servicio anterior en el que el tipo de problema se determina procesando el XML, por lo que este servicio es único e independiente del tipo de problema. |
| Parámetros | ▪ `sessionId`: Identificador de sesión.<br>▪ `xml`: Representación XML de la solución. |
| Devuelve | - |

Tabla A.33: Descripción del servicio `setSolution`.

| **Servicio** | `checkSession` |
|---|---|
| Función | Permite comprobar los errores existentes en la solución actual proporcionada previamente. |
| Parámetros | ▪ `sessionId`: Identificador de sesión. |
| Devuelve | Lista de restricciones violadas. Esta lista incluye el nombre de la restricción violada, un refuerzo, y los identificadores de los elementos que han producido cada error. |

Tabla A.34: Descripción del servicio `checkSession`.

| **Servicio** | `finishProblem` |
|---|---|
| Función | Permite indicar a CBMEngine que se ha terminado el problema sobre el que se estaba proporcionando evidencias pero que el alumno mantiene la sesión de trabajo. |
| Parámetros | ▪ `sessionId`: Identificador de la sesión asociada. |
| Devuelve | - |

Tabla A.35: Descripción del servicio `finishProblem`.

| **Servicio** | `closeSession` |
|---|---|
| Función | Permite cerrar una sesión de trabajo. |
| Parámetros | ▪ `sessionId`: Identificador de la sesión a cerrar. |
| Devuelve | - |

Tabla A.36: Descripción del servicio `closeSession`.

| **Servicio** | `storeEvidencesAndCheck` |
|---|---|
| Función | Permite realizar todos los pasos de comprobación de errores de una vez. Es decir, establece una nueva sesión, procesa la solución, comprueba los errores, y finaliza la sesión. |
| Parámetros | ▪ `userId`: Identificador del usuario al que corresponden las evidencias.<br>▪ `password`: Clave del usuario.<br>▪ `externalProblemId`: Identificador externo del problema al que corresponden las evidencias.<br>▪ `xml`: Representación XML de la solución. |
| Devuelve | Lista de restricciones violadas. Esta lista incluye el nombre de la restricción violada, un refuerzo, y los identificadores de los elementos que han producido cada error. |

Tabla A.37: Descripción del servicio `storeEvidencesAndCheck`.

## A.2. Protocolo de uso recomendado

Los servicios Web explicados anteriormente se deberían usar en el orden adecuado, de lo contrario, el sistema no podría dar una respuesta adecuada. Por ejemplo, no se pueden comprobar los errores cometidos por un alumno si no se ha proporcionado previamente una solución a CBMEngine. El orden recomendado y las posibles variantes de uso se mencionan a continuación agrupándolos en dos bloques: configuración de la plataforma y registro de la actividad del usuario. Previamente a cualquier operación de gestión el usuario administrador debe iniciar una sesión de administración con el servicio `createNewAdminSession`.

### A.2.1. Configuración de la plataforma

Antes de poder realizar cualquier actividad, el modelo de dominio del SE habrá sido elicitado en CBMEngine, por lo que las estructuras necesarias estarán disponibles para usarse en la plataforma. Cada SE dispondrá de un usuario administrador con el que deberá registrar al resto de usuarios, así como a las instancias de problemas que componen el modelo de dominio.

#### Registro de usuarios

1. Previamente a la creación de un usuario se debería comprobar si el usuario existe mediante el servicio `existsUser`.

2. Seguidamente, si éste no existe, se debería registrar mediante `addUser`.

3. Las tareas de administración (`getUser`, `updateUser`, y `deleteUser`) pueden realizarse una vez el usuario esté registrado.

**Registro de instancias de problemas**

1. Antes de registrar un problema es recomendable comprobar si éste existe anteriormente. Para eso, el servicio `existsProblem` comprueba que el identificador del problema, asociado al SE, no ha sido dado de alta previamente.

2. El registro de una instancia concreta de un problema se realiza mediante el servicio `addProblemXXX`, que estará condicionado al tipo de problema en cuestión.

3. Una vez el problema se haya registrado, las tareas de gestión pueden realizarse mediante los servicios `getProblemXXX`, `updateProblemXXX`, y `deleteProblem`.

**Otras opciones de configuración**

El uso de los mecanismos de la TRI requiere que las restricciones hayan sido calibradas a partir de información recopilada de los alumnos. Puesto que este servicio sólo toma la evidencia que ha sido proporcionada cuando el sistema ha estado funcionando en *modo calibración*, un SE debe tener en cuenta los tres modos de uso del sistema y seleccionarlos en el orden correcto con el servicio `changeWorkingMode`. Primeramente se debería realizar la calibración mediante el servicio `calibrate`. Para ello sería recomendable contar con la cantidad suficiente de información que permita estimar adecuadamente las CCR.

Cuando las restricciones se hayan calibrado se podrá usar la evaluación formativa y sumativa del alumno, la cual también usará la evidencia proporcionada mientras el sistema estaba en el modo asociado. Los servicios que proporcionan el conjunto de evaluaciones asociados a la traza del conocimiento son `getFormativeAssessment` y `getBoundedFormativeAssessment`, mientras que los asociados a la evaluación sumativa son `getSumativeAssessment` y `getBoundedSumativeAssessment`. Al seleccionar el modo de uso formativo o sumativo se debe también configurar el modo de agrupación de las restricciones con el servicio `changeGroupingMode`.

### A.2.2. Registro de la actividad del alumno

Para proporcionar la información de un alumno, un SE debería realizar la siguiente secuencia de pasos.

1. Iniciar sesión con el servicio `createNewSession`.

2. Seleccionar el siguiente problema. Aunque éste podrá ser determinado por algún mecanismo propio, es recomendable que se use el servicio `requestNextProblem` de CBMEngine, ya que utiliza los mecanismos de la TRI.

3. Indicar el problema sobre el que se van a proporcionar las evidencias. Alternativamente, se puede iniciar sesión y, a la vez, indicar este dato mediante el servicio `createNewSessionWithProblem`.

4. Proporcionar la solución que el alumno ha construido para el problema actual mediante el servicio `setSolutionXXX`, el cual será diferente dependiendo del tipo de problema que el alumno esté resolviendo. Alternativamente, se puede usar el servicio `setSolution` que incorpora tanto los datos de la solución como el tipo de problema asociado en una representación XML. El problema de este servicio

es que su implementación no es trivial y requiere desarrollar componentes que procesen la información contenida en el documento XML para ser transformada a estructuras entendibles por CBMEngine, con el consecuente esfuerzo adicional en la fase de elicitación del dominio.

5. Comprobar las restricciones violadas y satisfechas de la solución proporcionada mediante el servicio `checkSession`. CBMEngine informará de los errores producidos y actualizará el modelo del alumno con las nuevas evidencias.

6. El alumno puede realizar varios intentos sobre el mismo problema, siendo recomendable que, cuando se finalice éste, se informe a CBMEngine de ello mediante el servicio `finishProblem`. Con este servicio se fuerza la actualización del nivel de conocimiento del alumno.

7. Tras varios intentos sobre un problema y tras varios problemas realizados, el SE debería informar cuándo tiene lugar el fin de una sesión mediante el servicio `closeSession`.

Otra alternativa para proporcionar las evidencias del alumno es usar el servicio `storeEvidencesAndCheck` que realiza todos los pasos anteriores de una sola vez. Actualmente este servicio sólo permite proporcionar un único intento sobre un problema, aunque es posible extender el mecanismo de procesamiento. Tanto esta extensión, como el funcionamiento básico del servicio requiere un esfuerzo adicional al ser necesario procesar la información de la solución que se codifica en XML.

# Appendix B

# Summary in English

*Experience is what you get*
*when you don't get what you want*

Randy Paush (1960 - 2008)

**Summary:** For every chapter presented as part of this dissertation work, this appendix will sum up the most important aspects, trying to give a general overview of the work conducted. The conclusions chapter has been literally translated from Spanish since it contains all the contributions and open research lines.

## B.1. Introduction (Motivation and goals)

*Intelligent Learning Environments*(ITS) or, in general, Intelligent Learning Environments, are learning systems that usually combine Artificial Intelligence techniques and Psychology fundamentals in a computer system in order to provide the student with self-paced instructional environments. The basis of these systems is the student model which collects information about the learner such as his/her knowledge state in a particular domain. This model is updated with the information the system collects through the interaction with the student. Using the model and a specific learning strategy, the system usually adapts its content to the student's specific needs in order to act in the most effective way possible. For this reason, an accurate diagnosis influences the effectiveness and reliability of the learning process. The educational diagnosis is the process of gathering information in order to determine the student's knowledge, which, according to the degree of reliability and objectivity of the assessment, can be formal or informal.

Probably the most widely extended formal assessment instrument is the *Item Response Theory* (IRT), i.e. a well-founded psychometric theory focused on individual properties of questions, known as *items* in this environment. The central element of IRT is a density function, which relates the student's knowledge with the probability of answering an item correctly and is called *Item Characteristic Curve* (ICC). This data-driven curve allows computing the student's knowledge. Furthermore, the suitable use of this theory guarantees important properties such as invariance of measurement.

IRT is usually applied in testing environments where items play the role of measuring instruments. However, these items have an important limitation: they have been designed to measure knowledge associated mainly with theoretical concepts or facts, i.e. declarative knowledge. Accordingly, when the goal is to assess knowledge in procedural domains, e. g. those requiring solving a problem or making a more complex response, items seem not to be appropriate.

In the literature there can be found many proposals to approach the problem of tutoring the process of solving complex tasks. Probably one of the most popular and successful paradigms is the *Constraint-Based Modeling* (CBM), which has proved its efficiency through a number of studies in a wide range of areas of ITS. Its main strengths are simplicity of application, efficiency and flexibility in comparison to other existing approaches. However, the main problem of CBM is precisely the way in which the student's knowledge diagnosis is accomplished, since it is commonly done using heuristics.

The motivation of this thesis is to overcome the above-mentioned weaknesses. To this end, the main goal is to develop a probabilistic model that provides well-founded assessment mechanisms in procedural domains with the following characteristics: 1) it should be a formal model, aimed at producing a sounded judgment; 2) it has to be a quantitative model producing, thus, the judgment without considering subjective elements involved in the assessment process; 3) according to finality, it should be summative, i.e., produce a judgment. Regarding this last characteristic, it would be desirable for the model to be also formative. To this end the summative judgment should be extended with the feedback and required elements to guide learning.

The second main goal is to implement the model and evaluate it empirically in a real environment. To provide the student with a platform where he/she could solve complex tasks, a PSLE has been used. This term is normally used to refer to ITSs that collect evidence from the student through a problem solving process. Accordingly, the model will be particularized using the IRT as a formal assessment methodology and CBM as a PSLE paradigm. The proposal has been implemented in several CBM systems in which several experiments with real students have been conducted.

The third goal of this research is to generalize the proposal as much as possible. As a part of this, one of the products sought by this thesis is a framework that allows incorporating the proposed assessment model into an external PSLE. In this way, elicitation of new ITSs should focus only on developing the interface and communicating with the framework to get assessment in procedural domains.

## B.2. Background

### B.2.1. Problem solving learning environments

Chapter 2 mentions the general characteristics of PSLE, highlighting the cognitive tutors as one of the currently predominant paradigms to build these environments. The core of the chapter is an in-depth review of the CBM approach, the other predominant approach for student modeling in PSLE. Both techniques are not limited to student modeling, but they also set up a way to create other elements of an ITS. Thus, their use for building ITS is preferable to other student modeling techniques.

CBM is based on Ohlsson's theory of learning from performance errors (Ohlsson, 1992), according to which incomplete or incorrect student's knowledge can be used

within an ITS to provide guidance. This faulty knowledge is detected using constraints, which are the key element of CBM. Constraints are principles that must be followed by all correct solutions in the given instructional domain. If the student's solution violates any constraints, it is incorrect and the system provides the student with the appropriate feedback for remediation.

After an extensive review of the CBM paradigm, a number of studies demonstrated the CBM's efficiency as a modeling technique and instructional tool (Mitrovic et al., 2001, 2007; Mitrovic, 2012). These studies cover different areas of ITSs with satisfactory results in more mature research works, or promising ones in those that are still being developed. The main advantage of this approach is the ratio efficiency / simplicity it provides. Simplicity refers to the application of the technique, since it is reduced to define a set of constraints covering the knowledge to teach and the use of an inference engine to detect errors in the solutions the students build.

Comparing the two main modeling techniques, several similarities and differences can be clearly seen. First, the mechanism to provide immediate instruction, i.e. the *Model Tracing* (MT) in cognitive tutors and feedback in the CBM, does not differ in its application since both imply performing a pattern matching mechanism. Regarding the construction of the domain model, several works (Mitrovic et al., 2003) have shown that less effort is required in CBM than in cognitive tutors since encoding domain principles is easier than modeling all possible wrong steps, which could be a very expensive or even impossible to complete in ill-defined domains. From a technical point of view, the interface implementation and integration of the various components can be the most time-consuming part of CBM, unless some author tool is used. Nevertheless, any other technique requires the same or more effort in this task (Mitrovic et al., 2003). The main difference between the two techniques is the student model, which is used to guide instruction: while the MT in cognitive tutors uses a probabilistic approach, CBM systems employ heuristics.

Consequently, the most important limitation of CBM can be found in the mechanism that infers the student's knowledge, which is based on a heuristic function. This mechanism consists in determining the proportion of constraints the student knows. In turn, it is considered that a constraint is known if it is satisfied over 70 % of the times it has been relevant in a period of time, which is subjective because: 1) the threshold that determines whether a constraint is known could be lower, higher, or dynamic according to various student characteristics; and 2) the period of time is taken at the beginning or at the end of the interaction with the student, without being justified why that interval is the most appropriate. Ohlsson & Mitrovic (2006) already pointed out that, in order to improve pedagogical decision making in the CBM, it is necessary to have a long-term student model instead of the current model, which is short-term-like. This faulty component is similar to the *Knowledge Tracing* (KT) in cognitive tutors, with the inconvenience of lacking a well-founded basis in CBM.

The previous way to determine whether a student knows a constraint also affects adaptivity in CBM: estimates associated with the problem difficulty are performed based on the same principle by using the weighted sum of probability of the student having already learned each constraint relevant for the problem. This mechanism is a heuristic approximation of a problem difficulty and causes that, when it is used to select the next problem in a real environment, inadequate problems may be selected, either very simple or very complex ones (Mayo & Mitrovic, 2000). In addition, to determine whether a constraint is learnt, the weights used in the problem difficulty estimation are

determined using the number of predicates a constraint contains, which is neither an objective nor a formal reflection of the importance or influence a constraint may have on the problem difficulty.

In an attempt to correct this situation, objectives and well-founded techniques were used based on general theories of rationality (known as *normative theories*) by combining statistical decision theory (Savage, 1954) with Bayesian networks. Although these research works (Mayo & Mitrovic, 2000, 2001; Mitrovic et al., 2002) use a formal and effective approach, it has several drawbacks that makes it impractical in real systems. Firstly, the application of this probabilistic methodology is detrimental to one of its greatest advantages in its original formulation: ease of application and effort required, since it entails much more effort and time to implement the model. Furthermore, it has an important limitation in terms of model scalability due to the methodology, since it is only applicable to systems with a reduced number of constraints.

Some other sound techniques for determining a problem difficulty and diagnosing students are necessary to make the CBM paradigm able to provide an objective instructional process, and a well-founded and more efficient instruction. Primarily, it would be necessary to use other mechanisms to estimate the student's knowledge level, thereby strengthening the base on which the entire process of adaptation relies, which would include the adaptive selection of problems, presentation of suitable feedback, and any other kind of necessary pedagogical action.

Other minor limitations found during the review of the CBM paradigm lie in the use of semantic constraints, which requires a problem solver in some domains, as is the case of cognitive tutors. Besides, in order to achieve a better performance, highly diagnostically informative solutions are needed. These are solutions that provide information to perform the knowledge diagnosis. For example, a solution comprised of a single number is not highly informative since it does not provide more information about the principles involved in solving a problem. In domains with these solutions, resolution steps and *path constraints* are commonly used, and, as explained in Section 2.3.2, they correspond to production rules in cognitive tutors. As mentioned before, these are minor problems that do not reduce the flexibility, efficiency and applicability of CBM (Mitrovic, 2012).

### B.2.2. Assessment methods

Chapter 3 contains a review of assessment methods that have been developed to measure mental and psychological characteristics of people. In this sense, in the field of Psychometrics tests have been used as diagnosis tool. Test theories are a set of statistical and mathematical models that allow measurement of subjects'characteristics or traits, while providing methods to estimate accuracy as well. In order to use only terms being relevant for this thesis, henceforth it will be assumed the trait to be measured is the knowledge and the subject of a test is a student.

Two popular theories stand out from the rest: the *Classical Test Theory* (CTT), which focuses on studying the properties of tests, and the IRT, which focuses on studying the properties of items. The chapter gives a more detailed explanation about the IRT due to its relevance for this thesis. It has many advantages over the CTT, one of which being the independence of estimates with respect to the population from which they are obtained. This allows administering tests to different populations and maintaining the validity of the results.

The main element of the IRT is a curve that models the probability of responding

an item correctly given a certain student's knowledge: the ICC. Different ways of representing this curve lead to different IRT models. Among the different categorizations that can be made, we should highlight the characterization of the models by the mathematical function used to represent the curve, distinguishing between parametric and nonparametric models. The former is popular due to its easy handling, since the curve is represented by a set of parameters, while the second requires enumerating different values of the curve. Properties of IRT make this test theory one of the most popular methodologies of formal assessment. Its main inconvenience is that in order to apply it with a high degree of accuracy, a prior calibration process is necessary, which needs test data results from a student population large enough to ensure proper estimation of the ICC.

