Improving Student Performance Using Self-Assessment Tests

Eduardo Guzmán, Ricardo Conejo, and José-Luis Pérez-de-la-Cruz, *University of Málaga*

Two empirical studies of SIETTE, a Web-based adaptive-testing system, help determine whether self-assessment tests improve student performance on final exams.

Testing is the most generic and perhaps most widely used mechanism for student assessment. Most tests are based on the *Classical Test Theory*,[1] which says that a student's score is the sum of the scores obtained in all questions plus some kind of error. This theory, although simple to apply, has many limitations. The most relevant is that the student test result depends heavily on the individual's learning preferences or abilities and also on the actual test's format. According to this theory, tests aren't necessarily useful in intelligent educational systems, which require accurately obtaining the student's knowledge state to guide the learning process.

Yet the Web has created a new generation of intelligent systems—Adaptive Hypermedia Systems—which offer new types of instructional interaction. Educational AHSs adapt the learning process on the basis of the student's learning preferences, knowledge, and availability. One such Web-based tool is SIETTE (the *s*ystem of *i*ntelligent *e*valua*t*ion using *te*sts), which infers student knowledge using adaptive testing.

We conducted two empirical studies of SIETTE over four academic years for two courses: one on the Lisp programming language and another on language processors. Here we explore whether SIETTE's self-assessment tests improved student performance on the final exams.

SIETTE: An adaptive testing system

In the late '90s, we and several of our colleagues created SIETTE (www.lcc.uma.es/siette).[2] This year, we released the system's third version, which provides new functionalities for both teachers and students.

Teachers at the University of Málaga, the Polytechnic University of Madrid, and the Spanish Open University currently use SIETTE to administer tests for undergraduate and graduate computer science classes, graduate telecommunications and botany classes, and postgraduates studying computer science applied to mobile technologies. Our database contains approximately 91 courses, 2,086 concepts, and 317 tests, and 18,760 students have taken tests using SIETTE.

SIETTE's editing tools help teachers create tests for their courses. The teachers start by creating a hierarchy of the concepts the students must learn, then creating questions that relate to these concepts. These questions then form a bank from which the teachers can pull when creating tests. When creating a test, the teacher must consider these parameters:

- *Accessibility.* Is the test just for certain students (lab students, for example)? Can students use the test to help them study, or is it only for grading purposes?
- *Grading.* Can students see the solutions and, if so, when (after each answer or at the end of the test)?
- *Timing.* Is there a time limit for answering?
- *Assessment scale.* How many knowledge levels does the test cover?
- *Question-selection criterion.* Which criterion is used to dynamically select questions from the bank (for example, fixed, random, or adaptive according to the student's knowledge level)?
- *Test-finalization criterion.* How many questions will be administered to the student (for example, a fixed number or just those needed to estimate the student's knowledge with some statistical certainty)?

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

Several teachers can access the SIETTE Web site for the same course, so they can collaborate in developing its content.

In SIETTE's virtual classroom, students can take tests as graded exams or for self-assessment to evaluate their progress. The self-assessment tests can offer hints with the questions or provide feedback with the answers to rectify misconceptions and reinforce well-acquired knowledge.

SIETTE can function as an independent tool, or you can integrate it into other educational systems. Several tutoring systems combine SIETTE with other tools, and we're collaborating with the MEDEA[3] and ActiveMath[4] tutoring systems as a diagnosis module for updating their student models.

SIETTE's intelligent features

Experts agree that what constitutes intelligence in intelligent tutoring systems is "real-time cognitive diagnosis" and "adaptive remediation."[5] SIETTE aims to perform well in cognitive diagnosis, but we assume that other modules can address adaptive remediation. However, our studies show that mere exposure to a diagnosis tool can potentially have remedial effects.

Our system's intelligence depends on a model we built[6] based on two well-founded bases: the item response theory (IRT) and the computer adaptive testing theory. We can separate this intelligent behavior into two characteristics: assessment and adaptation.

