RÅDET FÖR HÖGRE UTBILDNING
COUNCIL FOR THE RENEWAL OF HIGHER EDUCATION

Project number: 136/00
Name: Torbjörn Jonsson
Institution: Dept. of Computer and Information Science
Linköping university
S-581 83 Linköping
Tel +46 (0)13 28 24 67
E-mail: torjo@ida.liu.se

Authentic examination for programming courses

Abstract

Project members: Torbjörn Jonsson, Pouria Loghmani and Simin Nadjm-Tehrani. Dept. of Computer and Information Science, Linköping University

This project deals with a new pedagogical view in programming courses, irrespective of the programming language, type of student and educational program. The idea is based on extensive studies around different examination forms, where individual grading, efficient and useful feedback and the authenticity of the examination form are used as basic criteria for the choice of examination method. We believe that the choice of method together with the added efficiency in the assessment process improves the quality of our programs.

We propose to implement and evaluate a computerised examination system that will enable a large number of students (in our tests approximately 100) and a number of teachers (in our tests up to 5) to take part in an on-line examination process simultaneously, communicate, take care of the student queries on-line, and so on.

After the first large scale trial of the system in a real setting we intend to evaluate the system in the following ways:

* Comparing the pass rates and grades of the computerised exam with a traditional one will be used to analyse the impact of the new method on formally measurable student achievements.

* Using evaluation forms after the exam to find out how the students experienced the examination situation as such, and compare the teacher communication and feedback obtained with their earlier (traditional) exams.

Our goal is to finish the implementation part of the examination system (the computerised environment implementation started in the Masters Thesis of Håkan Oswaldsson) and then evaluate this form of final examination in programming courses.

Final report summary

In general we can still see that programming courses have examination on paper instead of computer-aided exams. The laboratory work is done at the computer. Why not at the exam? The most authentic form of examination for programming courses should be at the computer.

An authentic examination must be distinguished from an automatic examination. In an automatic examination the involvement from the teacher is minimal, but in an authentic examination the examiner still has to make the exam. We don't think that a program is correct only if it computes the right result. In the first courses an important question is also how the program is written.

In an authentic examination system (AES) the students use the computer to create a program, test it and then send it to the examiner(s). The examiners test and look at the program. In our AES the examiner sets a preliminary result (PASSED, INCOMPLETE or FAILED). If INCOMPLETE is returned the student can make a new exam request on the same question. After a PASSED result the examiner sets a minimal approval level (a grade) for the student. This can be upgraded later in the exam.

An important question during an exam is that the student should be anonymous. In our AES we use a generated number as the identification of the student. This makes the student anonymous to the examiner and personal relations cannot be added to the examination process.

Another important aspect is that students and examiners are on-line. This gives the student the opportunity to send questions during the whole exam and not only at a few occasions as in a traditional paper based exam.

In a study made during the last year we found that the responses from students indicated an overwhelming support for this examination form. The study was made with "Industrial Management Engineering" and "Technical Biology" students (not "Computer Science" students!). One of the best things was that they had the grade already when they left the exam (student comments).

Evaluation of an Authentic Examination System (AES) for Programming Courses

Torbjörn Jonsson, Pouria Loghmani and Simin Nadjm-Tehrani
Department of Computer and Information Science
Linköping University, Sweden
{torjo,poulo,simin}@ida.liu.se

Abstract

This paper describes our experience with an authentic examination system for programming courses. We briefly describe the architecture of the system, and present results of evaluating the system in real examination situations. Some of the factors studied in detail are the on-line interactions between the students and examiners, the response times and their effects on the pressure experienced by student, the acceptance of the method among the students, and whether the examination form is gender-neutral.

Introduction

As experienced teachers in programming courses we have noticed the drawbacks in the traditional examination form used in programming courses. The students learn to program via laboratory exercises, but the final evaluation of their abilities and the grading of the examination are in a form that uses paper and pen instead of computers. Considering that the student will never use this mode for producing a program through the professional life, we consider this to be not a suitable method.

At the Department of Computer Science at Linköping University 12 fundamental programming courses for approximately 1000 students in different educational programs are taught annually. This paper deals with a new pedagogical view in these programming courses, which can be applied to any programming language, type of student and educational program. The idea is based on extensive studies around different examination forms, where individual grading, efficient and useful feedback and the authenticity of the examination form are used as basic criteria for the choice of examination method. We believe that the choice of method together with the added efficiency in the assessment process improves the quality of our study programmes. In particular, we believe that it will change the examination process from a summative to a normative assessment occasion [1].