An important application of IRT is the *Computerized Adaptive Test* (CAT), which is a type of test that adapts dynamically the presentation of items to each student's particular needs. Adaptation is done by presenting items appropriate to their knowledge level, shortening the length to get the same or even better reliability than traditional tests. A CAT establishes how to orchestrate the use of IRT at each step of administering a test, starting with the calibration of the ICC. When items are calibrated, they can be used to: determine the students'knowledge, based on their responses; adaptively select the next item to be shown; and identify when the test may finish, for example, by checking that the knowledge estimate has a minimum accuracy level.

Since one of the main objectives of this thesis is to apply the formal assessment mechanism commonly used in testing environments to PSLE, this chapter also briefly reviews the characteristics of research fields related to this type of assessment. To this end, the *automatic scoring* and *cognitive diagnosis* fields have been mentioned. However, existing applications are specific to the domain in which they are applied, and they do not define a generic methodology to be applied in other domains. With the aim of finding this genericity, the *Evidence Centered Design* (ECD) framework has been studied and explained. It is a proposal that tries to extend assessment mechanisms to any kind of task.

The ECD framework establishes a set of guidelines and generic elements without specifying a particular domain of application, assessment methodology, or restricting the form of the elements involved. The main advantage of ECD is that it can be applied as a guide in the development of any system where evidence can be identified. Nevertheless, each new system where it is applied requires a different study with different structures, elements and methods. In short, it involves designing the student and domain models completely from scratch. That is why, if some of its genericity were sacrificed by using a PSLE such as CBM or cognitive tutors, the task of building a new system would be reduced to the construction of the domain model.

## B.3. Proposal

### B.3.1. Summative assessment in procedural domains

Chapter 4 proposes a theoretical model that allows making summative assessment of a student in PSLE. Accordingly, a student modeling technique and an assessment methodology are both required. As modeling paradigm, it uses the CBM due to its efficiency, proven through a wide range of studies, and its simplicity of use in comparison to other techniques. Regarding the assessment methodology, it uses the IRT since it is

a well-founded approach with several advantages over more traditional approaches.

The model seeks to achieve the initial goals of this thesis, trying to be a reliable and formal mechanism; features that are obtained thanks to the objective and well-founded nature of the IRT. At the same time, it seeks to be a quantitative model, leaving out of the assessment subjective elements proper to qualitative models. It also tries to be generic on several levels: a) it does not impose a specific model of IRT, providing necessary definitions and guidelines that make it independent of the IRT model; b) it is applicable to any domain, which is inherent to the CBM paradigm; and c) as far as possible, it gives a generalization guidelines on PSLE in a general way. Finally, the model developed allows a summative assessment. As explained in the corresponding chapter, this is the first step in designing a formative assessment methodology for learning.

Use of IRT in CBM systems is possible due to an analogy found between items and constraints. Both are measuring instruments of knowledge that, though in different environments, collect correct or incorrect evidence of knowledge. In addition, environments where they are applied (testing environments for items and CBM tutors for constraints), maintain a structural similarity that favors the use of the formal assessment methodology in the student modeling paradigm.

In order to define our assessment model formally, the typical response model in testing environments (Guzmán, 2005) is taken as a reference to be extended to CBM systems. This model is based on a typical evaluation function determining whether an item is correct or not. Similarly, we define a constraint evaluation function, but considering differences introduced with CBM systems: the main elements of the system, which are the constraints and problems in the domain model, instead of items and tests; the response, which is represented by a solution to a problem as opposed to the options in items; the particularity of constraints that may be relevant or not to a problem; and the determination of the correctness of constraints, which is evaluated using the satisfaction condition instead of checking whether it matches a correct response pattern. The response model formalism in CBM systems is completed with the definition of curves that model the probability of a constraint being a correct (*Constraint Characteristic Curve* (CCC)) or incorrect evidence (*Opposite Constraint Characteristic Curve* (OCCC)).

As part of the assessment methodology, it is necessary to estimate constraints curves. This calibration requires taking into account a special consideration related to the assumption of invariance of IRT, which means that there is no learning during the process of collecting evidence. This conflicts with the core philosophy of CBM, which tries to make students learn by providing feedback. To address this situation, two evidence collection methods for calibration are proposed in this thesis: either removing the feedback directly, or using only the first time a constraint is relevant, palliating any learning effect. Both methods will result in a set of evidence reflected in a *performance matrix*, which has been formally defined from a series of functions based on the previous response model. The matrix will have the resulting performance value for each student (row) / constraint (column), and will be used within a specific calibration method. In order to maintain genericity, this method and the IRT model used to represent the CCC, which are related, have not been used in this assessment model.

Once the CCCs have been calibrated, the student's knowledge estimate can be obtained by applying one of the IRT methods, based on the likelihood function, or using a Bayesian approach. As the basis for these methods the likelihood function for

constraints has been formalized. This is a density function of knowledge that allows determining the most probable value of the student's knowledge by applying some estimator mechanism. The likelihood function is defined based on the combination of the probability functions of constraints that provide evidence about knowledge. As any IRT assessment model, the combination uses the CCC or the OCCC if the evidence is correct or incorrect, respectively. The expression of the function is done with a function to determine the relevance of a constraint, and using the content of the *performance matrix*. Since learning effect should be discarded, this matrix follows the same definition given in the calibration stage. Although the likelihood function is the basis for this method, and it doesn't modify its shape, the knowledge estimation method is specific to the IRT model. Therefore, it is let out of the model for genericity purposes.

Finally, we try to generalize, as far as possible, the application of the summative assessment model to any PSLE, regardless of the paradigm used for student modeling. The generalization is based, as in the ECD framework, on evidence gathered from the student, which serves as a knowledge measuring instrument, provided it meets the assumptions of application of the IRT. This generalization focuses on defining a matrix of performance that will be provided by the student modeling paradigm. This matrix will be used in the calibration and assessment. The first process, which depends of a model of IRT and a specific calibration method, is outside the model. The assessment however, would be made based on the characteristic curves and performance matrix, also generating a likelihood function that is used to estimate the student's knowledge in the same way as in the CBM + IRT proposal.

In many of the articles and studies related to the CBM, it is mentioned that constraints represent declarative knowledge (Mitrovic, 2012), but this does not mean that the assessment performed using constraints is a declarative knowledge assessment. Although constraints are typically associated with declarative knowledge, the student's solution is the result of applying both declarative and procedural knowledge, since the latter is necessary to solve the problems in procedural environments. Likewise, domain principles are not always associated with concepts, as mentioned in (Mitrovic & Ohlsson, 2006), which becomes evident after using the *path constraints* that represent procedural knowledge.

According to the previous paragraph, the result of constraints, either satisfaction or violation, is not the exclusive reflection of declarative or procedural knowledge, but the result of a mixture of both. In fact, even the relationship and dependence between the two types of knowledge remains an open question (Schneider & Stern, 2010). Thus, the constraints are a measuring instrument of knowledge in a general dimension (de Jong & Ferguson-Hessler, 1996), since they are unable to distinguish how much declarative or procedural knowledge is measured. In addition, as Mitrovic & Ohlsson (2007) pointed out, learning, measured in terms of constraints, follows a similar pattern as cognitive tutors in terms of production rules in cognitive tutors, whose a nature is closer to procedural knowledge.

### B.3.2. Formative assessment model

In this chapter the summative assessment model outlined previously has been extended with the elements that allow a formative assessment based on well-founded methods. These elements, which in CBM are directly related to feedback and next problem adaptation, have been extended with IRT. Given the importance of adaptation

in CBM, mechanisms of CAT are desirable for these tasks.

The chapter begins by reviewing the application of the proposed methodology on testing environments. The aim is not only to show that the model can be applied to overcome the limitation of these systems, but also to study the implications of using the IRT for problem selection in their original environment. Thus, considerations to be taken into account will become clearer than if they were made directly in CBM systems.

As a first step to incorporate complex tasks in testing environments and to study adaptation capabilities in problems selection, a new type of items, called *composed items*, have been designed. These items group a series of component items, similar to testlets (Rosenbaum, 1988). Component items can be real, i.e. they are directly associated with questions; or virtual, if they are modeling evidence from a complex task. Composed items have also an associated curve representing the probability of answering correctly given a certain value of knowledge. In a similar way to traditional ICCs, it is called *Composed Item Characteristic Curve* (CICC), and it is calculated using the ICC from its components. These curves allow applying CAT mechanisms in order to select composed items adaptively.

The use of IRT to perform adaptation in CBM systems is possible due to the analogy identified in the previous chapter, which is extended with composed items. In this way, constraints are equivalent to component items and problems are equivalent to composed items, which are the subject of the adaptation. In order to apply IRT mechanisms, an extension of the CBM classical structure is required. In the domain model, each constraint is extended with a CCC and the problems with a *Problem Characteristic Curve* (PCC), equivalent to the CICC in testing environments. Furthermore, a conceptual grouping is introduced, similarly to existing works made in some CBM systems, which groups constraints and problems to form new instructional strategies. In the student model, the heuristic estimate of the student's level is replaced by the estimates of knowledge provided by IRT for each of the concepts in the domain model. The third extended component is the pedagogical module, which incorporates instructional strategies based on the previous extended elements.

Calibration of ICCs in IRT has to be made using performance data from students who have previously solved these items. Moreover, in those datasets it is assumed that no student learning happens. However, CBM systems are mainly used for learning purposes, which makes the process of CCC calibration very difficult. To overcome this problem, this thesis proposes a mechanism to track knowledge in CBM systems by introducing the concept of CK-sessions. A CK-session is a group of evidence where student's knowledge has no significant change. Thus, summative assessment can be applied to obtain student's knowledge at different points in time. This new mechanism offers a way to build the performance matrix to apply the summative assessment. Furthermore, several possible methods to perform the grouping of evidence are proposed, such as using a threshold to discriminate the elapsed time between sessions, or grouping by time intervals where to have knowledge estimation is desired.

The advantage of this mechanism is that it allows calibration of constraints using data where learning has occurred. Nevertheless, if we want to use it "on the fly" while the student is using the system, a problem occurs. Namely, the mechanism is not able to model learning gain and, therefore, it is not accurate. In order to solve this, the utilization of IRT models dealing with learning is proposed, such as the one recommended by Lee et al. (2008). Nevertheless, these models are barely found in the literature,

and seem to need further investigation in order to achieve certain degree of maturity. Thus, their application and study are out of the scope of this thesis, which covers only traditional IRT models. Thus, the rest of the formative proposal assumes that an IRT model with learning is being used and that it is based on the ICCs, as any traditional IRT model.

To perform the student's formation, it is proposed to use formative assessment that will be driven by learning goals, which will be evaluated in a final summative assessment. To achieve these goals, a generalization over the domain model elements is used: the constraints and problems. This generalization groups constraints in concepts and problems in types on which a teacher can establish a minimum knowledge to be met. Based on the evidence gathered for each concept, the associated knowledge can be estimated, which allows assessing the achievement of learning goals.

At the core of the proposed model the implementation of the IRT mechanisms is performed on the elements involved in adaptation: problems selection and feedback. Each form of adaptation performed in the CBM has a counterpart method in the combined CBM + IRT model, with the substantial difference of being well-founded. As instructional strategies, a set of methods for adaptively selecting both feedback and the next problem have been proposed. The selection can be made based on various criteria that can be driven by learning goals, or the way as it is traditionally made in the IRT. In addition, an important part of the feedback the student should receive in order to know what needs to be improved is the open model, whose learning objectives and the level of achievement are displayed.

Based on the different formative elements two major functioning modes are proposed, which seek to provide or to avoid learning. Use of the system without learning is necessary to calibrate constraints or make the final summative assessment, while use of learning is associated with a formative assessment. Each mode of use is characterized by employing the most appropriate adaptive selection mechanisms, in conjunction with a form of grouping into CK-sessions, and a method of gathering evidence. A drawback of the proposal is that, to maintain security, it should have a larger number of different problems, so that the problems involved in learning are not presented again in the final summative assessment phase. Besides, the formative mode is still to be developed using an IRT model able to model learning.

Efficiency in many of the proposed mechanisms still needs to be proven. In this sense, the main limitation of the proposal in the formative aspect is that it is unknown which of the proposed instructional strategies are most effective in a real environment. Since CBM heuristic have been replaced by a well-founded probabilistic model, our hypothesis is that the behavior of the system will be more effective with this methodology. However, future research is needed to demonstrate the degree of compliance with the hypothesis.

IRT is not only useful in adaptive selection and assessment. It can also be used to determine the suitability of the constraints as a knowledge measuring instrument. We propose to use the information function over constraints. In this way, we can identify those constraints that do not have enough evidence, have been incorrectly coded, or represent conceptually inadequate principles. This function can be applied as an analysis tool in the domain model to filter out inappropriate constraints and generate a more reliable assessment. Furthermore, the technique can also be used to determine unhelpful problems from the point of view of the information provided.

According to the ECD framework, which has been mentioned in this summary in Section B.2.2 and presented in detail in Section 3.4.2, the proposed methodology is a

particular case of this framework, both in operation and architecture. Our proposal reduces the level of genericity, since it is applied only to PSLE, but it is not completely lost, since it can be applied to multiple domains. The advantage of using the CBM is that it establishes more specific guidelines in the construction of new systems, avoiding studying what will be the student model every time, or determining what constitutes evidence, as it is already implicit in the use of this methodology.

According to the ECD we can establish a specific correspondence between every ECD framework component and our proposal: the student model would be the traditional CBM student model, along with the conceptual grouping based on knowledge estimates produced by IRT. Within the evidence model, the evidence rules would consist of checking violated and satisfied constraints. The state model would be the union of the CK-sessions method to group evidence and IRT mechanisms to update the knowledge in the student model. Within the model of tasks, tasks, represented by problems, would have the PCC as representative characteristics, and work products would be the solutions built. The presentation model corresponds to the PSLE interface and the assembly model is the PSLE itself. As part of this thesis there has also been designed a system that combines all these features, as the assembly model does, which is part of the Implementation Chapter. The equivalence with the architecture of four processes is the same as explained in the example of Section 3.4.2.2, but using well-founded techniques of IRT instead of heuristics.

As can be seen, there is a huge number of combinations derived from mechanisms and associated with formative strategies and grouping in CK-sessions. This, together with the existence of other mechanisms that have not been identified yet and are probably even more effective than the above, opens up a universe of possibilities yet to be explored.

## B.4. Implemented tools

The corresponding chapter explains the tools that have been used to implement the theoretical assessment model of CBM with IRT. As for the work methodology, planned at the beginning of the thesis, it has been followed a bottom-up approach has been followed. According to it, specific tools were built first to explore the assumptions of this thesis empirically, and, once these assumptions were verified, common components and elements were combined into a more generic approach to develop more generic systems.

The first tool developed was *OOPS* (*Object-Oriented Programming System*), a PSLE in the domain of Object-Oriented programming, which is well-defined and has ill-defined tasks. The domain model is comprised of a total of 86 constraints, which are checked in a JESS inference engine (Friedman-Hill, 1997). The system incorporated some initial heuristics for updating the student model with the estimated level. Problem selection was originally also adaptive and heuristic-based using a method similar to the maximum likelihood of IRT.

Subsequently, *Simplex Tutor* was developed. It incorporated significant differences, mainly due to the technology used, which made the system development easier. It is focused on the domain of lineal optimization, using the Simplex and Two-phases algorithms, which is well-defined with also well-defined tasks and consists of a reduced set of 17 constraints. These constraints are checked in a JBoss Rules inference engine (Bali,

2009), which allows the direct use of Java objects. This makes its use easier, gives more power and produces a more efficient inference process than OOPS. The development allowed us to abstract patterns and common elements in the architecture of PSLE, which subsequently led to CBMEngine, which will be mentioned later.

Regarding the theoretical model for assessment in procedural domains using testing environments, the above-mentioned composed items have been implemented into the Siette system. As pointed out before in this summary, these items group several pieces of evidence, each one modeled as a component item. Besides its use to model constraints and problems, component items might be any type of items such as multiple choice or short answer.

Furthermore, the author of this thesis collaborated with the authors of SQL-Tutor (probably the most popular CBM-based system). The performance data of learners who used this system were treated as a source of data to experiment and improve our methodology. The relevant part of the tool we used refers to the student model that collects the activity in the system and can be used as evidence to apply our models. In order to process this information a new tool was developed: *SQL-Tutor Processor*, which applies our assessment techniques a posteriori and independently from SQL-Tutor. It uses existing information to form the performance matrix, format the data, and invoke MULTILOG software, which is responsible for carrying out IRT mechanisms. The number of options the tool possesses allows customizing every stage of the calibration and assessment process. Although SQL-Tutor Processor is designed to process data belonging to SQL-Tutor, it can be extended to any CBM tool that maintains a similar format in its activity log file. If this file is much different, the tool has reusable components, it can be reused by only replacing the file content processor component.

Drawing from the experience gathered in previous systems and generalizing their structure and models, *CBMEngine* was developed: a framework that provides external PSLE with our assessment model through a set of Web services. When using CBMEngine, an external PSLE only has to define the structures and rules involved in the domain model, which are used together in a JBoss Rules engine to perform the CBM reasoning. The student model is part of the system and is updated with the evidence the external systems provide. The assessment and adaptation using the IRT are part of the services provided, which, internally use the functionalities of other systems such as MULTILOG or Siette. The platform was integrated into Visual Nets and PIPSLE, two systems developed under the DEDALO platform which aims to provide assessment services and student modeling for building PSLE. CBMEngine is comparable to WETAS, one of the systems developed by Tanja Mitrovic's group. WETAS provides an environment on which the modeling and assessment techniques for ITS can be run. Nevertheless, CBMEngine uses formal mechanisms and has a tool, which will be mentioned afterwards, for authoring the domain (WETAS has no support for this task). Although CBMEngine is probably at the same level as WETAS, it still needs to extend many components in order to achieve the maturity of ASPIRE, the most important CBM authoring tool.