Assessment

Student-knowledge diagnosis systems must fulfill certain requirements to ensure scientific adequacy—particularly when such systems are part of intelligent-tutoring processes.[6] The diagnosis mechanisms must be

- *valid.* They shouldn't depend on the tool or how it's applied.
- *reliable.* The measurement's accuracy must be independent of the features being measured.
- *objective.* The results shouldn't be subject to the observer's opinion or personal perspective.

Teachers tend to use heuristic tests, which provide information about the student's knowledge state but don't provide the required validity, reliability, or objectivity.

An alternative to heuristic-based tests is IRT, which states that we can explain a student's test performance on the basis of his or her knowledge level.[1] IRT can offer valid, reliable, and objective results under certain hypotheses that can be statistically measured. Furthermore, IRT measures this knowledge level as a numeric value (usually on the real-number scale). In addition, IRT advocates that there is a probabilistic relationship between the student's knowledge level and his or her response to a question (called an *item* in IRT)—that is, we can predict the student's probability of answering the item correctly on the basis of his or her knowledge level. Furthermore, we can quantify this relationship in a function called the *item characteristic curve*. Mining ICCs from data sets of students who took tests that included a particular item is called *calibration*.

> The item response theory advocates that there is a probabilistic relationship between the student's knowledge level and his or her response to a question.

For SIETTE, we developed a diagnosis model based on IRT[2] that works with discrete values. Accordingly, ICCs are probability vectors where each value represents the probability of answering the item successfully. SIETTE uses probability distribution vectors to form student models—one for each concept. Each probability-distribution value represents the probability of having the corresponding knowledge level. SIETTE infers probability distributions from the ICCs of answered items. At the end of the test, SIETTE obtains the student knowledge level for a given concept from the corresponding probability distribution by calculating either its statistical mode or expected value. SIETTE measures knowledge levels in an enumerated scale beginning at zero. The teacher can determine the number of knowledge levels—that is, the assessment's granularity—on the basis of the number of categories used to classify the students. By default, SIETTE uses 12 knowledge levels, because this number has several divisors, so we can easily map this scale to other scales with two, three, four, or six knowledge levels.[6]

Using a discrete approach is computationally more efficient than using the classical continuous approach because continuous algorithms require iterative approximation techniques. Moreover, the size of the knowledge level scale doesn't scientifically affect this efficiency. Computational efficiency is important—especially in our case, where tests are administered via the Internet—because it ensures system scalability. In addition, SIETTE provides the student model with more than just a single value representing the student's knowledge state. We provide a vector indicating the probabilities of each knowledge level.

Adaptation

When a teacher orally evaluates a student, he or she initially asks a question of medium difficulty. If the student answers correctly, the next question will be a little more difficult. If the answer is incorrect, the next question will be easier. This continues until the teacher can accurately estimate the student's knowledge.

Adaptive tests aim to reproduce this behavior[7] and thus present one question at a time. The question selection is dynamically decided in terms of the temporary student model, which updates each time the student answers a question. In addition, the test finalization is decided adaptively in terms of the student model's accuracy. Accordingly, in adaptive tests, adaptation affects

- *concept selection.* Unlike classical adaptive testing techniques, SIETTE lets you simultaneously assess several concepts in the same test. SIETTE starts the testing process by first selecting the concept in which student knowledge estimation is less accurate.
- *question selection.* SIETTE then selects the question that it predicts will best estimate the student's knowledge level.
- *test finalization.* After the student answers the question, SIETTE inspects the student model to determine whether all the knowledge distributions are accurate enough to ensure reliable diagnoses. Often, adaptive finalization criteria are combined with other nonadaptive criteria, such as an upper bound on the number of questions. The rationale is to avoid excessive question overexposure and to give all students the sense that they're taking a test under the same conditions. Although adaptive tests accurately infer knowledge levels,

some students (especially those obtaining lower knowledge levels) don't understand why their tests contain fewer questions than those of other students.