For a number of years we have experimented with testing the students via computer-aided examinations in some pilot courses – an authentic examination form for this type of course. However, this examination form has not become more widespread due to insufficient support for the computer environment necessary for this kind of examination. During the past year a new authentic examination system (AES) has been developed, where all the students and the examining teachers are connected to the same system. The process, including communication and grading, is supported by this environment. In this paper we describe the examination system and our initial evaluations of this system in a number of relatively large examination sessions. The courses in question covered programming in Ada and were taken by first and second year students.

During the past year we have evaluated the AES. The instruments used for the evaluation consisted of questionnaires filled by 231 students over a period of 3 months and 4 examinations.

The paper is organised as follows. In section 1 we describe why the type of examination we propose is the most appropriate for programming courses and compare to some related systems. Section 2 includes a brief technical description of the examination systems, including its architectural design. In section 3 we describe how the computer system, that manages the examination process on-line, has to be augmented by rules set up in each particular course. Section 4 covers our evaluation methods and is followed by evaluation results in section 5. Section 6 concludes the paper.

1 Examination forms

Every examination method has specific characteristics that make it more or less appropriate to a particular course setting. Håkan Oswaldsson studied the range of possible examination forms for a typical programming course prior to the development of the current examination system in our department [5]. While several modes of examination can be considered as effective means for enhanced learning (e.g. home assignments, oral examinations following a design assignment, etc), there are not many examination types that combine the need for a summative assessment, with adequate feedback to induce learning. Combined with the large number of students that we are currently teaching, design of an ideal examination setting is a truly challenging task.

The work by Dawson-Howe is an early attempt to bring computer support into the process of programming assignment evaluation and administration [2]. The need for automated examination systems has become more pertinent during the late 90's with the advent of distance and life long learning. For example, at the Open University in UK there have been attempts to exchange student assignments, and their (subsequent) correction and assessment by examiners via MS Word documents [8]. However, the available reports (e.g. the work by Price and Petre) concentrate on the ease of administration for course assignment and grading, rather than the pedagogical feedback in an on-line authentic examination. In recent years several authors report on automatic assessment systems, mostly concentrating on presentation of the technical aspects of the system and the results of the students in terms of grading [4, 5, 7, 8]. While we share the aspiration of these research teams and conduct similar studies, our focus has been on the formal evaluation of how the students perceived the examination environment. In addition we have studied how they were affected by factors specific to authentic examinations, how the system performance and the examiners' on-line behaviour affects the perceived load on the student, and other such aspects.

2 Technical description of the AES

AES has been developed using the J2EE platform. This represents a single standard for implementing and deploying complex enterprise applications. Having been designed through an open process, J2EE meets a wide range of enterprise application requirements, including distribution-specific mechanisms such as messaging system, scalability and modularity.

The clients are based on the Model-View-Controller (MVC) application architecture, which separates three distinct forms of functionality within the application:

- The Model represents the structure of the data in the application, as well as application-specific operation on data.
- The View accesses data from the model and specifies how that data should be presented. Views in the AES consist of stand-alone applications that provide view functionality.
- The Controller translates user actions on the model and selects the appropriate view based on user preferences.

The AES is designed as a set of loosely coupled modules, which are tightly coupled internally. Grouping functionality into modules provides integration between classes that cooperate, yet decouples classes that refer to each other occasionally. Modular design supports the design goal that software will be reusable. Each module has an interface that defines the module's functional requirements and provides a place where later components may be integrated. The AES includes modules for:

- Student accounts
- Teacher accounts
- Exams
- Examination Processing
- Messaging
- Statistics

The AES design is divided into multiple tiers: the Client tier, the Middle tier (consisting of one or more sub-tiers), and the Backend tier (see figure 2.1). Partitioning the design into tiers allows us to choose the appropriate technology for a given situation. Multiple technologies can even be used to provide the same service in different situations. For example, HTML pages, JSP pages, and stand-alone applications can all be used in the client tier.

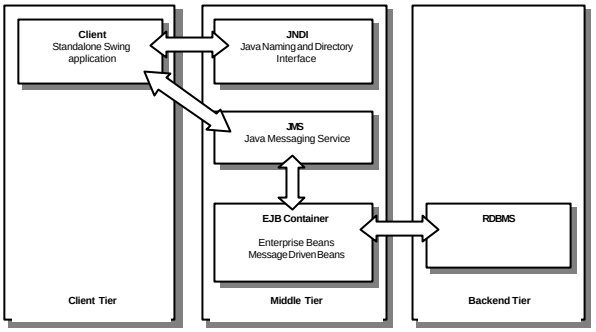

Figure 2.1: The AES design.