Finally, CBM-DoME (*Constraint-Based Modeling Domain Model Editor*) was developed to help constructing structures and rules that CBMEngine uses. The tool is an extension that, to date, allows the creation of simple structures and rules over them, being necessary to extend its functionality to reach the full potential that would be achieved using Drools in its native form at constraints encoding. This framework has not been used yet to build any system but it will be incorporated as a component in

CBMEngine framework. CBM-DoME as an authoring tool still lags far behind the tool ASPIRE since the functionality provided includes only the creation of two components of the domain. Besides, the process is still rudimentary compared to ASPIRE, which discovers constraints using a semi-automatic process and allows building a tutor almost entirely. However, it is a relatively young tool and our future work is intended to cover some of these differences.

Tools used (i.e. Siette, MULTILOG, CBMEngine, and CBM-Dome) perform as pieces of a puzzle which properly fitted allow firstly the utilization of the elements required to use the probabilistic model defined in previous chapters, and secondly its use to carry out assessment and adaptive content delivery.

## B.5. Experimentation

Following the bottom-up approach, originally planned as a working methodology, experiments have initially studied specific aspects of the methodology and later have attempted to cover broader aspects. The first experiment explored the feasibility and efficiency of CBM as a student modeling technique and instructional methodology. An experiment conducted with an experimental group compared the use of OOPS with a control group that did not use it. The results suggested that the technique significantly improved learning, increasing the number of students who passed from 48 % to 81 %.

Further experiments extended the study applying a summative assessment in domains of different nature. In simple domains, as the underlying for Simplex Tutor, as well as in complex domains, such as OOPS, the applicability of the methodology is clearly successful since these systems can use the methodology as part of their normal operation. Our experience with the systems and experiments served to review and refine the models, which were extended in the framework CBMEngine where applicability is even clearer.

In a first step, it has been tested whether assessment produced by our systems is comparable to formal assessment provided by a test. The results show that there is no significant statistical difference and the assessment produced using a problem requires less time and effort to be produced (answering all questions in a test is longer than solving a problem). In order to check other statistical properties, we studied the correlation between the two types of assessments, which was not excessively high. This does not mean that the systems are not providing proper assessment. What we think may be happening is a combination of factors influencing these results. First, studies have very few students, which can make the assessment easily affected by external factors that introduce noise in the results. Second, a test requires a declarative response, mostly to theoretical concepts; while in the systems, the resolution process is procedural in nature. Thus, while a test explicitly asks the concept, in a PSLE it is implicit in the resolution and the procedural knowledge is involved, which could be an influencing factor.

In experiments conducted on formative assessment, we studied the use of grouping in CK-sessions for calibration, trying various forms of grouping sessions. The results show that the grouping method using a TCK threshold is not a good option. Instead, the most beneficial one is grouping constraints by problems, which stands out from the other methods studied. However, it would still be desirable to explore other methods, such as the ones proposed in Section 5.2.2.3.

The invariance of the model has been studied in several experiments. The results, though promising, are not significant or conclusive due to the evidence available. In this sense, the evidence obtained from real students is not adequate, either because the number of students is too small or because the evidence collected is not sufficient. Therefore, extending this study, when a suitable data set is available, is necessary.

Regarding the study performed using the information function as a tool for determining the quality of constraints, the goal was to use an automatic mechanism to filter constraints that were not relevant or had little evidence. However, besides being useful in this regard, we discovered other uses of the function to study the quality of a CBM system domain model. Although the study used only 7 constraints, had very important and promising results that reveal the utility of the information function to detect constraints that are reflecting incorrect domain principles.

Although the applicability and feasibility of the assessment mechanisms has been tested, many other features are still pending, mainly due to the lack of real students. Although the validity study has focused on a particular type of branch of psychometrics (validity of the construct), there are many other types of validity that may be studied (Moss et al., 2006); other ways to check the reliability (Cook & Beckman, 2006); and various indices to study the accuracy and consistency provided by the IRT (Wyse & Hao, 2012). The aim of the empirical study in this thesis has tried to cover both summative and formative assessment, which has let to a study without very much deep. Since, from a psychometric point of view, it is interesting to study the mentioned properties in our assessment methodology, it has been proposed to analyse them in the future.

## B.6. Conclusions

Dr. Stellan Ohlsson, whose publications have been a landmark in the work done in this thesis, gave a speech at the *15th International Conference on Artificial Intelligence in Education* (Auckland, new Zealand June-July 2011), in which he presented an analogy between current research in the field of AI in Education and that conducted in the past in the field of aeronautics. The analogy is located in the efforts that were made in the past to achieve the goal of flying. Many planes were built and many attempts were made to stay aloft. However, not until various physical and aerodynamics principles were combined was it possible to overcome this barrier. Similarly, current research in AI in education has resulted in many planes or tutoring systems that seek to improve learning usually focusing on a single principle. As happened with aeronautics, future ITS should combine different principles of learning in the same system in order to maximize the effectiveness of educational tools.

A similar idea was defended by another of the most important authors for this thesis, Dr. Antonija Mitrovic, at the *11th International Conference on Intelligent Tutoring Systems* (Chania, Greece, June 2012). In a special panel about the future of Intelligent Tutoring Systems, in which the greatest experts in this field participated, Mitrovic showed the resemblance of current research in the field to islands of knowledge. Each island corresponds to a different research field where different members of the scientific community are working in. Thus, each field is usually isolated in the sense that the studies are performed on a specific topic, regardless of what is done on other islands. For the future, Mitrovic stated the need to join forces and combine principles in each

field to make a significant improvement in ITS.

The conducted research follows the philosophy of the two previous ideas, as it has approached two fields that were previously isolated from each other: a large island represented by the CBM student modeling paradigm, and the other represented by IRT formal assessment techniques. Both mentioned paradigms have been combined into a model that attempts to solve a twofold problem:

- First, the formal assessment methodologies, such as IRT, which are applied through tests, have not been designed at first to assess knowledge in tasks with a complex solving process, i.e. in procedural domains. Existing interactions in this line are limited, by the form of items, to simple tasks such as sorting or matching items. Therefore, PSLE is needed in order to allow students to carry out the desired complex interaction.
- Within existing PSLE paradigms, there are no mechanisms to formally determine the student's knowledge. Among these paradigms, CBM is presented as one of the easiest and most efficient alternatives. However, it lacks a well-founded assessment mechanism since estimates of what the student knows are based on heuristics.

This thesis seeks to solve the above problems by combining the two paradigms in such a way that complementary characteristics are exploited to form a beneficial model for both fields. Thus, formal assessment has been extended to be applied to procedural domains where the complexity of the tasks is not limited by the form of the items. At the same time, the PSLE in general, and CBM in particular are enriched with a new way to diagnose and to model the student.

The content of this chapter provides a compilation of the presented work in the following way: next section summarizes contributions of this thesis to the different disciplines employed. Then, in Section B.6.2, identified limitations in the proposed model are outlined. Finally, the research lines opened up by this work are presented, which in turn account for the future work plan.

### B.6.1. Contributions

The research carried out in this thesis contributes to the development of AI in education and educational assessment fields according to five major groups, which are enumerated below:

1. *Contributions to formal assessment*: A methodology has been defined for assessment in procedural domains by using evidence gathered through the interaction with the student. The most important features of the methodology are:
    - The methodology is systematic and well-founded. This is because the student's knowledge assessment is provided by the IRT. The use of assessment mechanisms of this technique, which are noted for their objectivity and reliability, make the judgment, in turn, also objective.
    - A quantitative assessment model, which unlike the qualitative models ignores the subjective elements that surround the assessment process of making the assessment of the student. Thereby, objectivity of the process is pursued.
    - The assessment methodology is summative by nature, providing a judgment of the student's knowledge. This is the key element in order to perform

a formative assessment of the student, which is seen in the next group of contributions.

- Although the basis of the methodology is the use of evidence to perform assessment in a general form, the model has been particularized by developing a response model based on the IRT that is applied to PSLE by CBM. Using an identified analogy between items in testing environments and constraints in the before-mentioned paradigm, a model has been formalized using constraints as a measuring instrument. The mechanisms of IRT can be applied to constraints as if they were items to estimate the student's knowledge using evidence collected from in procedural domains.
- The proposed methodology is intended to maintain as much genericity as possible. With this objective, the response model is independent of IRT model used and focuses on the use of the CCC, similar to the ICC in testing environments. The proposed model is independent of the domain where it is applied, which is a characteristic inherited from CBM. Besides, the model is generalizable to other types of PSLE where the characteristic curves can be used over evidence. In this sense, the source of evidence should meet the assumptions required by IRT to be applicable.
- Since PSLE is a type of ITS, its main objective is to improve the student's learning process. This conflicts with some of the assumptions that allow applying the IRT mechanisms. To solve this, as part of the methodology we propose two *evidence collection methods* that discard the effect of learning in these systems, allowing the application of IRT.
- The assessment mechanism can be applied adaptively in a PSLE in an analogous way to a CAT. However, instead of items, problems are selected adaptively in terms of the student's knowledge level. To do this, a new type of items has been defined in testing environments; the so-called *composed items*. They are similar to testlets since they group evidence but they are designed to model more complex sources of evidence. For example, composed items allow to model CBM problems, and their components model CBM constraints. Like any other items, they have a characteristic curve, called CICC, which is obtained from ICC of the component items. Based on the CICC, the basic IRT mechanisms of adaptive selection have been formalized to select composed items.
- The assessment process is much shorter than in testing environments. This is so because during the resolution of a problem, involved concepts are not asked about directly, but they are inherent to the solving process. Thus, the evidence can be gathered in a shorter time and using a few problems instead of a test consisting of a multitude of items associated with each concept involved.

The use of PSLE as a basis for the proposed methodology allows overcoming the main limitation of formal assessment in testing environments. In this sense, the novelty of the approach is that it extends knowledge assessment from previous systems with complex tasks where the students must perform a solving process. The nature of the tasks in which it can be applied is different from the typical single-answer or multiple choice items, and goes far beyond the more complex

interactions that exist up to date in testing environments, which deal with simple tasks such as sorting elements.

2. *Contributions to student's diagnosis in PSLE*: The above methodology as well as extending the scope of testing environments, allows to determine the student's knowledge in CBM systems and, more generally, in PSLE. The particular characteristics regarding these systems are:

   - The estimate of the student's knowledge is performed using a well-founded methodology. This is an advance in the CBM student modeling paradigm since, so far, most of the existing mechanisms for this task are based on heuristics. We propose an extension of the basic structure of the CBM in the following: the domain model, with problems and constraints curves; the student model, with well-founded estimate of the knowledge; and the pedagogical module, with the logic to handle the previous models.
   - Although there are some well-founded mechanisms for student modeling in the CBM using Bayesian networks, they have the limitation that they can only be applied to systems with a reduced domain model. The proposed assessment methodology, however, is applicable to systems with a large number of constraints. As an example of the scalability of the system, in experiments conducted with one of the most extensive and important systems, such as SQL-Tutor, with a domain model over 700 constraints, there could not be detected any effect on the efficiency, or if there was any, it was negligible.
   - Unlike any existing approach within the CBM and most of the existing methodologies in the field of PSLE, the proposal of this thesis is based on the student's knowledge inference rather than an estimate of learning. This feature confers to the system the ability to be used as an assessment tool, which others do not possess. More than an advantage over other methodologies, this is an added value since the goal of PSLE is to promote learning, not to assess.
   - Another feature inherent to the use of CBM is that the methodology can be applied to procedural domains where tasks are ill-defined, i.e. those where, in order to solve a problem, there are countless ways to reach a correct solution. In cognitive tutors, the other important paradigm within ITS, domain modeling in these domains becomes practically impossible since it requires modeling each step the student can take to solve a problem.

3. *Contributions to instruct students in the CBM*: The previous summative assessment methodology has been extended to define a formative assessment student model in CBM systems. Thus, the instructional methodology is characterized by:

   - According to the work of the most important authors in psychometrics (see Section 1.1.1), formative assessment has been designed as a summative assessment plus a feedback that guides the student instruction before some learning objectives will be assessed in a final summative assessment. This idea extends the CBM domain model using a conceptual grouping where learning objectives to be assessed in the final summative assessment can be established. Similarly, and inspired by the philosophy of criterion-referenced tests, we propose the extension of CBM with student model that reflects

what the student knows about the previous abstraction. This grouping is proposed in both, problems being grouped into types or categories, and constraints being grouped on domain concepts.

- The summative methodology itself only allows assessing at specific times or considering elements where learning has not yet occurred. This leaves out important information about the evolution of knowledge that can be used by the system to guide instruction. In order to alleviate this problem the summative assessment model is extended with a new concept called *CK-session*. It consists of grouping a student's evidence in time periods to perform knowledge tracing in CBM systems. This is equivalent to the KT of cognitive tutors. To separate the CK-sessions where assessment of student is conducted, five different criteria are proposed, some are oriented to an evaluative use of the system and others have an instructive character.
- Based on the above conceptual grouping, an assessment algorithm is formalized combining the evidence-collection mechanisms of the summative assessment methodology with the knowledge tracing. This algorithm allows determining at any time the student's knowledge related to each component of the grouping, which is associated with the degree of achievement of learning objectives and helps to direct instructional strategy.
- The instructional capacity of our approach comes from the main instrument of CBM that guides students'learning: adaptation. The proposed method redefines the way it is carried out by replacing heuristics by well-founded mechanisms based on IRT. Thus, adaptation is applied in two different ways:
  - By selecting the following problem: To this end, the selection mechanisms of IRT based on the PCC (equivalent to the CICC) can be used. However, student's learning is not contemplated as an objective in the IRT and, therefore, it is necessary to define new strategies that include this goal. In this sense we propose three groups of strategies:
    - Basic strategies, which don't take into account the learning objectives. Within this group 8 different strategies are proposed, some learning-oriented and other assessment-oriented.
    - Strategies driven by one or multiple specific objectives that are being covered. They use one of the previous basic strategies to select a problem but restricting its scope to the objectives set.
    - General strategies, in which there is no predefined goal and they seek to make students improve their learning in general. We propose two criteria associated with different strategies: the first uses a degree of compliance with the learning objectives and the second uses a new type of characteristic curve defined on the components of the conceptual grouping. In any case, they first select an objective using either of these criteria and then apply any of the strategies driven by this objective.
  - Through feedback: In CBM, when several instances of feedback need to be presented, it is necessary to determine which one to show first. The ordering mechanism is another form of adaptation. To replace the original adaptation procedure, also based on heuristics, we propose three

techniques using the assessment model to determine the most appropriate feedback for the student. The first uses the CCC to this end; the second is based on the use of the conceptual grouping and learning objectives; and the third uses feedback directly associated with concepts rather than constraints. The latter is inspired by similar mechanisms already existing in the CBM but they differ on the foundation provided by the IRT.

These forms of adaptation, jointly with an open learner model which is shown using the knowledge estimates of the conceptual grouping, serve as feedback to complement summative assessment to produce the formative assessment of the student.

- We propose three recommended operation modes that any CBM system using our formative assessment model should implement. Depending on its purpose, each mode implies an appropriate evidence collection method, a grouping methodology in CK-sessions, and an adaptation strategy in line with the operation mode. The three modes have a different purpose: the curves calibration process, which seeks to obtain the CCC required by the summative assessment process; the final summative assessment, in which the main objective is to assess; and formative assessment, whose focus is to make the student learn.

4. *Contributions to construction of PSLE*: Based on the assessment methodology that combines CBM with IRT, the following facilities to construct PSLE are provided:

   - Model generalization: Using the summative assessment methodology, some general guidelines and requirements are given, which would allow applying the summative assessment of the IRT to other PSLE that are not built upon the CBM but can use evidence as a basis for implementing the IRT mechanisms.
   - Using the analogy between testing environments and CBM systems, a technique to study the quality of constraints for assessment has been proposed. This technique consists in using existing IRT tools to analyze quality of items, but applied to CBM. The method uses the information function to detect different error conditions in an existing domain model: constraints incorrectly coded, constraints correctly coded but modeling domain principles with an inappropriate level of generality, and constraints correct but without enough evidence. Regarding the level of generality, constraints can be grouping several domain principles or they might be too specific, in which case it would be desirable to split or group them, respectively. Thus, the technique can be used during the construction to enhance the elements of the domain model of a PSLE using our methodology.
   - In order to facilitate the construction of a future PSLE with the proposed methodology, the *CBMEngine* framework has been developed. It offers assessment services to be used in external tutoring systems. Thus, the framework is a reusable component that any PSLE under the CBM paradigm can use, provided that they register their domain model and pass the necessary

information to the framework. To construct the domain model in the framework it has been implemented a visual authoring tool called CBM-DoME has been implemented. The tool is still being developed but already allows editing constraints and structures necessary to apply error detection mechanisms of CBM. CBMEngine has been used to provide assessment to two tutoring systems: Visual Nets, which is focused on communications networking concepts, and PIPSE, which deals with investment project management.

5. *Contributions from the implementation point of view*: Besides the aforementioned systems (CBMEngine and CBM-Dome) and its integration with external systems, previous assessment models have been implemented in several tutoring systems that can be used to assess / teach in the following areas:

   - The OOPS tutor allows to teach students the basics of object-oriented programming.
   - The Simplex tutor focuses on the domain of linear optimization using the Simplex and the Two-Phases algorithms.
   - Another tool that is implemented is SQLTutor Log Processor. Although it is specifically designed to process data from SQL-Tutor, it can be easily reused almost entirely to process data from other tutoring systems and generate a posteriori summative assessment.

### B.6.2. Limitations

Although the proposed combination between IRT and CBM provides a solution to the limitations detected in both paradigms, the proposed model also has its limitations. These are listed below:

- A drawback of this approach is inherited from the IRT and is located in the calibration required to apply the methodology. Since the reliability of the assessment depends on the correct estimation of the model, a high student population is needed in order to make the sample sufficiently representative. The higher the number of constraints, the greater this population should be to ensure that there is enough evidence over the whole set of them.