Combining adaptive testing with IRT-based inferences contributes (when both are used appropriately) to obtaining valid, reliable, and objective inferences. The knowledge level calculated from an adaptive test is valid because it doesn't depend on the test—that is, if no learning process occurs between two adaptive tests on the same concept, the results we obtain in both should be similar. Another advantage is that each student receives a different set of questions, leading to more reliable evaluations.[7] In addition, this kind of test requires fewer questions than conventional tests, and students don't feel frustrated, because question difficulty is based on their performance.

Adaptive testing's main drawback is that SIETTE must first calibrate the ICCs, which requires huge data sets of students who have previously taken a conventional test with these questions. In SIETTE, we've developed a statistical procedure based on kernel smoothing requiring fewer data sets than classical proposals.[6] Furthermore, adaptive tests need extremely large question pools to ensure proper adaptation. However, our system can administer not only adaptive but also conventional tests, so teachers can first use a conventional test with only a few questions. Then, after collecting a sufficient number of test sessions, teachers can calibrate the questions and from then on use IRT for evaluations.

Adaptive tests

When a student takes an adaptive test, SIETTE first initializes a student model. For each concept assessed in the test, it creates a knowledge-level distribution. Initially, all levels are constant flat distributions (all values have the same probability). Then, the iterative question-administration process begins.

SIETTE selects each question according to the selection criterion used, which uses student-model knowledge distributions to determine the most informative question. If the teacher decides to use the Bayesian approach, SIETTE will automatically select the concepts and questions in the same step. So, for tests that aim to assess multiple concepts simultaneously, the Bayesian selection criterion will select the most informative question related to the concept for which the student knowledge estimation is the least accurate. Therefore, when the test ends, the system infers the student's knowledge level for all concepts with sufficient accuracy. The rationale of this selection criterion is to compute the expected variance of the student model's posterior knowledge distribution. SIETTE performs this for each question as follows:

1. Compute the probability p that the student will correctly answer the question and the probability that the student won't $(1-p)$. We can estimate p by the dot product between the current estimated student knowledge distribution and the ICC (for the current concept being considered).
2. Compute both posterior distributions of the estimated student's knowledge level, applying Bayes rule, supposing that the student correctly answers the question and then supposing that he or she fails.
3. Calculate the variances of the two posterior distributions.
4. Compute the expected posterior variance using the weighted sum of the variances obtained in step 3 and their corresponding probabilities calculated in step 1.

Combining adaptive testing with IRT-based inferences contributes (when both are used appropriately) to obtaining valid, reliable, and objective inferences.

SIETTE selects the question that will provide the more accurate estimation of the student's knowledge level—that is, the question with the minimum expected posterior variance.

The system updates the student model to reflect whether the student answered the question correctly. Either way, it updates only the student knowledge distribution corresponding to the concept the selected question aims to assess; it doesn't change anything related to other concepts.

After the system updates the student model, it checks the finalization criterion—for example, the accuracy-based criterion requires computing each knowledge distribution's variance. If all variances are below a threshold, the testing algorithm stops. Otherwise, the system reapplies the selection criterion. After the test, the system can calculate the knowledge level in different ways—for instance, using the knowledge distribution mode (that is, the most likely knowledge value).

To contribute to a better understanding of how adaptive tests function, consider the following example. A student is going to take an adaptive test involving two concepts, A and B. So, two knowledge distributions (one for each concept) form the student model. To simplify, we also assume that the test question pool comprises only 10 questions: P1–P10. The first five questions assess concept A; the rest assess concept B. All the questions are true/false, so only two answers exist. The assessment granularity is six knowledge levels, from 0 to 5. The threshold for the test finalization is set to 0.21.