Each of the three tiers plays a specific role in the design.

The Client tier is responsible for presenting data to the user, interacting with the user, and communicating with the other tiers of the system. In this case the Client tier is the only part of the system visible to the user. The AES Client tier consists mainly of a stand-alone application that communicates with the other tiers through well-defined interfaces. A message-oriented approach based on JMS (Java Messaging System) has been chosen to take care of the communication between the Client tier and the Middle tier.

The Middle tier is responsible for any processing involving Enterprise JavaBeans. Enterprise JavaBeans are software components that extend servers to perform application specific functionality. The interface between these components and their containers is defined in the Enterprise JavaBeans specification. The containers provide services to the Enterprise JavaBeans instances they contain, such as controlling transactions, managing security, thread or other resource pooling, and handling persistence, among other high-level system tasks.

The Backend tier is the system information infrastructure. This tier includes one or more relational database management systems and potentially other information assets that could be useful, e.g. the central university course results administration system (LADOK). The EIS tier also enforces security and offers scalability. The Backend tier provides a layer of software that maps existing data and

application resources into the design of AES in an implementation-neutral way.

The system is separated into five different functional layers, each with its own responsibilities and its own API. These layers are physically split across the three different tiers. The persistence layer, for example, provides the mechanisms necessary to permanently save object state.

It provides basic CRUD (create, read, update, delete) services and also deals with the object-to-relational mapping issues. This leads to a more flexible and maintainable system, e.g. layers can be changed with no effect on other layers, as long as the API remains constant.

3 Examination set-up

The examination system is only one part of the examination process. The second part is the set-up (the rules) we have for the students. We have tried a few set-ups over a number of years (using a prototype for the system for 5-6 years).

3.1 The first set-up

The first version allowed the students to write the programs using a computer instead of writing on paper. We found this method to be an improvement because we did not have to read "illegible" texts and the submitted solutions could be tested afterwards. Grades were based on the number of correctly solved exercises.

A problem with this set-up was that all the grading still had to be done after the exam was finished. Most of the students waited to send in the solutions until the last minute of the exam.

3.2 The second set-up

Our intention was to have an examination where the students should have a response from the examiner(s) within a few minutes and where grades were given to the students when they left the exam. We also intended to provide the student with the possibility of getting a response for each exercise within a few minutes, so they could correct a nearly correct solution.

The second set-up (which we use today) is based on both number of correctly solved exercises and the amount of time taken to solve them. A number of deadlines are given. If the student wants a high grade he/she has to solve a number of exercises within a pre-specified time limit.

The current examination process follows a few steps:

1. The student sends an examination request for an exercise to the examiner(s).
2. The examiners can return one of the following results.
   - **Passed** - the solution is correct.
   - **Incomplete** - the solution has errors, and must be corrected. It's possible to make a new attempt later.
   - **Fail** - the solution is incorrect and the student is not allowed to continue to work on this exercise.
3. Every examination attempt and the result will contribute to the final exam grade, and the student is informed of his/her current grade. If the student submits a new examination request on an additional exercise he/she can reach a higher grade.

This examination process is built into our current AES, but the rules (time limits etc.) can be changed for separate courses. This makes the system flexible.

Time limits and grading

In the courses this system was tested there were three exercises in each exam and the requirements for different grades were:

- For the grade 5 (excellent) the student must complete:
  - 3 exercises correct in 3 hours or
  - 2 exercises correct in 2 hours
- For the grade 4 (very good) the student must complete:
  - 2 exercises correct in 3 hours or
  - 1 exercise correct in 1.5 hours
- For the grade 3 (passed) the student must complete:
  - 1 exercise correct in 4 hours

The above set-up together with the AES support gives us the opportunity to grade the students during the exam. Students who have solved an exercise are informed of the grade they have reached. If they are satisfied with that grade they can leave the exam (many students leave after one to two hours when they have grade 4 or 5).

Student questions

In an ordinary computer-aided exam, a number of questions are submitted by the students, where the answer can either be classified as personal or as interesting for all students. The examiner can decide if he/she will send the answer to the whole group of students or just to a specific student. The number of questions seems to be relatively constant during the exam (approximately 2-5 questions per 5 minutes). Most questions are sent in during the beginning of the exam, which can be explained by the fact that the students ask about specific things pertaining to the exercises and that there are more students in the beginning of the exam.