- The degree of formality and objectivity of the methodology depends on the quality of every constraint of the domain model. That is why, if an assessment with a high degree of objectivity and reliability is desired, a thorough study of each constraint is required to ensure that it reflects an appropriate principle and has no anomalies. This process involves a study that is not limited to the application of the method proposed in this thesis to study the quality of constraints after calibration, but it extends to the entire application process, using tools such as those proposed in Section 5.4.3 to ensure the quality of the domain model. For this reason, the application of this methodology to certify a student's knowledge level in a rigorous manner requires an exhaustive analysis.

- Another limitation, inherited from CBM, is found when applying the proposed methodology to domains where problems solutions are not highly informative diagnostically. This means that the solution itself does not provide sufficient information to diagnose the student. In such domains, it is necessary to use the

path constraints introduced by Mitrovic & Ohlsson (2006) and explained in Section 2.3.2, which are equivalent to production rules in cognitive tutors. In this case, the local independence assumption cannot be satisfied by constraints since two rules associated with a joint resolution path may not be independent events. Further research is needed on this issue.

- If the CBMEngine framework is used to assess and it uses testing environments mechanism, it is necessary to maintain a redundant domain and student model in the two systems. First, under CBMEngine to gather evidence and apply the constraints, and, simultaneously, in the testing environment in order to apply the inference mechanisms of the IRT. Currently, there is no mechanism to create the domain automatically in the testing environment from a domain model stored in CBMEngine. This implies the building process must be performed manually, resulting in a tedious task.

- Another limitation of the proposal is that in order to increase the system security associated with the problems that are used in the final summative assessment it would be required to maintain a high number of problems. This is because it should be ensured that problems presented during the formation stage do not compromise the final assessment stage, being necessary to maintain a set of problems specifically for assessment and others to be applied in formation.

- There is a need to conduct an empirical study to determine how effective the proposal is as an instructional tool. Such a study extends beyond the scope of this thesis, but has been included within the open research lines. However it is mentioned here to note that, in the state the current proposal is, its educational efficiency is unknown.

- Although the implemented tutoring systems, OOPS and Simplex Tutor allow student assessment using tools such as MULTILOG, the formative assessment mechanism is not integrated as part of them. This is because these tools have been used primarily to study the summative assessment methodology. Another feature that can be considered a limitation is the type of feedback provided by these systems, which covers only two of the several types proposed within CBM, explained in Section 2.3.4. Furthermore, the OOPS system is defined to teach a language that is no longer used and, therefore, nowadays it cannot be used any more.

### B.6.3. Open research lines

According to Dr. Mitrovic's metaphor that opened this chapter, the proposed combination establishes a communication bridge that becomes beneficial for both islands of knowledge involved in this thesis. Given the dimension of what still remains to be investigated, it could be said that we have just built the bridge. However, traffic expected with this communication will extend, through a huge range of possibilities, the efficiency of AI in education.

The lines that still remain open, and where the future work is headed, have been pointed out throughout this document and especially in Chapter 5, where the formative model to improve student learning is presented. These lines are:

- Different calibration methods emerged as an idea in the final stage of this thesis, when experiments had already been conducted, as a result of the union of various criteria that led to the operation modes presented in Section 5.3.5. For this reason, it is also required to study the effectiveness of these three strategies in relation to the quality of model fit.

- In order to take into account the student's changing knowledge and trace it, the CK-sessions mechanism has been proposed, which operates according to various criteria (see Section 5.2.2.3). In our experiments we have studied one of these criteria, however, it is necessary to consider the others and determine which one is the most appropriate according to the needs of the assessment being made, either final summative or formative.

- In all the conducted studies parametric models of IRT have been used due to their simple implementation and because the constraints are equivalent to items with only two answers, which makes these models more than sufficient for an initial study. However, it is necessary to compare various aspects of parametric models against non-parametric ones in their use for assessment of knowledge in procedural domains, such as the quality of the calibration fit, efficiency, effect on the implementation, etc. Besides, it would be interesting to study the use of multidimensional models that would allow extending the conceptual grouping of constraints to a hierarchical structure with several levels of groupings.

- Theoretically, the result of the first time a constraint is relevant reflects the knowledge of student avoiding the learning provided by the feedback. However, it is also reasonable to think that other methods may be equally or more effective if they take into account the evidence that this method discards. For example, if the student satisfies a constraint the first time that it is relevant and then violates it three consecutive times, it is more logical to consider the violation than satisfaction. In this sense, the proposed methods do not use this additional information that could make the model more effective. This type of mechanisms, as pointed out in Section B.3.2 of this appendix, is not suitable for formative purposes. Therefore, it should be studied only as an alternative for summative assessment or constraints calibration.

- A slight modification that can influence making the judgment in the final summative assessment phase is to check the hypothesis which states that, at this phase, it would be advisable to use the lowest level of feedback. This type of feedback should be compared with the option of avoiding feedback in order to determine which approach is the most effective for this purpose.

- One of the great tasks remaining to be done is the study of the formative part of the proposal: firstly, the utilization of IRT models dealing with learning, in order to provide adaptation while the student is using the system; and secondly, the efficiency of instructional strategies proposed in Section 5.3.3. Regarding the latter, it is necessary to investigate various aspects over different combinations that can be made. As for feedback, since it is related only to the process of formative assessment, it must be determined which mechanism has a greater positive influence on learning. Regarding problems selection, it is necessary to study the many possible combinations to see which one has the greatest impact

according to two different goals: assessment and learning. When applying the final summative assessment, the important thing is to make a judgment and, consequently, the methods that make the assessment process more effective should be studied. When applying formative assessment, it is desirable that students learn and master the learning objectives. In this case, it should be studied which of the strategies perform better for this goal. It is anticipated that given the number of possible strategies and the different aspects involved in this task, may require a considerable effort. However, it is one of the most crucial for the future development of the methodology. As a part of this task, it would be necessary to consider other mechanisms with exposure control such as those proposed in (Barrada, 2012).

- After accomplishing the previous line, it should be verified if our proposal improves the student's instruction in comparison to traditional CBM systems. Although the base of the model is well-founded, compared with CBM heuristics, it is necessary to confirm that the approach actually improves instructive efficiency of the system. This would require comparing the learning effect between traditional CBM systems and the same system but implementing our methodology. Further in the future, it would be also desirable to compare the effectiveness of CBM + IRT methodology with the one achieved using cognitive tutors.

- During the development of this thesis, one of the problems that we had to face was the limited availability of real data on which to perform the studies. When we were provided with big datasets, such as SQL-Tutor, they were not applying our methodology and lot of data had to be discarded. For these reasons, it would be desirable to investigate the effect of the methodology in a large population where conclusions reached are significant, and with a period of usage large enough to determine whether the proposed approach is effective in a more realistic environment.

- To study the quality of constraints as an assessment instrument, we used the information function applied to constraints. Specifically, we have used the area under the curve determined by the aforementioned function as an indicative of quality. There are others properties of the curve, like the kurtosis or the maximum value, which can be used to detect abnormalities in the domain model. It would be interesting to study these properties in the information function and compare them with the one used in this thesis. In addition, there are other tools in testing environments that could provide different utilities to improve the domain model, and hence should be investigated as well.

- The framework CBMEngine still requires extending its functionality to provide a complete and self-contained component that can be used by external CBM systems. Primarily, we must expand the CBM-DoME tool for editing the domain model and integrate it as part of that framework. Regarding the use of the Siette system for assessment, it is also needed to complete the integration between the two systems and develop a mechanism to automatically generate the underlying model in the testing environment, avoiding the tedious task of manually creating composed items. Moreover, it would also be advisable to study the efficiency of using the framework in external systems as the communication protocol relies on

Web services and it is possible that they impose some restrictions or necessary improvements associated with efficiency.

- Our implemented tutoring systems should be extended to incorporate our formative assessment mechanism. This requires these systems to communicate with CBMEngine, which implies restructuring them to address different types of feedback and operation modes (calibration, formation and assessment); extend the operating logic; open the student model, for example using the system developed within the research group the doctoral candidate belongs to: Ingrid (Cruces et al., 2010; Conejo et al., 2011); and define their domain model in CBMEngine. Besides, OOPS uses an obsolete language, which should be changed for a modern one like Java if the system is to be used again in the future.

- Another feature that would be desirable and has not been considered in this work is the ability to add new constraints once the student has started using the system. This would require researching mechanisms to calibrate the new constraints without having to re-calibrate the complete set and its effect on efficiency. Also, it would be advisable to study the use of linking and equating methods in systems with a large number of constraints, where it is necessary to divide the total set in several sessions to perform the calibration.

- Apart from applying the proposed model to improve estimates provided by heuristics of CBM, there are numerous studies that extend the usefulness of this paradigm to different areas of ITS, such as the tutorials dialogues, collaboration, the affective aspect, etc. It would be desirable to study whether the proposal can improve system efficiency in these fields.

- Finally, as mentioned in Section 7.8, the study of the various psychometric properties has been quite narrow, focusing on a particular type of validity. However, there are many other types of validity that may be studied (Moss et al., 2006). In this field there are other ways to check the reliability of the assessment instrument (Cook & Beckman, 2006), as well as, various indices to study the accuracy and consistency provided by the IRT (Wyse & Hao, 2012). It would be very interesting to explore these measures and methods for studying other properties of the assessment methodology, and thus enhance the efficiency of the methodology itself.

# Bibliografía

*Un dicho ingenioso no prueba nada*
Voltaire (1694-1778)


Abad, F. J., Garrido, J., Olea, J. y Ponsoda, V. *Introducción a la psicometría*. Universidad Autónoma de Madrid, 2006.

AFT (American Federation of Teachers), NCME (National Council on Measurement in Education) y NEA (National Education Association). Standards for teacher competence in educational assessment of students. *Educational Measurement: Issues and Practices*, vol. 9, páginas 30–32, 1990.

Aleven, V., Koedinger, K. y Popescu, O. A tutorial dialog system to support self-explanation: Evaluation and open questions. En *Proceedings of the 11th International Conference on Artificial Intelligence in Education*, páginas 39–46. 2003.

Aleven, V., McLaren, B., Sewall, J. y Koedinger, K. Cognitive tutor authoring tools (ctat): Preliminary evaluation of efficiency gains. En *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, páginas 61–70. 2006.

Almond, R. G., Steinberg, L. S. y Mislevy, R. J. Enhancing the design and delivery of assessment systems: A four-process architecture. *Journal of Technology, Learning, and Assessment*, vol. 1(5), 2002.

Amalathas, S., Mitrovic, A., Saravanan, R. y Evison, D. Developing an intelligent tutoring system for palm oil in aspire. En *Proceedings of 18th International Conference on Computers in Education*, páginas 101–103. 2010.

Andersen, E. B. The solution of a set of conditional estimation equations. *Journal of the Royal Statistical Society*, vol. 33, páginas 42–54, 1972.

Anderson, J. y Pelletier, R. A development system for model-tracing tutors. En *Proceedings of the International Conference of the Learning Sciences*, páginas 1–8. 1991.

Anderson, J. R. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA, USA, 1983.

Anderson, J. R. *Rules of the Mind*. Lawrence Erlbaum Associates, 1993.

Anderson, J. R., Boyle, C., Corbett, A. y Lewis, M. Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, vol. 42, páginas 7–49, 1990.

ANDERSON, J. R., BOYLE, C. F. y YOST, G. The geometry tutor. En *Proceedings of the 9th international joint conference on Artificial intelligence*, páginas 1–7. 1985.

ANDERSON, J. R., CORBETT, A. T., KOEDINGER, K. R. y PELLETIER, R. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, vol. 4(2), páginas 167–207, 1995.

ANDERSON, J. R., FARRELL, R. y SAUERS, R. Learning to program in lisp. *Cognitive Science*, vol. 8, páginas 87–130, 1984.

ANDERSON, J. R., GREENO, J. G., KLINE, P. J. y NEVES, D. M. *Acquisition of Problem-solving Skill*, páginas 191–230. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.

ANDERSON, J. R. y LEBIERE, C. *The atomic components of thought*. Lawrence Erlbaum Associates, 1998.

ARNOTT, E., HASTINGS, P. y ALLBRITTON, D. Research methods tutor: Evaluation of a dialogue-based tutoring system in the classroom. *Behavior Research Methods*, vol. 40(3), páginas 694–672, 2008.

BADDELEY, A. D. *Human Memory: Theory and Practice*. Psychology Press, 1997.

BAGHAEI, N. A collaborative constraint-based adaptive system for learning object-oriented analysis and design using uml. En *Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, páginas 398–403. 2006.

BAGHAEI, N. y MITROVIC, A. A constraint-based collaborative environment for learning uml class diagrams. En *Proceedings of 8th International Conference on Intelligent Tutoring Systems*, vol. 4053 de *LNCS*, páginas 176–186. 2006.

BAGHAEI, N. y MITROVIC, A. From modelling domain knowledge to metacognitive skills: Extending a constraint-based tutoring system to support collaboration. En *Proceedings of 11th International Conference on User Modeling*, páginas 217–227. 2007.

BAGHAEI, N., MITROVIC, A. y IRWIN, W. A constraint-based tutor for learning object-oriented analysis and design using uml. En *Proceedings of 13th International Conference on Computers in Education*, páginas 11–18. 2005.

BAGHAEI, N., MITROVIC, A. y IRWIN, W. Problem-solving support in a constraint-based tutor for uml class diagrams. *Technology, Instruction, Cognition and Learning Journal (TICL)*, vol. 4(2), páginas 113–137, 2006.

BAGHAEI, N., MITROVIC, A. y IRWIN, W. Supporting collaborative learning and problem-solving in a constraint-based cscl environment for uml class diagrams. *International Journal of Computer-Supported Collaborative Learning*, vol. 2, páginas 159–190, 2007.

BAKER, F. y KIM, S. *Item Response Theory: Parameter Estimation Techniques, Second Edition*. Taylor & Francis, 2004.

BAKER, F. B. *The Basics of Item Response Theory. Second Edition*. Eric Publications, 2001.

BAKER, R. S., MITROVIC, A. y MATHEWS, M. Detecting gaming the system in constraint-based tutors. En *Proceedings of the International Conference on User Modelling, Adaptation and Presentation 2010*, vol. 6075, páginas 267–278. 2010.

BALI, M. *Drools JBoss Rules 5.0 Developer's Guide*. Packt Publishing, 2009.

BARRADA, J. R. Tests adaptativos informatizados: Una perspectiva general. *Anales de Psicología*, vol. 28, páginas 289–302, 2012.

BARROW, D., MITROVIC, A., OHLSSON, S. y GRIMLEY, M. Assessing the impact of positive feedback in constraint-based tutors. En *Intelligent Tutoring Systems* (editado por B. P. Woolf, E. Aïmeur, R. Nkambou y S. P. Lajoie), vol. 5091 de *LNCS*, páginas 250–259. Springer, 2008.

BAYES, T. An essay toward solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, vol. 53, páginas 370–418, 1763.

BEJAR, I. I., LAWLESS, R. R., MORLEY, M. E., WAGNER, M. E., BENNETT, R. E. y REVUELTA, J. A feasibility study of on-the-fly item generation in adaptive testing. Informe Técnico (GRE Professional Rep. No. 98-12P, ETS RR-02-23), Educational Testing Service, 2002.

BILLINGSLEY, W. y ROBINSON, P. Towards an intelligent online book for discrete mathematics. En *Proceedings of International Conference on Active Media Technology*, páginas 291–296. 2005.

BILLINGSLEY, W., ROBINSON, P., ASHDOWN, M. y HANSON, C. Intelligent tutoring and supervised problem solving in the browser. En *Proceedings of IADIS International Conference on WWW Internet*, páginas 806–811. 2004.

BINET, A., SIMON, T. y H., T. C. *A method of measuring the development of the intelligence of young children*. Chicago Med. Book Co., Chicago, IL, 1913.

BIRNBAUM, A. Efficient design and use of tests of mental ability for various decision-making problems. Informe Técnico 58-16 Project No. 7755-23, USAF School of Aviation Medicine, Randolph Air Force Base, Texas, 1957a.

BIRNBAUM, A. Further considerations of efficiency in tests of a mental ability. Informe Técnico 17 Project No. 7755-23, USAF School of Aviation Medicine, Randolph Air Force Base, Texas, 1957b.

BIRNBAUM, A. On the estimation of mental ability. Informe Técnico 15 Project No. 7755-23, USAF School of Aviation Medicine, Randolph Air Force Base, Texas, 1957c.

BIRNBAUM, A. *Some latent trait models and their use in inferring an examinee's mental ability*. Addison-Wesley, Reading, MA, 1968.

BLACK, P. y WILIAM, D. Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, vol. 80(2), páginas 139–148, 1998.

BLESSING, S., GILBERT, S., OURADO, S. y RITTER, S. Lowering the bar for creating model-tracing intelligent tutoring systems. En *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, páginas 443–450. 2007.

BLESSING, S. B. *A Programming by Demonstration Authoring Tool for Model-Tracing Tutors*, páginas 93–119. 2003.

BLOOM, B. The 2-sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, vol. 13, páginas 4–16, 1984.

BLOOM, B. S., HASTINGS, J. T. y MADAUS, G. F. *Handbook On Formative and Summative Evaluation of Student Learning*. McGraw-Hill, 1971.

BOAKE, C. From the binet-simon to the wechsler-bellevue: Tracing the history of intelligence testing. *Journal of Clinical and Experimental Neuropsychology*, vol. 24(3), página 383, 2002.

BOCK, R. D. Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, vol. 37, páginas 29–51, 1972.

BOCK, R. D. A brief history of item response theory. *Educational Measurement: Issues and Practice*, vol. 16, páginas 21–32, 1997.

BOCK, R. D. y AITKIN, M. Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, vol. 45(4), páginas 443–459, 1981.