Figure 1 shows the evolution of the adaptive test sessions. Figure 1a shows a matrix with the ICC values for each knowledge level. These values have been computed by discretizing a three parameter-based logistic function:

$$p\left(u_i = 1 \middle| \theta\right) = c_i + \left(1 - c_i\right)\frac{1}{1 + e^{-1.7a_i\left(\theta - b_i\right)}}$$

In this function,

- θ is the knowledge level,
- u_i is the student's answer (when $u_i = 1$, the answer is correct),
- a_i is the item discrimination factor (the higher this value, the higher the probability of answering the item),
- b_i is the question difficulty (the knowledge level for which the probability of answering correctly is the same as answering incorrectly), and
- c_i is the question guessing factor (the probability of answering correctly with knowledge level equal to zero).

This function is the most commonly used in IRT for modeling ICCs.

Figure 1b shows the student model—that is, the probability distribution for each concept of the test, and also the variance of these distributions, since this value determines whether to end the test. Below the student model are the answer distributions for six questions.

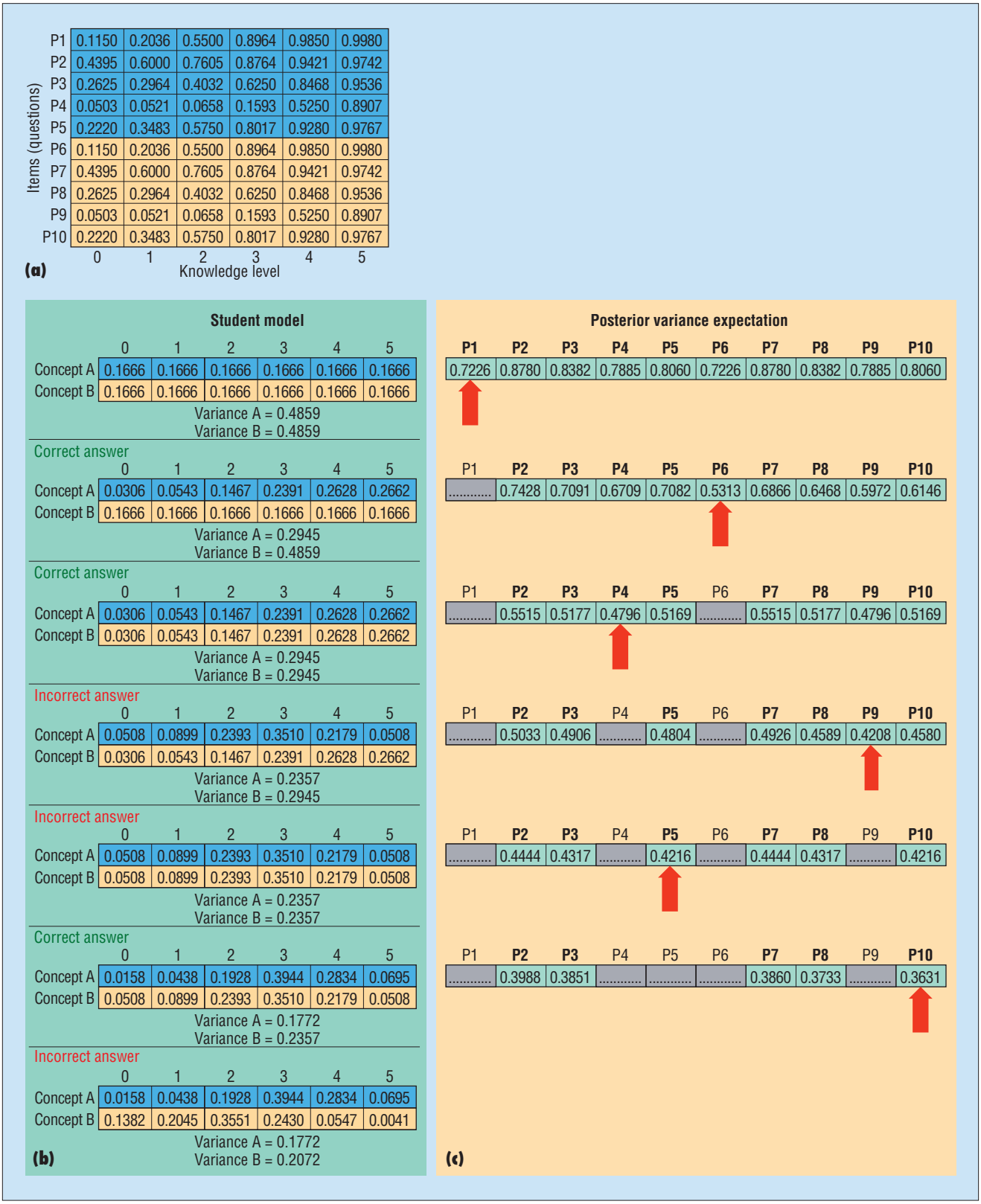

Figure 1. An example of adaptive-testing administration: (a) the Item Characteristic Curve matrix, (b) the student model and the answer distributions for six questions, and (c) the posterior variance expectation. The red arrows show the question that minimizes the expected ICC value.

Table 1. Test results for the Lisp exams.

Exam date	No. of students	Students who passed (%)	Mean	Standard deviation
Dec. 2003	83	57.8	5.759	2.882
Feb. 2004	44	59.1	5.659	2.449
June 2004	9	66.7	6.778	1.986
Sept. 2004	29	62.1	5.828	2.842
Dec. 2004	79	49.4	5.367	1.427
Feb. 2005	54	70.4	6.019	1.078
June 2005	9	33.3	4.222	2.108
Sept. 2005	16	99.0	6.882	2.369
Feb. 2006*	93	67.7	6.441	2.590
June 2006*	28	71.4	6.250	2.171
Sept. 2006*†	24	19.0	4.250	2.642

* Students had access to a self-assessment test prior to taking the exam.
† Students experienced connectivity problems during the exam.

Table 2. The number of valid sessions in the Lisp self-assessment open test.

Open test sessions	No. of users	No. of sessions	Mean	Standard deviation
Before Feb. 2006	103	439	7.349	2.592
Before June 2006	42	276	6.605	2.819
Before Sept. 2006	20	230	6.991	2.642
Total	165	945	6.982	2.688