Submission/approval attempts

In an ordinary computer-aided exam we have a large number of examination requests from the students. As we can see in figure 3.1 we have a relatively high frequency in the period from 30 minutes to 3 hours. After that, most of the students leave (they can't get a higher grade than 3 after that time).

Around the deadlines we can see that the examination attempts appear more often, but not significantly more often. Still, the increase of examination requests leads to more work for the examiners. This can result in an increase in the response time (waiting time for the student).

4 Evaluation methods

The development of the current system started in summer 2001 and continued through winter 2001/2002. When we began testing this system we wanted as a test example a course with a large number of students. One of our introductory courses in programming has around 270 students each year, so that was our first choice. Approximately 180 of these students are Industrial Management Engineering students and the rest are Technical Biology students. Our statistics are based on their first examination in this course, which took place in March 2002.

We also used a retake exam in this course to do a new study with a new set of questions. This evaluation was done in May 2002.

In these two studies, students filled in questionnaires directly after the exam. The final questionnaire had two parts. The first part was mainly questions where answers are in free text format. The second part included questions with scaled answers (grade on to five, disagree - agree, worse - better). The first part was used in three evaluations. The more extensive questionnaire with two parts was used only for the last evaluation (i.e. for the two last exams). The appendix shows the final questionnaire.

Both types of questionnaires were anonymous and the questionnaires were filled in *after* the grading was done for the exams. The students had already received their grades when they filled in the questionnaires. We believe that this provides a measure of objectivity on the student side.

We also used the log files from the AES for the exams to get statistical trends about grades, gender, response times for questions respectively approval attempts among others (see section 5).

5 Evaluation results

Unfortunately almost all students had no previous experience with paper based programming examinations, so the replies could not be used for comparisons with that examination form. However, we used the response to study other questions in detail (specially the part related to the time/stress factor).

First, how often the students sent in a request (questions or approval attempts), and how long the time for a response was? Secondly, how well was the examination system accepted by the students? A third question was a comparison by grades between the genders.

The response rate of the questionnaires was quite good. We had four exams during the evaluation period with the following response rates:

Exam 1: 76 answers of 112 students (67.8 %)
Exam 2: 87 answers of 105 students (82.8 %)
Exam 3: 50 answers of 66 students (75.7 %)
Exam 4: 18 answers of 22 students (81.8 %)

The first three questionnaires were done at the first examination occasion for the students and the fourth one was done in a retake examination where all the students were students with no grade from an earlier exam.

5.1 Events during an examination

The number of events, questions and examination requests, spread over an examination session of 4 hours can be an interesting metric to look at. The major negative factor that was indicated in the questionnaires was the feeling of time pressure or stress. 17% of the free text answers had some connection to this factor. From a technical point of view we were also interested in finding that the capacity of the system was adequate. Therefore we have summarised the number of interactions taking place in every exam.

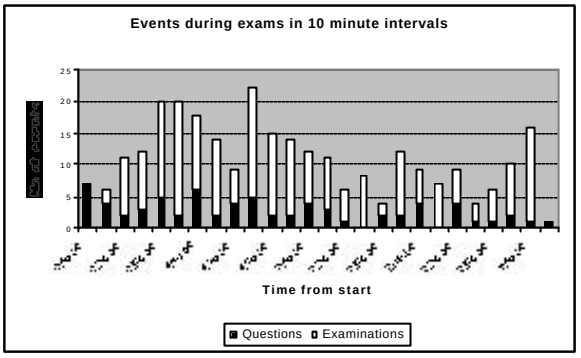

Figure 5.1: Student events (questions and examination requests) during an exam.

In figure 5.1 we can see that the number of questions is higher in the beginning of an examination, but we have question events over the whole examination time.

The number of examination requests is relative to time. There were a few requests in the first half an hour and that the first two hours are busy for the examiners. The request rate is quite high when we reach the time limits for the grades (especially the 4 hour limit).

From a technical point of view the system performance under the above loads has been adequate. To study the student experience of stress due to waiting time we have calculated the average waiting for the answer to a question and an approval of an examination request respectively. We have also looked at the extreme values.

It turns out that for a question the shortest answering time was 30 seconds and the longest 6 minutes. The corresponding figures for approval attempts were 1 minute and 10 minutes respectively. The first type of interaction took 2 minutes and 42 seconds, and the second type 3 minutes and 31 seconds on average for one particular exam.

The student responses, from the questionnaires, on this amount of time is that it is acceptable to wait a minute or

two for an answer on a question and that a few minutes waiting for a result on an examination request is all right.