BOCK, R. D. y LIEBERMAN, M. Fitting a response model for n dichotomously scored items. *Psychometrika*, vol. 35, páginas 179–197, 1970.

BONTCHEVA, K. y DIMITROVA, V. Examining the use of conceptual graphs in adaptive web-based systems that aid terminology learning. *International Journal of Artificial Intelligence tools*, vol. 13, páginas 299–332, 2004.

BRUSILOVSKY, P. A framework for intelligent knowledge sequencing and task sequencing. En *Intelligent Tutoring Systems, Second International Conference, ITS '92* (editado por C. Frasson, G. Gauthier y G. I. McCalla), vol. 608 de *LNCS*, páginas 499–506. Springer, 1992.

BRUSILOVSKY, P. Adaptive and intelligent technologies for web-based education. *Künstliche Intelligenz*, vol. 13(4), páginas 19–25, 1999.

BULL, S. Supporting learning with open learner models. páginas 47–61. 2004.

BULL, S. Preferred features of open learner models for university students. En *11th International Conference on Intelligent Tutoring Systems* (editado por S. A. Cerri, W. J. Clancey, G. Papadourakis y K. Panourgia), vol. 7315 de *LNCS*, páginas 411–421. Springer-Verlag, 2012.

CADE, W., COPELAND, J., PERSON, N. y D'MELLO, S. Dialogue modes in expert tutoring. En *Proceedings of Intelligent Tutoring Systems* (editado por E. Aïmeur, R. Nkambou y S. Lajoie), páginas 470–479. New York, NY, 2008.

CAMILLI, G. Origin of the scaling constant $d\bar{1}.7$ in item response theory. *Journal of Educational and Behavioral Statistics*, vol. 19(3), páginas 293–295, 1994.

CATTELL, J. M. y GALTON, F. Mental tests and measurements. *Mind*, vol. 15, páginas 373–381, 1980.

CAULEY, K. M. Studying knowledge acquisition: Distinctions among procedural, conceptual and logical knowledge. En *Annual Meeting of the American Educational Research Association*. 1986.

CEN, H., KOEDINGER, K. R. y JUNKER, B. Learning factors analysis - a general method for cognitive model evaluation and improvement. En *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, páginas 164–175. 2006.

CERRI, S. A., CLANCEY, W. J., PAPADOURAKIS, G. y PANOURGIA, K. *Proceedings of 11th International Conference on Intelligent Tutoring Systems*, vol. 7315 de *LNCS*. Springer-Verlag, 2012.

CHEN, P. P.-S. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, vol. 1(1), páginas 9–36, 1976.

CHILD, D. *The essentials of factor analysis*. Cassell, 1990.

CHIN, D. N. Empirical evaluation of user models and User-Adapted systems. *User Modeling and User-Adapted Interaction*, vol. 11(1-2), páginas 181–194, 2000.

CHOI, S. W. y SWARTZ, R. J. Comparison of cat item selection criteria for polytomous items. *Applied Psychological Measurement*, vol. 33(6), páginas 419–440, 2009.

CODD, E. F. Recent investigations into relational data base systems. Informe Técnico RJ1385, IBM, 1974.

COHEN, N. J. y SQUIRE, L. R. Preserved learning and retention of pattern-analyzing skill in amnesia: Dissociation of knowing how and knowing that. *Science*, vol. 21, páginas 207–210, 1980.

CONEJO, R., GUZMÁN, E., MILLÁN, E., TRELLA, M., PÉREZ-DE-LA-CRUZ, J. L. y RÍOS, A. Siette: A web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education*, vol. 14(1), páginas 29–61, 2004.

CONEJO, R., MILLÁN, E., PÉREZ-DE-LA CRUZ, J. L. y TRELLA, M. An empirical approach to on-line learning in siette. En *Proceedings of 3rd International Conference on Intelligent Tutoring Systems* (editado por C. F. y. K. V. G. Gauthier), páginas 604–614. 2000.

CONEJO, R., TRELLA, M., CRUCES, I. y GARCIA, R. Ingrid: A web service tool for hierarchical open learner model visualization. En *User Modeling, Adaptation and Personalization Workshops*, páginas 406–409. 2011.

COOK, D. A. y BECKMAN, T. J. Current concepts in validity and reliability for psychometric instruments: Theory and application. *The American Journal of Medicine*, vol. 119(2), páginas 7–16, 2006.

CORBETT, A. y ANDERSON, J. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, vol. 4, páginas 253–278, 1995.

CRONBACH, L. Coefficient alpha and the internal structure of tests. *Psychometrika*, vol. 16, páginas 297–334, 1951.

CROWLEY, R., LEGOWSKI, E., MEDVEDEVA, O., TSEYTLIN, E., ROH, E. y JUKIC, D. An its for medical classification problem-solving: Effects of tutoring and representations. En *Proceedings of the 2005 conference on Artificial Intelligence in Education*, páginas 192–199. 2005.

CRUCES, I., TRELLA, M., CONEJO, R. y GÁLVEZ, J. Student modeling services for hybrid web applications. En *Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems (UMAP 2010)* (editado por M. Yudelson, M. Pechenizkiy, E. Herder, G.-J. Houben y F. Abel), páginas 1–12. 2010.

DANTZIG, G. B. On the non-existence of tests of student's hypothesis having power functions independent of $\sigma$. *Annals of Mathematical Statistics*, vol. 11(2), páginas 186–192, 1940.

DEDALO. Dedalo: Desarrollo y evaluación de herramientas para el diagnóstico y el aprendizaje de los conocimientos matemáticos. 2009. Disponible en http://dedalo.lcc.uma.es (último acceso, octubre de 2012).

DEMARS, C. *Item Response Theory*. Series in understanding statistics: Measurement. Oxford University Press, USA, 2010.

DEMICHIEL, L. y KEITH, M. Java Persistence API. En *JavaOne Conference*. 2006.

DESMARAIS, M. y BAKER, R. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, vol. 22, páginas 9–38, 2012.

DI EUGENIO, B., FOSSATI, D., OHLSSON, S. y COSEJO, D. Towards explaining effective tutorial dialogues. En *Proceedings of 31th Annual Conference of the Cognitive Science Society* (editado por N. Taatgen y H. van Rijn), páginas 1430–1435. Cognitive Science Society, Austin, TX, 2009.

DIMITROVA, V. Style-olm: Interactive open learner modelling. *International Journal of Artificial Intelligence in Education*, vol. 13(1), páginas 35–78, 2003.

DOORENBOS, R. B. *Production Matching for Large Learning Systems*. Tesis Doctoral, Carnigie-Mellon University, Pittsburgh, PA Dept. of Computer Science, 1995.

DOUGLAS, J. Joint consistency of nonparametric item characteristic curve and ability estimation. *Psychometrika*, vol. 62, páginas 7–28, 1997.

DOUGLAS, J. y COHEN, A. Nonparametric item response function estimation for assessing parametric model fit. *Applied Psychological Measurement*, vol. 25(3), páginas 234–243, 2001.

DUAN, D., MITROVIC, A. y CHURCHER, N. Evaluating the effectiveness of multiple open student models in eer-tutor. En *Proceedings of 18th International Conference on Computers in Education*, páginas 86–88. 2010.

EHLERS, R. *Maximum Likelihood estimation procedures for categorical data*. Tesis Doctoral, Faculty of Natural and Agricultural Sciences, University of Pretoria, 2002.

EMBRETSON, S. y REISE, S. *Item Response Theory for Psychologists*. Multivariate Applications Book Series. Taylor & Francis, 2000.

EUBANK, R. *Nonparametric Regression and Spline Smoothing, Second Edition*. Statistics, Textbooks and Monographs. Taylor & Francis, 1999.

EVENS, M. y MICHAEL, J. *One-on-One Tutoring by Humans and Computers*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2006.

FERNÁNDEZ, J. y GÁLVEZ, J. *CBM-DoME: Un editor Web de modelos de dominio en Sistemas de enseñanza inteligentes bajo el paradigma del Modelado Basado en Restricciones*. Proyecto Fin de Carrera, Universidad de Málaga, 2011.

FORGY, C. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, vol. 19(1), páginas 17–37, 1982.

FOSSATI, D. The role of positive feedback in intelligent tutoring systems. En *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*, páginas 31–36. Association for Computational Linguistics, Stroudsburg, PA, USA, 2008.

FRANZ INC. Allegro common lisp. 1998. Disponible en http://www.franz.com/products/allegrocl/ (último acceso, octubre de 2012).

FRIEDMAN-HILL, E. J. Jess, the java expert system shell. Sandia National Laboratories, Livermore, CA, 1997.

GALLOPOULOS, E., HOUSTIS, E. y RICE, J. Computer as thinker/doer: problem-solving environments for computational science. *Computational Science Engineering, IEEE*, vol. 1(2), páginas 11 –23, 1994.

GÁLVEZ, J. A probabilistic model for student knowledge diagnosis in learning environments. En *Proceedings of the 2009 conference on Artificial Intelligence in Education* (editado por V. Dimitrova, R. Mizoguchi, B. du Boulay y A. Graesser), páginas 759–760. IOS Press, 2009.

GÁLVEZ, J., GÓMEZ, F. I., GUZMÁN, E. y CONEJO, R. Un sistema inteligente para el aprendizaje de fundamentos de programación orientada a objetos. En *Actas de la XII Conferencia de la Asociación Española para la Inteligencia Artificial* (editado por D. Borrajo, L. Castillo y J. M. Corchado), vol. 2, páginas 329–338. 2007.

GÁLVEZ, J., GUZMÁN, E. y CONEJO, R. A soa-based framework for constructing problem solving environments. En *The 8th IEEE International Conference on Advanced Learning Technologies* (editado por P. Díaz, Kinshuk, I. Aedo y E. Mora), páginas 126–127. IEEE Computer Society Conference Publishing Services, 2008.

GÁLVEZ, J., GUZMÁN, E. y CONEJO, R. A blended e-learning experience in a course of object oriented programming fundamentals. *Knowledge-Based Systems*, vol. 22(4), páginas 279–286, 2009a.

GÁLVEZ, J., GUZMÁN, E. y CONEJO, R. Data-driven student knowledge assessment through ill-defined procedural tasks. En *Current Topics in Artificial Intelligence, 13th Conference of the Spanish Association for Artificial Intelligence* (editado por P. M. González, L. M. Andaluz y R. M. Gasca), vol. 5988 de *LNAI*, páginas 233–241. Springer, 2009b.

GÁLVEZ, J., GUZMÁN, E. y CONEJO, R. Exploring quality of constraints for assessment in problem solving environments. En *11th International Conference on Intelligent Tutoring Systems* (editado por S. A. Cerri, W. J. Clancey, G. Papadourakis y K. Panourgia), vol. 7315 de *LNCS*, páginas 310–319. Springer-Verlag, 2012.

GÁLVEZ, J., GUZMÁN, E., CONEJO, R. y MILLÁN, E. Student knowledge diagnosis using item response theory and constraint-based modeling. En *Proceedings of the 2009 conference on Artificial Intelligence in Education* (editado por V. Dimitrova, R. Mizoguchi, B. du Boulay y A. Graesser), páginas 291–298. IOS Press, 2009c.

GEORGIEV, N. Item analysis of c, d and e series from raven's standard progressive matrices with item response theory two-parameter logistic model. *Europe's Journal of Psychology*, vol. 4(3), 2008.

GLASER, R. Instructional technology and the measurement of learning outcomes: Some questions. *American Psychologist*, vol. 18, páginas 519–521, 1963.

GOKHALE, A. A. Collaborative learning enhances critical thinking. *J. Technology Education*, vol. 7, páginas 22–30, 1995.

GÓMEZ, F. I. y GUZMÁN, E. Tutor web de fundamentos de programación orientada a objetos. En *Proyecto Fin de Carrera en la titulación de Ingeniería Informática*. 2006.

GRAESSER, A., CHIPMAN, P., HAYNES, B. y OLNEY, A. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions in Education*, vol. 48(4), páginas 612–618, 2005.

GUTTMAN, L. A basis for analyzing test-retest reliability. *Psychometrika*, vol. 10, páginas 255–282, 1945.

GUZMÁN, E. *Un modelo de evaluación cognitiva basado en tests adaptativos informatizados para el diagnóstico en sistemas tutores inteligentes*. Tesis Doctoral, Universidad de Málaga, 2005.

GUZMÁN, E. y CONEJO, R. A brief introduction to the new architecture of siette. En *Proceedings of the 3rd international conference on adaptive hypermedia and adaptive web-based systems* (editado por P. D. Bra y W. Nejdl), páginas 405–408. 2004a.

GUZMÁN, E. y CONEJO, R. A library of templates for exercise construction in an adaptive assessment system. *Technology, Instruction, Cognition and Learning*, vol. 2(1-2), páginas 21–43, 2004b.

GUZMÁN, E., CONEJO, R. y DE-LA CRUZ, J. L. P. Adaptive testing for hierarchical student models. *User Modeling and User-Adapted Interaction*, vol. 17(1-2), páginas 119–157, 2007.

GUZMÁN, E., CONEJO, R. y PÉREZ-DE-LA-CRUZ, J. L. Improving student performance using self-assessment tests. *IEEE Intelligent Systems*, vol. 22, páginas 46–52, 2007.

HALEY, D. C. Estimation of the dosage mortality relationship when the dose is subject to error. Informe Técnico SOL ONR 15, Applied Mathematics and Statistics Laboratory, Stanford University, Palo Alto - CA (USA), 1952.

HAMBLETON, R. y JONES, R. W. Comparison of classical test theory and item response theory and their applications to test development. *Educational measurement: issues and practice*, vol. 12(3), páginas 38–47, 1993.

HAMBLETON, R. K., SWAMINATHAN, H. y ROGERS, H. J. *Fundamentals of Item Response Theory*. Measurement Methods for the Social Science. Sage Publications, Inc, 1991.

HÄRDLE, W. *Nonparametric and Semiparametric Models*. Springer, 2004.

HARTLEY, D. y MITROVIC, A. Supporting learning by opening the student model. En *Proceedings of 6th International Conference on Intelligent Tutoring Systems*, vol. 2363 de *LNCS*, páginas 453–462. Springer-Verlag, 2002.

HEINZE, A. y PROCTER, C. Online communication and information technology education. *Journal of Information Technology Education*, vol. 5, páginas 235–249, 2006.

HEMKER, B. T., SIJTSMA, K., MOLENAAR, I. W. y JUNKER, B. W. Stochastic ordering using the latent trait and the sum score in polytomous irt models. *Psychometrika*, vol. 62(3), páginas 331–347, 1997.

HOLLAND, J., BAGHAEI, N., MATHEWS, M. y MITROVIC, A. The effects of domain and collaboration feedback on learning in a collaborative intelligent tutoring system. En *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, vol. 6738, páginas 469–471. Springer, 2011.

HOLLAND, J., MITROVIC, A. y MARTIN, B. J-latte: a constraint-based tutor for java. En *Proceedings of 17th International on Conference Computers in Education*, páginas 142–146. 2009.

HOLLAND, P. y ROSENBAUM, P. Conditional association and unidimensionality in monotone latent variable models. *The Annals of Statistics*, vol. 14(4), páginas 1523–1543, 1986.

HOLT, P., DUBS, S., JONES, M. y GREER, J. The state of student modelling. En *Student Modeling: the Key to Individualized Knowledge- based Instruction* (editado por J. Greer y G. McCalla), vol. 125, páginas 3–35. Springer-Verlag, New York, 1994.

HONTANGAS, P., PONSODA, V., OLEA, J. y ABAD, F. Los test adaptativos informatizados en la frontera del siglo xxi: una revisión. *Metodología de las Ciencias del Comportamiento*, vol. 2(2), páginas 183–216, 2000.

HOPKINS, K. y STANLEY, J. Test validity. *Educational and psychological measurement and evaluation*, páginas 76–106, 1981.

HOPPE, U., OGATA, H. y SOLLER, A. *Role of Technology in Computer-Supported Collaborative Learning: Studies in Technology Enhanced Collaborative Learning*. Springer, 2007.

HUBA, M. E. y FREED, J. E. *Learner-Centered Assessment on College Campuses: shifting the focus from teaching to learning*. Allyn and Bacon, 2000.

HUEBNER, A. An overview of recent developments in cognitive diagnostic computer adaptive assessments. *Practical Research Assessment and Evaluation*, vol. 15(3), 2010.

INABA, A. y MIZOGUCHI, R. Learners'roles and predictable educational benefits in collaborative learning; an ontological approach to support design and analysis of cscl. En *Proceedings of 7th International Conference on Intelligent Tutoring Systems* (editado por J. Lester, R. Vicari y F. Paraguacu), páginas 285–294. 2004.

ISOMORPHIC SOFTWARE. Smartclient. 2009. Disponible en http://www.smartclient.com/ (último acceso, octubre de 2012).

JENNRICH, R. I. y SAMPSON, P. F. Newton-raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, vol. 18(1), páginas 11–17, 1976.

JIMÉNEZ-DÍAZ, G., GÓMEZ-ALBARRÁN, M., GÓMEZ-MARTÍN, M. y GONZÁLEZ-CALERO, P. Virplay: Playing roles to understand dynamic behavior. En *Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts*. 2005.

DE JONG, T. y FERGUSON-HESSLER, M. G. M. Types and qualities of knowledge. *Educational Psychologist*, vol. 31(2), páginas 105–113, 1996.

JUNKER, B. y ELLIS, J. A characterization of monotone unidimensional latent variable models. *The Annals of Statistics*, vol. 25(3), páginas 1327–1343, 1997.

JUNKER, B. y SIJTSMA, K. Nonparametric item response theory in action: An overview of the special issue. *Applied Psychological Measurement*, vol. 25, páginas 211–220, 2001.

JUNKER, B. W. The role of nonparametric analysis in assessment modeling: Then and now. En *Looking Back* (editado por N. J. Dorans y S. Sinharay), vol. 202 de *Lecture Notes in Statistics*, páginas 67–85. Springer New York, 2011.