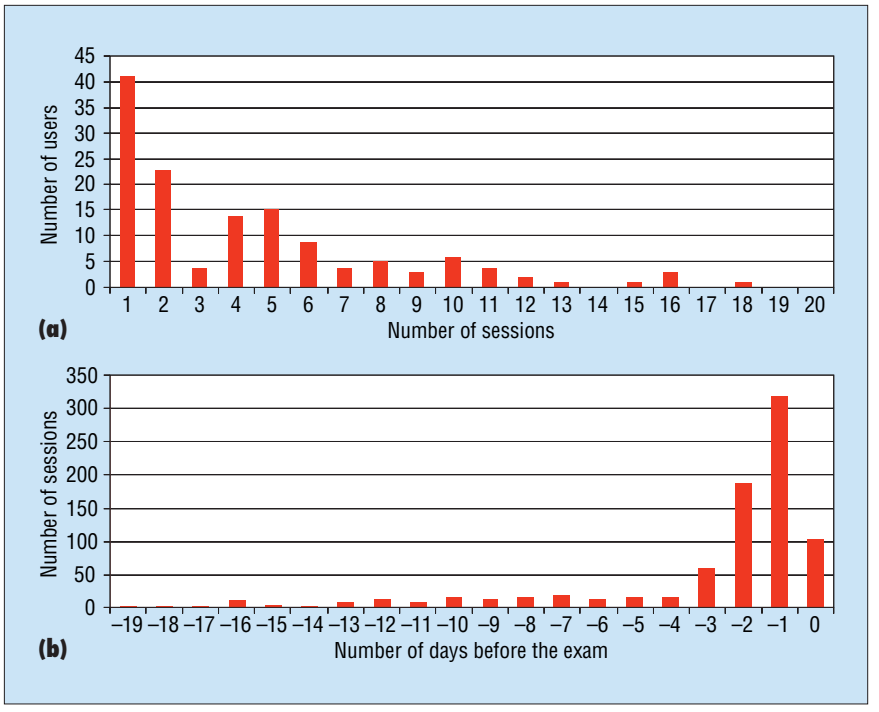

Figure 2. The number of (a) students who took the open test given the number of test sessions and (b) sessions that were taken in relation to the day of the final exam.

Figure 1c shows the calculations needed to adaptively select the most suitable question. The question selected is the one that minimizes this expected value (indicated with a red arrow). Every time SIETTE selects a question, it removes it from this test session's question pool. After the question selection, in the next row, we emulated the student-answering process.

As explained before, SIETTE repeats this process until the threshold is reached. In the example, six questions have been required to infer the student knowledge with the desired accuracy (0.21). Once the test is finished, SIETTE infers the student knowledge level in each concept. In this case, the student knowledge level in concept A is 3 (because the highest probability for the final estimated student knowledge distribution is 0.3944). In concept B, the student knowledge level is 2 (with 0.3551 as the highest probability).

Experiments

The University of Málaga's Artificial Intelligence and Knowledge Engineering (AI&KE) course has used SIETTE for three academic years (since the second half of 2003). The annual course is split into two 14-week semesters and requires that students learn the Lisp programming language. Teachers evaluate students' knowledge of Lisp using a SIETTE-administered test. They teach Lisp during the first semester, from October to February. Students have three opportunities to pass the SIETTE test—in December, February, and June. They can also retake the exam the following September. Only the course's students can access the tests, and only using the PCs located in the school's teaching laboratories. The questions are always multiple-choice with three possible answers, only one of which is correct (students can skip questions).

We've collected 458 test sessions corresponding to the following exam sessions: December 2003; February, June, September, and December 2004; February, June, and September 2005; and February, June, and December 2006. Table 1 shows the percentage of students who passed the test (that is, achieved a knowledge level greater than or equal to 6) and the average knowledge level (on a 0–11 scale) obtained.

During the 2005–2006 academic year, teachers created an open test and made it available on the Internet to AI&KE students to give them a drill-and-practice environment for self-assessment. Teachers selected test questions from a pool of 135 questions, all

of which had appeared in previous tests and whose questions were determined by the teachers. We restricted the maximum number of questions to 20 to avoid overexposure. From January to September 2006, students could access this test freely from their homes. They performed a total of 1,108 test sessions.

To analyze this information, we filtered the student test sessions to eliminate incomplete sessions (4.9 percent) and sessions completed in less than five minutes (13.9 percent). Table 2 contains the filtered data of students who took the open test before the actual exam.

Students made wide use of the self-assessment test. Figure 2a shows the absolute frequencies. A large percentage of students took the open test only once, but certain individuals took it up to 50 times. Also, as figure 2b illustrates, most of the self-assessment test sessions were taken less than three days before the exam date (71 percent). This suggests that students considered the self-assessment test a useful tool to prepare for the exam. Moreover, we should mention that the score obtained in the self-assessment test was higher than in the exams, but we consider this fact meaningless because these tests weren't taken in a controlled environment.

We also analyzed how the self-assessment test influenced student performance for the February, June, and September 2006 exams. We compared the average scores for this year with the average scores in previous years. We consider the comparison to be fair because course content and teachers were the same, and other factors involving the students' cognitive states were similar. We used the classical statistical hypothesis test (independent one-tailed t-test). We can clearly reject the null hypothesis (which states that the means of both samples are the same) in all cases except September 2006.

We discard the last exam because the conditions under which we administered the test were considerably different. That particular day, numerous connection problems created difficulties for the students. Every time the students lost their Internet connection, they had to begin their test again. This likely caused stress for the students, possibly affecting their performance. In fact, the performance in that student sample was the worst we obtained (only 19 percent of individuals passed the exam).