KAY, J. Learner know thyself student models to give learner control and responsibility. En *Proceedings of 5th International Conference on Computers in Education*, páginas 18–26. 1997.

KHAN, M. y JAIN, P. *Theory and Problems in Financial Management*. McGraw Hill Education, 1999.

KLEINBAUM, D. G. y KLEIN, M. *Logistic regression: A self-learning text, Third edition*. Springer, 2010.

KODAGANALLUR, V., WEITZ, R. y ROSENTHAL, D. A comparison of model-tracing and constraint-based intelligent tutoring paradigms. *International Journal of Artificial Intelligence in Education*, vol. 15(2), páginas 117–144, 2005.

KODAGANALLUR, V., WEITZ, R. R. y ROSENTHAL, D. An assessment of constraint-based tutors: A response to mitrovic and ohlsson's critique of "a comparison of model-tracing and constraint-based intelligent tutoring paradigms". *International Journal of Artificial Intelligence in Education*, vol. 16, páginas 291–321, 2006.

KOEDINGER, K., ALEVEN, V., HEFFERNAN, N., MCLAREN, B. y HOCKENBERRY, M. Opening the door to nonprogrammers: Authoring intelligent tutor behavior by demonstration. En *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, páginas 162–174. 2004.

KOEDINGER, K. R. y ALEVEN, V. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, vol. 19, páginas 239–264, 2007.

KOEDINGER, K. R., ANDERSON, J. R., HADLEY, W. H. y MARK, M. A. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, vol. 8, páginas 30–43, 1997.

KUDER, G. F. y RICHARDSON, M. W. The theory of the estimation of test reliability. *Psychometrika*, vol. 2, páginas 151–160, 1937.

LANGLEY, P. W., OHLSSON, S. y SAGE, S. A machine learning approach to student modeling. technical report. Informe Técnico CMU-RI-TR-84-7, Carnegie Mellon University, Pittsburgh, PA., 1984.

LAZARSFELD, P. F. *The Logical and Mathematical Foundations of Latent Structure Analysis*, páginas 362–412. Princeton University Press, 1950a.

LAZARSFELD, P. F. *Some Latent Structures*, páginas 413–472. Princeton University Press, 1950b.

LE, N.-T. A constraint-based assessment approach for free-form design of class diagrams using uml. En *Proceedings of Workshop on Intelligent Tutoring Systems for Ill-Defined Domains, 8th International Conference on Intelligent Tutoring Systems* (editado por P. N. L.-C. Ashley, K.), páginas 11–19. 2006.

LE, N.-T., MENZEL, W. y PINKWART, N. Evaluation of a constraint-based homework assistance system for logic programming. En *Proceedings of 17th International Conference on Computers in Education* (editado por J. Lee, C. Liu y C. Looi), páginas 51–58. APSCE, Putrajaya, 2009.

LEE, Y.-J., PALAZZO, D. J., WARNAKULASOORIYA, R. y PRITCHARD, D. E. Measuring student learning with item response theory. *Physical Review Special Topics - Physics Education Research*, vol. 4, 010102, páginas 1–6, 2008.

LEE, Y.-S. A comparison of methods for nonparametric estimation of item characteristic curves for binary items. *Applied Psychological Measurement*, vol. 31(2), páginas 121–134, 2007.

LI, N., COHEN, W. W. y KOEDINGER, K. R. Efficient cross-domain learning of complex skills. En *11th International Conference on Intelligent Tutoring Systems*, páginas 493–498. 2012.

VAN DER LINDEN, W. J. Empirical initialization of the trait estimator in adaptive testing. *Applied Psychological Measurement*, vol. 23, páginas 21–29, 1999.

VAN DER LINDEN, W. J. Equating scores from adaptive to linear tests. *Applied Psychological Measurement*, vol. 30(6), páginas 493–508, 2006.

VAN DER LINDEN, W. J. Constrained adaptive testing with shadow tests. En *Elements of Adaptive Testing* (editado por W. J. van der Linden y C. A. Glas), páginas 31–55. Springer New York, 2010.

VAN DER LINDEN, W. J. y GLAS, C. A. W. Computerized adaptive testing: Theory and practice. 2000.

VAN DER LINDEN, W. J. y HAMBLETON, R. K. *Handbook of Modern Item Response Theory*. Springer, 1996.

VAN DER LINDEN, W. J. y PASHLEY, P. J. *Item selection and ability estimation adaptive testing*, páginas 3–10. Springer, 2010.

LITTMAN, D. y SOLOWAY, E. *Evaluating ITS: The cognitive science perspective*, páginas 209–242. Lawrence Erlbaum Associates, Inc., 1988.

LÓPEZ, J. M., MILLÁN, E., DE-LA CRUZ, J. L. P. y TRIGUERO, F. Ilesa: a web-based intelligent learning environment for the simplex algorithm. En *Proc. of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering* (editado por C. Alvegård), páginas 399–406. Göteborg, Sweden, 1998.

LORD, F. M. *A theory of test scores*. Psychometric monograph. Psychometric Society, 1952.

LORD, F. M. Estimating test reliability. *Educational and Psychological Measurement*, vol. 15, páginas 325–336, 1955.

LORD, F. M. y NOVICK, M. R. *Statistical theories of mental test scores*. Addison-Wesley, Reading, MA, 1968.

LUECHT, R. M. y SIRECI, S. G. A review of models for computer-based testing. Informe Técnico 2011-12, College Board, 2011.

MABBOTT, A. y BULL, S. Alternative views on knowledge: Presentation of open learner models. En *7th International Conference on Intelligent Tutoring Systems* (editado por R. V. . F. P. J.C. Lester), páginas 689–698. Springer-Verlag, 2004.

MALLERY, J. C. A common lisp hypermedia server. En *Proceedings of 1st Int. Conference on the World Wide Web*, páginas 745–749. 1994.

MARTIN, B., KIRKBRIDE, T., MITROVIC, A., HOLLAND, J. y ZAKHAROV, K. An intelligent tutoring system for medical imaging. En *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, páginas 502–509. 2009.

MARTIN, B. y MITROVIC, A. Tailoring feedback by correcting student answers. En *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, páginas 383–392. Springer-Verlag, London, UK, UK, 2000.

MARTIN, B. y MITROVIC, A. Authoring web-based tutoring systems with wetas. En *Proceedings of 10th International Conference on Computers in Education*, páginas 183–187. 2002a.

MARTIN, B. y MITROVIC, A. Automatic problem generation in constraint-based tutors. En *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, páginas 388–398. Springer-Verlag, London, UK, 2002b.

MARTIN, B. y MITROVIC, A. Wetas: A web-based authoring system for constraint-based its. En *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, páginas 543–546. Springer-Verlag, London, UK, 2002c.

MARTIN, B. y MITROVIC, A. Using learning curves to mine student models. En *10th International Conference on User Modeling*, páginas 79–88. 2005.

MARTIN, B. y MITROVIC, A. The effect of adapting feedback generality in its. En *Adaptive Hypermedia and Adaptive Web-Based Systems*, 4018, páginas 192–202. 2006.

MARTIN, B. y MITROVIC, A. Helping teachers build its with domain schema. En *Proceedings of the 9th international conference on Intelligent Tutoring Systems*, páginas 194–203. Springer-Verlag, Berlin, Heidelberg, 2008.

MARTIN, B., MITROVIC, A., KOEDINGER, K. R. y MATHAN, S. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, vol. 21(3), páginas 249–283, 2011.

MARTIN, B., MITROVIC, A. y SURAWEERA, P. Domain modelling with ontology: A case study. En *Proceedings of the 5th International Workshop on Authoring of Adaptive and Adaptable Hypermedia* (editado por A. Cristea y R. Carro), páginas 4–11. 2007.

MARTIN, J. y VANLEHN, K. Student assessment using bayesian nets. *International Journal of Human-Computer Studies*, vol. 42, páginas 575–591, 1995.

MARZANO, R. J. Standardized tests: Do they measure general cognitive abilities? answer: No. *NASSP Bulletin*, vol. 74(526), páginas 93–101, 1990.

MASTERS, G. N. A rasch model for partial credit scoring. *Psychometrika*, vol. 47(3), páginas 149–174, 1982.

MASTERS, G. N. *The partial credit model*, páginas 109–122. Springer, 2010.

MATHEWS, M. y MITROVIC, A. Investigating the effectiveness of problem templates on learning in itss. En *Proceedings of 13th International Conference on Artificial Intelligence in Education* (editado por R. Luckin, K. Koedinger y J. Greer), páginas 611–613. 2007.

MATHEWS, M., MITROVIC, A., LIN, B., HOLLAND, J., y CHURCHER, N. Do your eyes give it away? using eye tracking data to understand studentsáttitudes towards open student model representations. En *11th International Conference on Intelligent Tutoring Systems* (editado por S. A. Cerri, W. J. Clancey, G. Papadourakis y K. Panourgia), vol. 7315 de *LNCS*, páginas 422–427. Springer-Verlag, 2012.

MATHEWS, M., MITROVIC, A. y THOMSON, D. Analysing high-level help-seeking behaviour in itss. En *5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, páginas 312–315. 2008.

MAYO, M. y MITROVIC, A. Using a probabilistic student model to control problem difficulty. En *Intelligent Tutoring Systems* (editado por G. Gauthier, C. Frasson y K. VanLehn), vol. 1839 de *LNCS*, páginas 524–533. Springer Berlin / Heidelberg, 2000.

MAYO, M. y MITROVIC, A. Optimising its behaviour with bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, vol. 12, páginas 124–153, 2001.

MAYO, M., MITROVIC, A. y MCKENZIE, J. Capit: an intelligent tutoring system for capitalisation and punctuation. *Proceedings. International Workshop on Advanced Learning Technologies, 2000*, páginas 151–154, 2000.

MCLAREN, B., LIM, S. y KOEDINGER, K. When and how often should worked examples be given to students? new results and a summary of the current state of research. En *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, páginas 2176–2181. 2008.

MENZEL, W. Constraint-based modeling and ambiguity. *International Journal of Artificial Intelligence in Education*, vol. 16(1), páginas 29–63, 2006.

MILIK, N., MARSHALL, M. y MITROVIC, A. Responding to free-form student questions in erm-tutor. En *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, páginas 707–709. Springer-Verlag, 2006a.

MILIK, N., MARSHALL, M. y MITROVIC, A. Teaching logical database design in erm-tutor. En *Proceedings of 8th International Conference on Intelligent Tutoring Systems* (editado por M. Ikeda, K. Ashley y T.-W. Chan), vol. 4053 de *LNCS*, páginas 707–709. 2006b.

MILLÁN, E., GARCÍA-HERVÁS, E., GUZMÁN, E., ÁNGEL RUEDA y DE-LA CRUZ, J. L. P. Tapli: An adaptive web-based learning environment for linear programming. En *10ª Conferencia de la Asociaciín Española para la Inteligencia Artificial* (editado por R. Conejo, M. Urretavizcaya y J. L. P. de-la Cruz), vol. 3040 de *LNCS*, páginas 676–685. Springer, 2003.

MILLÁN, E., MANDOW, L. y REY, L. Eplar: Un entorno para la enseñanza de la programación lineal. 1999.

MILLS, C. y DALGARNO, B. A conceptual model for game-based intelligent tutoring systems. En *ICT: Providing choices for learners and learning, Proceedings ASCILITE conference*, páginas 692–701. Singapore, 2007.

MISLEVY, R. J. Evidence-centered design for simulation-based assessment. Informe Técnico CRESST REPORT 800, The National Center for Research on Evaluation, Standards, and Student Testing, University of California, Los Angeles, 2011.

MISLEVY, R. J., ALMOND, R. G. y LUKAS, J. F. A brief introduction to evidence-centered design. Informe Técnico CSE Report 632, The National Center for Research on Evaluation, Standards, and Student Testing, University of California, Los Angeles, 2003a.

MISLEVY, R. J. y RICONSCENTE, M. M. *Evidence-centered assessment design: Layers, concepts, and terminology*, páginas 61–90. L. Erlbaum Associates, 2006.

MISLEVY, R. J., STEINBERG, L. S. y ALMOND, R. G. On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*, vol. 1, páginas 3–67, 2003b.

MITROVIC, A. Sql-tutor: a preliminary report. En *Technical Report TR-COSC 08/97*. Department of Computer Science, University of Canterbury, 1997.

MITROVIC, A. Experiences in implementing Constraint-Based modeling in SQL-Tutor. En *Intelligent Tutoring Systems*, páginas 414–423. 1998a.

MITROVIC, A. A knowledge-based teaching system for sql. En *Proceedings of ED-MEDIA 98* (editado por T. Ottmann y I. Tomek), páginas 1027–1032. Vancouver, CA, 1998b.

MITROVIC, A. Learning sql with a computerized tutor. En *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, páginas 307–311. ACM, New York, NY, USA, 1998c.

MITROVIC, A. Normit: A web-enabled tutor for database normalization. En *ICCE '02: Proceedings of the International Conference on Computers in Education*, páginas 1276–1280. 2002.

MITROVIC, A. An intelligent sql tutor on the web. *Int. Journal of Artificial Intelligence in Eucation*, vol. 13(2-4), páginas 173–197, 2003a.

MITROVIC, A. Supporting self-explanation in a data normalization tutor. En *Proceedings of the 11th International Conference on Artificial Intelligence in Education*, páginas 565–577. 2003b.

MITROVIC, A. The effect of explaining on learning: a case study with a data normalization tutor. En *Proceedings of 12th International Conference on Artificial Intelligence in Education* (editado por C.-K. Looi, G. McCalla, B. Bredeweg y J. Breuker), vol. 13, páginas 499–506. IOS Press, 2005a.

MITROVIC, A. Scaffolding answer explanation in a data normalization tutor. *Facta universitatis*, vol. 18, páginas 151–163, 2005b.

MITROVIC, A. Large-scale deployment of three intelligent web-based database tutors. *Journal of Computing and Inforation Technology*, vol. 14(4), páginas 275–281, 2006.

MITROVIC, A. Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, vol. 22, páginas 39–72, 2012.

MITROVIC, A., KOEDINGER, K. R. y MARTIN, B. A comparative analysis of cognitive tutoring and constraint-based modeling. En *Proceedings of the Ninth International Conference on User Modeling* (editado por P. Brusilovsky, A. T. Corbett y F. de Rosis), LNAI, páginas 313–322. Springer, 2003.

MITROVIC, A. y MARTIN, B. Evaluating the effectiveness of feedback in SQL-Tutor. En *Advanced Learning Technologies, 2000. IWALT 2000. Proceedings. International Workshop on*, páginas 143–144. 2000.

MITROVIC, A. y MARTIN, B. Evaluating the effects of open student models on learning. En *Proceedings of 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems* (editado por P. de Bra, P. Brusilovsky y R. Conejo), vol. 2347 de *LNCS*, páginas 296–305. Springer-Verlag, 2002.

MITROVIC, A. y MARTIN, B. Scaffolding and fading problem selection in sql-tutor. En *Proc. 11th Int. Conference on Artificial Intelligence in Education* (editado por U. Hoppe, F. Verdejo y J. Kay), páginas 479–481. IOS Press, 2003.

MITROVIC, A. y MARTIN, B. Evaluating adaptive problem selection. En *AH*, páginas 185–194. 2004.

MITROVIC, A., MARTIN, B. y MAYO, M. Using evaluation to shape its design: Results and experiences with sql-tutor. *User Modeling and User-Adapted Interaction*, vol. 12(2-3), páginas 243–279, 2002.

MITROVIC, A., MARTIN, B. y SURAWEERA, P. Intelligent tutors for all: The constraint-based approach. *IEEE Intelligent Systems*, vol. 22(4), páginas 38–45, 2007.

MITROVIC, A., MARTIN, B., SURAWEERA, P., ZAKHAROV, K., MILIK, N. y HOLLAND, J. Aspire: Student modelling and domain specification. En *Technical Report TR-08/05*. Intelligent Computer Tutoring Group, Department of Computer Science and Software Engineering, University of Canterbury,, 2005.

MITROVIC, A., MARTIN, B., SURAWEERA, P., ZAKHAROV, K., MILIK, N., HOLLAND, J. y MCGUIGAN, N. Aspire: An authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education*, vol. 19(2), páginas 155–188, 2009.

MITROVIC, A., MAYO, M., SURAWEERA, P. y MARTIN, B. Constraint-based tutors: A success story. En *Proceedings of the 14th International conference on Industrial and engineering applications of artificial intelligence and expert systems*, páginas 931–940. Springer-Verlag, 2001.

MITROVIC, A., MCGUIGAN, N., MARTIN, B., SURAWEERA, P., MILIK, N. y HOLLAND, J. Authoring constraint-based systems in aspire: a case study of a capital investment tutor. En *Proceedings of ED-MEDIA 2008*, páginas 4607–4616. 2008.

MITROVIC, A. y OHLSSON, S. Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education*, vol. 10, páginas 238–256, 1999.

MITROVIC, A. y OHLSSON, S. A critique of kodaganallur, weitz and rosenthal, "a comparison of model-tracing and constraint-based intelligent tutoring paradigms". *International Journal of Artificial Intelligence in Education*, vol. 16, páginas 277–289, 2006.

MITROVIC, A. y OHLSSON, S. Fidelity and efficiency of knowledge representations for intelligent tutoring systems. *Technology, Instruction, Cognition and Learning*, vol. 5(2-3-4), páginas 101–132, 2007.

MITROVIC, A., SURAWEERA, P., MARTIN, B. y WEERASINGHE, A. DB-Suite: experiences with three intelligent, Web-Based database tutors. *Journal of Interactive Learning Research*, vol. 15(4), página 409, 2004.