Nevertheless, considering all the data (even from the September 2006 exam), as table 3 shows, the increase in student performance for that year is still statistically significant ($p < 0.01$). If we discard the September 2006 data, we can quantify the improvement in the final results as 8.6 percent (± 9.8 percent) with 95 percent confidence. This is because before the open test, the average number of individuals who passed their exams was 60.0 percent (knowledge level greater than or equal to 6). After introducing the self-assessment test (again, not counting the September 2006 data), an average of 68.6 percent of students passed.

Table 3. Mean comparison and value of p to reject the null hypothesis.

Exam date	No. of students	Mean	Standard deviation	Significance
Feb. 2004 and 2005	98	5.857	2.192	
Feb. 2006	93	6.441	2.590	$p < 0.0098$
June 2004 and 2005	18	5.500	3.009	
June 2006	28	6.250	2.171	$p < 0.0555$
Sept. 2004 and 2005	45	6.203	2.714	
Sept. 2006	24	4.250	2.642	$p < 0.00001$
All				
Dec. 2003–Sept. 2005	323	5.744	2.660	
Feb. 2006–Sept. 2006	145	6.041	2.632	$p < 0.0157$

To determine whether this effect is a consequence of using the self-assessment tests, we conducted a second experiment on a different course corresponding to data from the 2006–2007 academic year. This time, we selected the annual Language Processors (Compilers Construction) course, which requires learning how to use Lex, a lexical analyzer generator. According to the course schedule, teachers explain this concept in November. The students must take two exams that include an assessment test about Lex: one in December and another in February. Both exams are administered under the same conditions as the Lisp exam. Table 4 shows the results.

Table 4. Test results for Lex exams.

Exam date	No. of students	Mean	Standard deviation
Dec. 2006	68	6.397	2.280
Feb. 2007	91	7.418	2.574

Then we created an open test of Lex for student self-assessment. We limited the number of sessions to a single session for a day (that is, students could take the open test only once a day), and we made the test available only the last week before the exam. The purpose of these restrictions was to try to reduce the noise in the data collected. Those students who had previously supplied their email addresses were offered the possibility of taking the self-assessment open test voluntarily prior to the February test. This makes a semirandom division between an experimental and a control group. A completely random experiment is difficult to achieve in this case, because we must guarantee the same opportunities to all students, and self-assessment tests are taken at home in uncontrolled conditions. However, we compared the results obtained by the experimental and control group in the first test of December and in the second test of February (see table 5).

Not all students took both tests, so the numbers vary from one comparison to another. Both groups performed similarly on the December test, where no previous self-assessment tests were made available. However, the experimental group performed much better in February than the control group, and the results are statistically significant.

In addition, we made a pairwise comparison to determine whether the students of the control group increased their performance from the December to the February test. To this end, we studied the difference in the score obtained in both tests. It's positive and statistically significant in the experimental group, and positive (but not significant) in the control group (see table 6). In any case, the experimental group increased its performance much more than the control group ($p < 0.00021$). Concerning the percentage of students passing both exams (assuming both tests are equally difficult), we can quantify the absolute improvement as 27.6 percent (± 15.3 percent) for the experimental group with the standard 95 percent confidence. The

Table 5. Comparison between experimental and control groups for Language Processors course.

Exam group	No. of students	Students who passed (%)	Mean	Standard deviation	Significance
Dec. 2006 control group	20	65.00	6.550	2.188	
Dec. 2006 experimental group	41	68.29	6.293	2.380	$p > 0.55$
Feb. 2007 control group	42	64.29	6.190	2.805	
Feb. 2007 experimental group	49	95.92	8.469	1.804	$p < 0.00000$

Table 6. Pairwise comparison between the control and experimental groups.

Exam group	No. of students	Mean	Standard deviation	Significance (mean < 0)	Significance (t-test)
Dec. 2006 and Feb. 2007 control group	20	+0.950	2.704	$p < 0.13$	$p < 0.00021$
Dec. 2006 and Feb. 2007 experimental group	41	+2.439	2.346	$p < 0.0000001$	$p < 0.00021$

control group's performance decreased, but the amount was negligible (0.71 percent).

Finally, we also studied the relationship between the exam scores and the number of self-assessment test sessions, considering the self-assessment test's minimum, maximum and average scores. We found a small positive but no significant linear correlation. Accordingly, we want to conduct further research.

Our results are promising, and student feedback has been positive. In the future, we plan to continue collecting data from new exam sessions to verify what we have suggested—which is that the more data we will have, the more significant improvement for all knowledge levels we should obtain. In addition, we would like to introduce further instructional features in our self-assessment tests by means of feedback, to guide the instructional process correctly. We intend to compare SIETTE with other testing environments that might be freely available to students. We also plan to enable or disable some of its features to study the real effectiveness in learning benefits. ⬒

The Authors

Eduardo Guzmán is an assistant professor of computer languages and systems at the University of Málaga. His research interests include intelligent tutoring systems, student modeling, and adaptive testing. He received his PhD in computer science from the University of Málaga. Contact him at the Univ. de Málaga, Departamento de Lenguajes y Ciencias de la Computación, Bulevar Louis Pasteur, 35, Campus de Teatinos, 29071, Málaga, Spain; guzman@lcc.uma.es.

Ricardo Conejo is an associate professor of computer languages and systems at the University of Málaga. He is the director of the IAIA (Research and Applications of Artificial Intelligence) research group. His research interests include intelligent tutoring systems, student modeling, and adaptive testing. He received his PhD in civil engineering from the Polytechnic University of Madrid. Contact him at the Univ. of Málaga, Departamento de Lenguajes y Ciencias de la Computación, Bulevar Louis Pasteur, 35, Campus de Teatinos, 29071, Málaga, Spain; conejo@lcc.uma.es.

José-Luis Pérez-de-la-Cruz is an associate professor of computer science and artificial intelligence at the University of Málaga. His research interests include heuristic search, multiagent systems, and engineering and educational applications of AI. He received his PhD in engineering from the Polytechnic University of Madrid. Contact him at the Univ. of Málaga, Departamento de Lenguajes y Ciencias de la Computación, Bulevar Louis Pasteur, 35, Campus de Teatinos, 29071, Málaga, Spain; perez@lcc.uma.es.

References

1. S.E. Embretson and P. Reise, *Item Response Theory for Psychologists*, Lawrence Erlbaum, 2000.
2. R. Conejo et al., "SIETTE: A Web-Based Tool for Adaptive Testing," *J. Artificial Intelligence in Education*, vol. 14, no. 1, 2004, pp. 29–61.
3. M. Trella et al., "An Educational Component-Based Framework for Web-ITS Development," *Proc. Int'l Conf. Web Eng.* (ICWE 2003), LNCS 2722, Springer, 2003, pp. 134–143.
4. E. Melis et al., "ActiveMath: A Generic and Adaptive Web-Based Learning Environment," *Int'l J. Artificial Intelligence in Education*, vol. 12, no. 1, 2001, pp. 385–407.
5. V.J. Shute and J. Psotka, "Intelligent Tutoring Systems: Past, Present, and Future," *Handbook of Research for Educational Communications and Technology*, D.H. Jonassen, ed., MacMillan, 1996, pp. 570–600.
6. E. Guzmán, R. Conejo, and J.L. Pérez-de-la-Cruz, "Adaptive Testing for Hierarchical Student Models," *User Modeling and User-Adapted Interaction*, vol. 17, no. 1, 2007, pp. 119–157.
7. H. Wainer, *Computerized Adaptive Testing: A Primer*, Lawrence Erlbaum, 1990.