MITROVIC, A., SURAWEERA, P., MARTIN, B., ZAKHAROV, K., MILIK, N. y HOLLAND, J. Authoring constraint-based tutors in aspire. En *Proceedings of 8th International Conference on Intelligent Tutoring Systems* (editado por M. Ikeda, K. D. Ashley y T.-W. Chan), vol. 4053, páginas 41–50. Springer, 2006.

MITROVIC, A. y WEERASINGHE, A. Revisiting ill-definedness and the consequences for itss. En *14th International Conference on Artificial Intelligence in Education*, páginas 375–382. 2009.

MITROVIC, A., WILLIAMSON, C., BEBBINGTON, A., MATHEWS, M., SURAWEERA, P., MARTIN, B., THOMSON, D. y HOLLAND, J. Thermo-tutor: An intelligent tutoring system for thermodynamics. En *Learning Environments and Ecosystems in Engineering Education*, páginas 378–385. 2011.

MOLENAAR, I. W. Thirty years of nonparametric item response theory. *Applied Psychological Measurement*, vol. 25(3), páginas 295–299, 2001.

MORSCHEL, I. Smalltutor - an intelligent tutoring system for object-oriented-programming. páginas 19–25. 1993.

MOSS, P. A., GIRARD, B. J. y HANIFORD, L. C. Validity in educational assessment. *Review of Research in Education*, vol. 30(1), páginas 109–162, 2006.

MUÑIZ, J. *Teoría clásica de los tests*. Pirámide, 2003.

MUÑIZ, J. Las teorías de los tests: teoría clásica y teoría de respuesta a los ítems. *Papeles del Psicólogo*, vol. 31(1), páginas 57–66, 2010.

MURAKI, E. A generalized partial credit model: Application to an em algorithm. *Applied Psychological Measurement*, vol. 16, páginas 159–176, 1982.

MURRAY, T. Authoring intelligent tutoring systems: an analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, vol. 10, páginas 98–129, 1999.

MURRAY, T. *Authoring Tools for Advanced Learning environments*, capítulo An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art, páginas 491–544. Kluwer Academic Publishers, 2003.

NAVARRETE, C., WILDE, J., NELSON, C., MARTINEZ, R. y HARGETT, G. *Informal Assessment in Educational Evaluation: Implications for Bilingual Education Programs*. National Clearinghouse for Bilingual Education: Program information guide series 3, summer, Washington, DC, 1990.

NERING, M. y OSTINI, R. *Handbook of polytomous item response theory models*. Taylor & Francis Group, 2010.

NKAMBOU, R., BOURDEAU, J. y MIZOGUCHI, R. *Advances in Intelligent Tutoring Systems*. Springer, 2010.

OH, Y., GROSS, M. D., ISHIZAKI, S. y DO, Y.-L. Constraint-based design critic for flat-pack furniture design. En *Proceedings of 17th International Conference on Computers in Education*, páginas 19–26. 2009.

OHLSSON, S. Some principles of intelligent tutoring. *Instructional Science*, vol. 14, páginas 293–326, 1986.

OHLSSON, S. Constraint-based student modeling. *International Journal of Artificial Intelligence in Education*, vol. 3(4), páginas 429–447, 1992.

OHLSSON, S. *The Interaction between Knowledge and Practice in the Acquisition of Cognitive Skills*, páginas 147–208. Norwell, MA: Kluwer, 1993.

OHLSSON, S. *Constraint-based Student Modeling*, capítulo Student Modeling: the Key to Individualized Knowledge-based Instruction, páginas 167–189. Springer-Verlag, 1994.

OHLSSON, S. Learning from performance errors. *Psychological Review*, vol. 103(2), páginas 241–262, 1996.

OHLSSON, S., DI EUGENIO, B., CHOW, B., FOSSATI, D., LU, X. y KERSHAW, T. Beyond the code-and-count analysis of tutoring dialogues. En *Proceedings of Artificial Intelligence in Education: Building Technology Rich Learning Contexts thatWork* (editado por R. Luckin, K. Koedinger y J. Greer), páginas 349–356. IOS Press, Amsterdam, 2007.

OHLSSON, S. y MITROVIC, A. Constraint-based knowledge representation for individualized instruction. *Computer Science and Information Systems*, vol. 3, páginas 1–22, 2006.

OHLSSON, S. y REES, E. The function of conceptual understanding in the learning of arithmetic procedures. *Cognition and Instruction*, vol. 8(2), páginas 103–179, 1991.

OLEA, J., ABAD, F. J. y BARRADA, J. R. Test informatizados y otros nuevos tipos de tests. *Papeles del Psicólogo*, vol. 31(1), páginas 94–107, 2010.

OLEA, V., J. Y PONSODA. Tests adaptativos informatizados. 2002. Disponible en http://aristidesvara.net/pgnWeb/metodologia/psicometria/teoria_respuesta/index.html (último acceso, octubre de 2012).

OWEN, R. J. A bayesian sequential procedure for quantal response in the context of adaptive mental testing. *Journal of the American Statistical Association*, vol. 70(350), páginas 351–371, 1975.

PANI, R. *Integral Education: thought & Practical*. APH Publishing, 2007.

PARDOS, Z. A. y HEFFERNAN, N. T. Kt-idem: Introducing item difficulty to the knowledge tracing model. En *Proceedings of the 19th International Conference on User Modeling, Adaptation and Personalization*, páginas 243–254. 2011.

PARSHALL, C. G., HARMES, J. C., DAVEY, T. y PASHLEY, P. J. Innovative items for computerized testing. En *Elements of Adaptive Testing* (editado por W. J. van der Linden y C. A. W. Glas), Statistics for Social and Behavioral Sciences, páginas 215–230. Springer, 2010.

PAVLIK, P. I., CEN, H. y KOEDINGER, K. R. Learning factors transfer analysis: Using learning curve analysis to automatically generate domain models. En *Proceedings of the 2nd International Conference on Educational Data Mining*, páginas 121–130. 2009a.

PAVLIK, P. I., CEN, H. y KOEDINGER, K. R. Performance factors analysis - a new alternative to knowledge tracing. En *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, páginas 531–538. 2009b.

PETRY, P. y ROSATELLI, M. Algolc: a learning companion system for teaching and learning algorithms. En *Proceeding of the 8th International Conference on Intelligent Tutoring Systems*, páginas 775–777. 2006.

PIAGET, J. Piaget's theory. *Carmichael's Manual of Child Psychology*, vol. 1, páginas 703–732, 1970.

PILLAY, N. A generic architecture for the development of intelligent programming tutors. *International Journal of Continuing Lifelong Learning*, vol. 10, páginas 275–285, 2000.

POLSON, M. C. y RICHARDSON, J. J. *Foundations of Intelligent Tutoring Sytems*. Psychology Press, 1988.

PONSODA, V. Overview of the computerized adaptive testing special section. *Psicología*, vol. 21, páginas 115–120, 2000.

RAMAPRASAD, A. On the definition of feedback. *Behavioural Science*, vol. 28, páginas 4–13, 1983.

RAMSAY, J. Kernel smoothing approaches to nonparametric item characteristic curve estimation. *Psychometrika*, vol. 56, páginas 611–630, 1991.

RAO, C. y SINHARAY, S. *Handbook of Statistics: Psychometrics*. Elsevier, 2007.

RASCH, G. *Probabilistic models for some intelligence and attainment tests*. Institute for Educational Research, Copenhagen, Denmark, 1960.

RAZZAQ, L., FENG, M., HEFFERNAN, N., KOEDINGER, K., NUZZO-JONES, G., JUNKER, B., MACASEK, M., RASMUSSEN, K., TURNER, T. y WALONOSKI, J. Blending assessment and instructional assistance. En *Intelligent Educational Machines within the Intelligent Systems Engineering Book Series*, páginas 23–49. Springer Berlin Heidelberg, 2007.

RAZZAQ, L., PATVARCZKI, J., ALMEIDA, S. F., VARTAK, M., FENG, M., HEFFERNAN, N. T. y KOEDINGER, K. R. The ASSISTment builder: Supporting the life cycle of tutoring system content creation. *IEEE Transactions on Learning Technologies*, vol. 2(2), páginas 157–166, 2009.

RECKASE, D. *Multidimensional Item Response Theory*. Statistics for Social and Behavioral Sciences. Springer, 2009.

RECKASE, M. D. Designing item pools to optimize the functioning of a computerized adaptive test. *Psychological Test and Assessment Modeling*, vol. 52(2), páginas 127–141, 2010.

REEVE, B. B. y FAYERS, P. *Applying Item Response Theory modelling for evaluating questionnaire item and scale properties*, páginas 55–73. Taylor & Francis, 2005.

RÍOS, A., CONEJO, R., TRELLA, M., MILLÁN, E. y PÉREZ-DE-LA CRUZ, J. L. Aprendizaje automático de las curvas características de las preguntas en un sistema de generación automática de tests. En *conferencia española para la inteligencia artificial*. 1999a.

RÍOS, A., PÉREZ-DE-LA CRUZ, J. L. y CONEJO, R. Siette: Intelligent evaluation system using test for teleeducation. En *Workshop on intelligent tutoring systems on the web, 4th International Conference on Intelligent Tutoring Systems*. 1998.

RÍOS, A., MILLÁN, EVA TRELLA, M., PÉREZ-DE-LA CRUZ, J. L. y CONEJO, R. Internet based evaluation system. En *Artificial Intelligence in Education: Open Learning Environments* (editado por S. Lajoie y M. Vivet), páginas 387–394. 1999b.

ROBERTS, E. y ENGEL, G. Computing curricula 2001: Final report of the joint acm/ieee-cs task force on computer science education. 2001. Disponible en http://www.acm.org/education/curric_vols/cc2001.pdf (último acceso, octubre de 2012).

ROSATELLI, M. y SELF, J. A collaborative case study system for distance learning. *International Journal of Artificial Intelligence in Education*, vol. 14(1), páginas 1–29, 2004.

ROSENBAUM, P. Items bundles. *Psychometrika*, vol. 53, páginas 349–359, 1988.

RUBIO, E., GÁLVEZ, J. y GUZMÁN, E. *OOPS, Object Oriented Programming System*. Proyecto Fin de Carrera, Universidad de Málaga, 2009.

RUPP, A. y TEMPLIN, J. *Diagnostic Measurement: Theory, Methods, and Applications*. Guilford Publications, 2010.

SADLER, D. R. Formative assessment and the design of instructional systems. *Instructional Science*, vol. 18, páginas 119–44, 1989.

SAMEJIMA, F. Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph*, (17), 1969.

SAMEJIMA, F. *The general graded response model*, páginas 77–108. Springer, 2010.

SAMPIERI, R. H., COLLADO, C. F. y LUCIO, P. B. *Metodología de la investigación, 6a ed.*. McGrawHill, 2006.

SANS, R., GÁLVEZ, J. y GUZMÁN, E. *OOPS 2.0: Desarrollo de una nueva interfaz basada en tecnologías de la Web 2.0 para un tutor de Programación Orientada a Objetos*. Proyecto Fin de Carrera, Universidad de Málaga, 2010.

SAVAGE, L. J. *The Foundations of Statistics*. Wiley, 1954.

SCHALK, C., BURNS, E. y HOLMES, J. *JavaServer Faces: The Complete Reference (Complete Reference Series)*. McGraw-Hill Osborne Media, 2006.

SCHNEIDER, M. y STERN, E. The developmental relations between conceptual and procedural knowledge: A multimethod approach. *Developmental Psychology*, vol. 46(1), páginas 178–192, 2010.

SCRIVEN, M. *Perspectives of curriculum evaluation*, capítulo The methodology of evaluation, páginas 39–83. Rand McNally, Chicago, IL, 1967.

SELF, J. Bypassing the intractable problem of student modelling. En *Intelligent Tutoring Systems: at the crossroads of artificial intelligence and education* (editado por C. Frasson y G. Gauthier), páginas 107–123. Ablex, Norwood, NJ, 1990.

SELF, J. The defining characteristics of intelligent tutoring systems research: Itss care, precisely. *International Journal of Artificial Intelligence in Education*, vol. 10, páginas 350–364, 1999.

SHUTE, V. J. y PSOTKA, J. Intelligent tutoring systems: Past, present, and future. En *Handbook of Research forEducational Communications and Technology* (editado por D. H. Jonassen), páginas 570–600. Scholastic Publications, 1996.

SHUTE, V. J. y REGIAN, J. W. Principles for evaluating intelligent tutoring systems. *International Journal of Artificial Intelligence in Education: Special Issue on Evaluation*, vol. 4(2/3), páginas 245–271, 1993.

SIJTSMA, K. y MOLENAAR, I. *Introduction to Nonparametric Item Response Theory*. Measurement Methods for the Social Sciences Series. SAGE Publications, 2002.

SINHARAY, S. y JOHNSON, M. Analysis of data from an admissions test with item models. Informe Técnico RR-05-06, Educational Testing Service, 2005.

SLEEMAN, D. H. y BROWN, J. S. Academic Press, 1982.

SPEARMAN, C. The proof and measurement of association between two things. *American Journal of Psychology*, vol. 15, páginas 72–101, 1904.

SPEARMAN, C. Demonstration of formulae for true measurement of correlation. *American Journal of Psychology*, vol. 18, páginas 161–169, 1907.

STIGGINS, R. y CHAPPUIS, J. What a difference a word makes: Assessment "for" learning rather than assessment "of" learning helps students succeed. *Journal of Staff Development*, vol. 27(1), páginas 10–14, 2006.

SUEN, H. *Principles of test theories*. L. Erlbaum Associates, 1990.

SURAWEERA, P. y MITROVIC, A. Designing an intelligent tutoring system for database. En *Proceedings of 9th Int. Conference on Human-Computer Interaction* (editado por M. J. Smith y G. Salvendy), páginas 745–749. 2001.

SURAWEERA, P. y MITROVIC, A. Kermit: A constraint-based tutor for database modeling. En *Proceedings of the 6th International Conference on Intelligent Tutoring Systems* (editado por S. A. Cerri, G. Gouardères y F. Paraguaçu), vol. 2363, páginas 377–387. Springer, 2002.

SURAWEERA, P. y MITROVIC, A. An intelligent tutoring system for entity relationship modelling. *International Journal of Artificial Intelligence in Education*, vol. 14(3-4), páginas 375–417, 2004.

SURAWEERA, P., MITROVIC, A. y MARTIN, B. The role of domain ontology in knowledge acquisition for ITSs. En *Proceedings of 7th International Conference on Intelligent Tutoring Systems*, páginas 207–216. 2004a.

SURAWEERA, P., MITROVIC, A. y MARTIN, B. The use of ontologies in ITS domain knowledge authoring. páginas 41–49. 2004b.

SURAWEERA, P., MITROVIC, A. y MARTIN, B. A knowledge acquisition system for constraint-based intelligent tutoring systems. En *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, páginas 638–645. 2005.

SURAWEERA, P., MITROVIC, A. y MARTIN, B. Widening the knowledge acquisition bottleneck for constraint-based tutors. *International Journal on Artificial Intelligence in Education*, vol. 20(2), páginas 137–173, 2010.

SURHONE, L. M., TIMPLEDON, M. T. y MARSEKEN, S. F. *Spearman-Brown Prediction Formula*. VDM Verlag Dr. Mueller e.K., 2010.

SWELLER, J., MERRIENBOER, J. V. y PAAS, F. Cognitive architecture and instructional design. *Educational Psychology Review*, vol. 10(3), páginas 251–296, 1998.

TARAS, M. Assessment - summative and formative - some theoretical reflections. *British Journal of Educational Studies*, vol. 53(4), páginas 466–478, 2005.

TARAS, M. Assessment for learning: understanding theory to improve practice. *Journal of Further and Higher Education*, vol. 31(4), página 363, 2007.

TATSUOKA, K. K. *Cognitive Assessment: A Link Between Statistical Pattern Recognition and Classification Problems and Psychometrics: A Link Between Statistical Pattern Recognition and Classification Problems and Psychometrics*. Taylor & Francis, 2009.

THISSEN, D., CAI, L. y BOCK, R. D. *The nominal categories item response model*, páginas 43–75. Springer, 2010.

THISSEN, D., CHEN, W.-H. y BOCK, R. *Multilog (version 7)*. Scientific Software International, Lincolnwood, IL, 2003.

THISSEN, D. y STEINBERG, L. A taxonomy of item response models. *Psychometrika*, vol. 51, páginas 567–577, 1986.

THOMSON, D. y MITROVIC, A. Preliminary evaluation of a negotiable student model in a constraint-based its. *Research and Practice in Technology Enhanced Learning*, vol. 5(1), páginas 19–33, 2010.

THURSTONE, L. L. A method of scaling psychological and educational tests. *Journal of Educational Psychology*, vol. 16, páginas 433–451, 1925.

THURSTONE, L. L. A law of comparative judgment. *Journal of Educational Psychology*, vol. 34, páginas 273–286, 1927.

TISSOT, P. Terminology of vocational training policy: a multilingual glossary for an enlarged europe. 2004.

TYLER, R. W. General statement on evaluation. *Journal of Educational Research*, vol. 35, páginas 492–501, 1942.

VANLEHN, K. Conceptual and meta learning during coached problem solving. En *Intelligent Tutoring Systems* (editado por C. Frasson, G. Gauthier y A. M. Lesgold), vol. 1086, páginas 29–47. Springer-Verlag, New York, 1996.

VANLEHN, K. The behaviour of tutoring systems. *International Journal of Artificial Intelligence in Education*, vol. 16(3), páginas 227–265, 2006.

VANLEHN, K., JORDAN, P. W., ROSE, C. P., BHEMBE, D., BOETTNER, M., GAYDOS, A., MAKATCHEV, M., PAPPUSWAMY, U., RINGENBERG, M., ROQUE, A., SILER, S. y SRIVASTAVA, R. The architecture of why2-atlas: A coach for qualitative physics essay writing. En *Proceedings of 6th International Conference on Intelligent Tutoring Systems*, vol. 2363 de *LNCS*, páginas 158–167. Springer, 2002.

VANLEHN, K., LYNCH, C., SCHULTZ, K., SHAPIRO, J. A., SHELBY, R. H. y TAYLOR, L. E. A. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, vol. 15(3), páginas 157–204, 2005.

VERDEJO, M. F. Building a student model for an intelligent tutoring system. En *Student modelling: The key to individualized knowledge-based instruction* (editado por J. E. Greer y G. McCalla), vol. 125, páginas 147–163. Springer Verlag, New York, 1994.

VIGOTSKY, L. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978.

WAINER, H. Cats: Whither and whence. *Psicológica*, vol. 21, páginas 121–133, 2000.

WAINER, H. y MISLEVY, R. J. *Item response theory, calibration, and estimation*. Taylor & Francis, 2000.

WEBB, N. M. y SHAVELSON, R. J. Generalizability theory: Overview. *Encyclopedia of Statistics in Behavioral Science*, vol. 2, páginas 717–719, 2005.

WEERASINGHE, A. y MITROVIC, A. Enhancing learning through self-explanation. En *Proceedings of 10th International Conference on Computers in Education*, vol. 1, páginas 244–248. 2002.

WEERASINGHE, A. y MITROVIC, A. Facilitating deep learning through self-explanation in an open-ended domain. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 10(1), páginas 3–19, 2006.

WEERASINGHE, A., MITROVIC, A. y MARTIN, B. A preliminary study of a general model for supporting tutorial dialogues. En *Proceedings of 16th International Conference on Computers in Education*, páginas 125–132. 2008.

WEERASINGHE, A., MITROVIC, A. y MARTIN, B. Towards individualized dialogue support for ill-defined domains. *International Journal of Artificial Intelligence in Education*, vol. 19(4), páginas 357–379, 2009.

WEERASINGHE, A., MITROVIC, A., THOMSON, D., MOGIN, P. y MARTIN, B. Evaluating a general model of adaptive tutorial dialogues. En *Proceedings of 15th International Conference on Artificial Intelligence in Education*, páginas 383–390. 2011.

WEERASINGHE, A., MITROVIC, A., ZIJL, M. V. y MARTIN, B. Evaluating the effectiveness of adaptive tutorial dialogues in eer-tutor. En *Proceedings of 18th International Conference on Computers in Education* (editado por S. Wong), páginas 33–40. 2010.

WILLIAMSON, D. M., MISLEVY, R. J. y BEJAR, I. I. *Automated Scoring of Complex Tasks in Computer-Based Testing*. Taylor & Francis, 2006.

WINNE, P. H. A landscape of issues in evaluating adaptive learning systems. *International Journal of Artificial Intelligence in Education*, vol. 4(4), páginas 309–332, 1993.

WINOGRAD, T. *Frame representations and the declarative- procedural controversy*, páginas 185–210. Academic Press, 1975.

WYGANT, R. M. Clips-a powerful development and delivery expert system tool. *Computers in Engineering*, vol. 17(1-4), páginas 546–549, 1989.

WYSE, A. E. y HAO, S. An evaluation of item response theory classification accuracy and consistency indices. *Applied Psychological Measurement*, 2012.

ZAKHAROV, K., MITROVIC, A. y JOHNSTON, L. Pedagogical agents trying on a caring mentor role. En *Proceedings of 13th International Conference on Artificial Intelligence in Education* (editado por R. Luckin, K. R. Koedinger y J. E. Greer), LNCS, páginas 59–66. IOS Press, 2007.

ZAKHAROV, K., MITROVIC, A. y JOHNSTON, L. Towards emotionally-intelligent pedagogical agents. En *Proceedings of 9th International Conference on Intelligent Tutoring Systems* (editado por B. Woolf, E. Aïmeur, R. Nkambou y S. Lajoie), vol. 5091 de *LNCS*, páginas 19–28. Springer Berlin / Heidelberg, 2008.

ZAKHAROV, K., MITROVIC, A. y OHLSSON, S. Feedback micro-engineering in eer-tutor. En *Proceedinds of AIED 2005* (editado por C.-K. Looi, G. McCalla, B. Bredeweg y J. Breuker), páginas 718–725. 2005.

ZEKL, A. y MORSCHEL, I. Embedding authoring support in an its for the learning of object-oriented programming. En *Proc. of IEEE First International Conference on Multi-Media Engineering Education*, páginas 59–64. Australia, 1994.

# Índice alfabético

Dimitrova (2003), 20, 288
Di Eugenio et al. (2009), 34, 288
Doorenbos (1995), 169, 288
Douglas y Cohen (2001), 63, 288
Douglas (1997), 63, 288
Duan et al. (2010), 36, 41, 288
Ehlers (2002), 70, 288
Embretson y Reise (2000), 55, 288
Eubank (1999), 73, 288
Evens y Michael (2006), 21, 289
Fernández y Gálvez (2011), 195, 289
Forgy (1982), 33, 163, 289
Fossati (2008), 34, 289
Franz Inc (1998), 39, 289
Friedman-Hill (1997), 163, 268, 289
Gálvez et al. (2007), 154, 155, 157, 204, 208, 289
Gálvez et al. (2008), 165, 184, 289
Gálvez et al. (2009a), 154, 204, 289
Gálvez et al. (2009b), 154, 211, 289
Gálvez et al. (2009c), 208, 290
Gálvez et al. (2012), 184, 191, 220, 289
Gálvez (2009), 164, 289
Gómez y Guzmán (2006), 154, 290
Gallopoulos et al. (1994), 15, 289
Georgiev (2008), 55, 290
Glaser (1963), 53, 290
Gokhale (1995), 20, 290
Graesser et al. (2005), 21, 290
Guttman (1945), 53, 290
Guzmán et al. (2007), 94, 130, 290
Guzmán y Conejo (2004a), 174, 177, 290
Guzmán y Conejo (2004b), 174, 290
Guzmán (2005), 68, 76, 96, 100, 108–110, 130, 174, 189, 264, 290
Guzmán et al. (2007), 174, 290
Haley (1952), 61, 290
Hambleton et al. (1991), 54–56, 60, 65, 66, 143, 145, 217, 291
Hambleton y Jones (1993), 54, 66, 290
Hartley y Mitrovic (2002), 35, 36, 41, 136, 291
Heinze y Procter (2006), 204, 291
Hemker et al. (1997), 63, 64, 291
Holland et al. (2009), 43, 155, 291
Holland et al. (2011), 37, 42, 291
Holland y Rosenbaum (1986), 63, 291
Holt et al. (1994), 17, 18, 291
Hontangas et al. (2000), 50, 58, 291
Hopkins y Stanley (1981), 8, 291
Hoppe et al. (2007), 20, 291
Huba y Freed (2000), 4, 291
Huebner (2010), 77, 291
Härdle (2004), 73, 291
Inaba y Mizoguchi (2004), 20, 292
Isomorphic Software (2009), 195, 292
Jennrich y Sampson (1976), 70, 292
Jiménez-Díaz et al. (2005), 154, 292
Junker y Ellis (1997), 63, 292
Junker y Sijtsma (2001), 63, 292
Junker (2011), 63, 95, 292
Kay (1997), 20, 292
Khan y Jain (1999), 192, 292
Kleinbaum y Klein (2010), 73, 292
Kodaganallur et al. (2005), 89, 292
Kodaganallur et al. (2006), 89, 292
Koedinger et al. (1997), 23, 24, 34, 293
Koedinger et al. (2004), 24, 292
Koedinger y Aleven (2007), 24, 293
Kuder y Richardson (1937), 53, 293
López et al. (1998), 165, 170, 294
Langley et al. (1984), 20, 293
Lazarsfeld (1950a), 55, 293
Lazarsfeld (1950b), 55, 293
Le et al. (2009), 34, 293
Lee et al. (2008), 125, 147, 266, 293
Lee (2007), 95, 293
Le (2006), 43, 293
Li et al. (2012), 20, 24, 293
Littman y Soloway (1988), 203, 294
Lord y Novick (1968), 51, 52, 55, 294
Lord (1952), 55, 294
Lord (1955), 54, 294
Luecht y Sireci (2011), 50, 294
Mabbott y Bull (2004), 20, 294
Mallery (1994), 39, 294
Martin et al. (2007), 44, 295
Martin et al. (2009), 45, 294
Martin et al. (2011), 128, 135, 143, 295
Martin y Mitrovic (2000), 27, 32, 35, 188, 294
Martin y Mitrovic (2002a), 43, 44, 294
Martin y Mitrovic (2002b), 29, 32, 36, 294
Martin y Mitrovic (2002c), 43, 295
Martin y Mitrovic (2005), 35, 40, 128, 135, 295

Martin y Mitrovic (2006), 35, 40, 128, 135, 142, 295
Martin y Mitrovic (2008), 45, 295
Martin y VanLehn (1995), 20, 295
Marzano (1990), 8, 50, 295
Masters (1982), 59, 295
Masters (2010), 59, 64, 295
Mathews et al. (2008), 33, 41, 295
Mathews et al. (2012), 36, 41, 136, 295
Mathews y Mitrovic (2007), 40, 295
Mayo et al. (2000), 42, 296
Mayo y Mitrovic (2000), 31, 35, 36, 47, 261, 262, 295
Mayo y Mitrovic (2001), 31, 36, 42, 47, 262, 296
McLaren et al. (2008), 24, 296
Menzel (2006), 43, 296
Milik et al. (2006a), 43, 296
Milik et al. (2006b), 43, 296
Millán et al. (1999), 165, 296
Millán et al. (2003), 165, 296
Mills y Dalgarno (2007), 43, 296
Mislevy et al. (2003a), 78, 79, 296
Mislevy et al. (2003b), 78, 83, 297
Mislevy y Riconscente (2006), 78, 296
Mislevy (2011), 78, 79, 81, 83, 296
Mitrovic et al. (2001), 25, 46, 261, 298
Mitrovic et al. (2002), 31, 33, 47, 167, 262, 298
Mitrovic et al. (2003), 47, 89, 90, 261, 297
Mitrovic et al. (2004), 28, 299
Mitrovic et al. (2005), 186, 298
Mitrovic et al. (2006), 45, 299
Mitrovic et al. (2007), 25, 46, 90, 261, 298
Mitrovic et al. (2008), 45, 298
Mitrovic et al. (2009), 45, 298
Mitrovic et al. (2011), 45, 299
Mitrovic y Martin (2000), 33, 167, 297
Mitrovic y Martin (2002), 36, 40, 136, 298
Mitrovic y Martin (2003), 36, 298
Mitrovic y Martin (2004), 30, 35, 40, 298
Mitrovic y Ohlsson (1999), 17, 38, 180, 298
Mitrovic y Ohlsson (2006), 27, 28, 42, 89, 90, 111, 188, 234, 265, 278, 298
Mitrovic y Ohlsson (2007), 28, 112, 265, 298
Mitrovic y Weerasinghe (2009), 15, 38, 39, 153, 154, 164, 299
Mitrovic (1997), 34, 297
Mitrovic (1998a), 27, 28, 186, 188, 297
Mitrovic (1998b), 28, 34, 38, 169, 297
Mitrovic (1998c), 28, 297
Mitrovic (2002), 41, 297
Mitrovic (2003a), 35, 38, 39, 132, 180, 297
Mitrovic (2003b), 42, 297
Mitrovic (2005a), 37, 42, 297
Mitrovic (2005b), 42, 297
Mitrovic (2006), 28, 30, 38, 215, 297
Mitrovic (2012), 25, 34, 38, 39, 43, 46, 48, 111, 261, 262, 265, 297
Molenaar (2001), 63, 73, 299
Morschel (1993), 154, 299
Moss et al. (2006), 201, 224, 238, 271, 281, 299
Muñiz (2003), 51, 299
Muñiz (2010), 52, 62, 66, 299
Muraki (1982), 58, 64, 299
Murray (1999), 3, 21, 299
Murray (2003), 21, 299
Navarrete et al. (1990), 4, 299
Nering y Ostini (2010), 58, 64, 299
Nkambou et al. (2010), 3, 21, 299
Oh et al. (2009), 43, 300
Ohlsson et al. (2007), 34, 300
Ohlsson y Mitrovic (2006), 33, 47, 261, 300
Ohlsson y Rees (1991), 27, 300
Ohlsson (1986), 17, 77, 300
Ohlsson (1992), 25, 90, 260, 300
Ohlsson (1993), 25, 27, 90, 300
Ohlsson (1994), 10, 25–27, 90, 154, 300
Ohlsson (1996), 25, 33, 300
Olea et al. (2010), 50, 300
Olea (2002), 50, 67, 69, 72, 300
Owen (1975), 75, 300
Pani (2007), 7, 300
Pardos y Heffernan (2011), 93, 300
Parshall et al. (2010), 50, 300
Pavlik et al. (2009a), 23, 301
Pavlik et al. (2009b), 23, 93, 301
Petry y Rosatelli (2006), 43, 301
Piaget (1970), 6, 301
Pillay (2000), 155, 301
Polson y Richardson (1988), 3, 301
Ponsoda (2000), 67, 301
Ríos et al. (1998), 174, 302
Ríos et al. (1999a), 174, 302

Ríos et al. (1999b), 174, 302
Ramaprasad (1983), 5, 92, 113, 301
Ramsay (1991), 63, 73, 301
Rao y Sinharay (2007), 61, 64, 301
Rasch (1960), 55, 62, 64, 301
Razzaq et al. (2007), 204, 301
Razzaq et al. (2009), 25, 301
Reckase (2009), 57, 61, 301
Reckase (2010), 68, 302
Reeve y Fayers (2005), 56, 302
Roberts y Engel (2001), 170, 302
Rosatelli y Self (2004), 43, 302
Rosenbaum (1988), 50, 115, 266, 302
Rubio et al. (2009), 157, 302
Rupp y Templin (2010), 57, 302
Sadler (1989), 5, 92, 113, 302
Samejima (1969), 58, 302
Samejima (2010), 58, 64, 302
Sampieri et al. (2006), 201, 302
Sans et al. (2010), 157, 302
Savage (1954), 47, 262, 302
Schalk et al. (2006), 165, 303
Schneider y Stern (2010), 6, 111, 265, 303
Scriven (1967), 4, 5, 92, 113, 303
Self (1990), 17, 303
Self (1999), 3, 303
Shute y Psotka (1996), 3, 18, 303
Shute y Regian (1993), 203, 303
Sijtsma y Molenaar (2002), 62, 63, 303
Sinharay y Johnson (2005), 116, 303
Sleeman y Brown (1982), 18, 303
Spearman (1904), 51, 303
Spearman (1907), 51, 303
Stiggins y Chappuis (2006), 5, 303
Suen (1990), 51, 303
Suraweera et al. (2004a), 44, 304
Suraweera et al. (2004b), 44, 304
Suraweera et al. (2005), 29, 44, 304
Suraweera et al. (2010), 42, 304
Suraweera y Mitrovic (2001), 41, 303
Suraweera y Mitrovic (2002), 41, 303
Suraweera y Mitrovic (2004), 35, 41, 128, 133, 304
Surhone et al. (2010), 53, 304
Sweller et al. (1998), 18, 38, 39, 165, 192, 304
Taras (2005), 4, 5, 92, 113, 304
Taras (2007), 4, 5, 304
Tatsuoka (2009), 78, 304
Thissen et al. (2003), 58, 180, 183, 304
Thissen et al. (2010), 58, 64, 304
Thissen y Steinberg (1986), 63, 304
Thomson y Mitrovic (2010), 36, 41, 136, 304
Thurstone (1925), 10, 55, 304
Thurstone (1927), 55, 305
Tissot (2004), 4, 305
Tyler (1942), 5, 305
VanLehn et al. (2005), 24, 305
VanLehn (1996), 6, 305
VanLehn (2006), 34, 305
Vanlehn et al. (2002), 21, 305
Verdejo (1994), 18, 305
Vigotsky (1978), 35, 132, 163, 305
Wainer y Mislevy (2000), 67, 73, 305
Wainer (2000), 67, 68, 305
Webb y Shavelson (2005), 53, 305
Weerasinghe et al. (2008), 37, 43, 305
Weerasinghe et al. (2009), 37, 41, 43, 306
Weerasinghe et al. (2010), 37, 41, 306
Weerasinghe et al. (2011), 37, 41, 42, 306
Weerasinghe y Mitrovic (2002), 37, 41, 305
Weerasinghe y Mitrovic (2006), 37, 305
Williamson et al. (2006), 77, 306
Winne (1993), 203, 306
Winograd (1975), 6, 306
Wygant (1989), 160, 306
Wyse y Hao (2012), 224, 238, 271, 281, 306
Zakharov et al. (2005), 33, 40, 306
Zakharov et al. (2007), 37, 41, 306
Zakharov et al. (2008), 37, 41, 306
Zekl y Morschel (1994), 154, 306
de Jong y Ferguson-Hessler (1996), 9, 95, 112, 265, 292
van der Linden y Glas (2000), 67, 294
van der Linden y Hambleton (1996), 62, 75, 294
van der Linden y Pashley (2010), 74, 76, 294
van der Linden (1999), 69, 293
van der Linden (2006), 74, 293
van der Linden (2010), 50, 76, 133, 293
AFT (American Federation of Teachers) et al. (1990), 4, 283

*La Naturaleza nos muestra sólo la cola del león.*
*Pero no tengo duda de que pertenece al león,*
*incluso aunque éste no pueda revelarse de una vez*
*ante nuestros ojos debido a su enorme dimensión.*

*Albert Einstein (1879 - 1955)*

*No sé lo que puedo parecer al mundo;*
*pero para mí mismo, sólo he sido como un niño*
*jugando a la orilla del mar, y divirtiéndome*
*al hallar de vez en cuando un guijarro más suave*
*o una concha más hermosa que de costumbre,*
*mientras que el gran océano de la verdad*
*permanecía sin descubrir ante mí.*

*Isaac Newton (1642 - 1727)*