Based on this view we conclude that waiting time is not a contributing factor to the stress experienced by the students.

5.2 Acceptance by students

The student responses indicated an overwhelming support for this examination form. 94.5% of the students who returned the questionnaire preferred this examination form to a traditional paper and pencil exam.

Many free text answers referred to the examination form being close to a realistic scenario and were positive about the possibility to compile and test (a total number of 94 such comments).

In the exam where quantitative questions about the examination form were added to the questionnaire, 16 of 17 students answered that this form was closer to a realistic situation compared to other examination forms. The majority of students considered themselves to be anonymous with respect to examiners during the exam.

5.3 Grade comparisons (male-female)

We have made a comparison of grades in the first examination between the male end female groups of the students in a course. The numbers we use are normalised so we can compare the figures directly.

As shown in figure 5.2, the grades for the female students are on average lower than the grades for the male students. We were interested in this metric to find out whether the

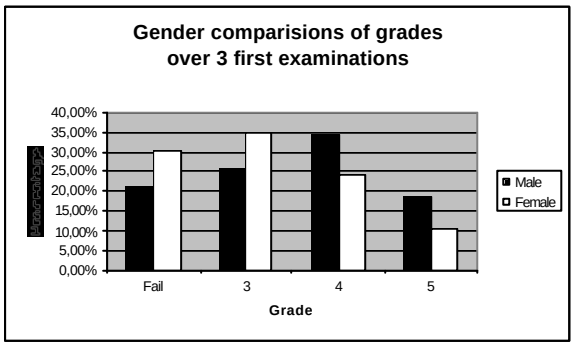

Figure 5.2: Grades related to gender of students.

examination form is gender-neutral. As it turns out we cannot draw this conclusion. However, one possible explanation is that most of the students who have programmed prior to taking the course are male.

Another aspect of the differences in grades could be that we have two different groups of students in this course where the group with a large proportion of female students (Technical Biology) reads the course during their first year and the other group is reading the course during their second year. The students in the second year are likely to have better study habits and are more experienced and have more theoretical knowledge.

A third aspect is that the group with a higher ratio of female students only has this programming course as obligatory in the whole educational programme. The other group of students has more courses in programming afterwards and are possibly more motivated to study and reach higher grades in this course.

This question is an obvious point for further study.

6 Conclusions and ongoing work

This paper has summarized an early experience with an authentic examination system for programming courses. The current formal evaluations of the examination system and the examination setting has provided us with a number of insights on the effectiveness of the system as a tool for learning and for assessment. While the initial evaluations are positive and point towards the success of this examination method for majority of the students, the input from the students opens up new directions for research, and new ideas on how to improve the environment.

Future directions of work are the integration of a new automatic correction system into our on-line and off-line student evaluations, and the exposing of the environment to larger number of students, specially those that already have paper and pencil exam experiences.

References

[1]J. Biggs, Teaching for Quality Learning at University, Open University Press, 1999.

[2]K.M. Dawson-Howe. Automatic Submission and Administration of Programming Assignments. SIGCSE Bulletin, 27(4), December 1995.

[3]J. English, Experience with a computer-Assisted Formal Programming Examination, Proceedings of ITiCSE 2002, p51-54.

[4]C. Higgins, P. Symeonidis, and A. Tsintsifas, The Marking System for CourseMaster, Proceedings of ITiCSE 2002, p46-50.

[5]L. Malmi, A. Korhonen, and R. Saikkonen, Experiences in Automatic Assessment on Mass Courses and Issues for Designing Virtual Courses, Proceedings of ITiCSE 2002, p 55-59.

[6]H. Oswaldsson, Development of an examination system. Masters Thesis LiTH-IDA-Ex-00/73, Dept. of Computer and Information Science, Linköping University, September 2000.

[7]A. Pardo, A Multi-Agent Platform for Automatic Assignment Management, Proceedings of ITiCSE 2002, p60-p64.

[8]B. Price and M. Petre, Teaching Programming through Paperless Assignments: an empirical evaluation of instructor feedback. Technical report, Open University, January 2001.

Appendix: Example questionnaire

Previous exam types

Have you ever taken a written exam in a programming course before?

Is this the first time you have taken a computer-based exam?

Would you prefer a regular written exam instead?

Classify comparison to traditional paper exams: Worse Equal Better

Possibility to ask questions during the exam

Possibility to redo a question during the exam

Possibility to learn something during the exam

Anonymity of exam correction

Testing critical thinking, not just memorisation

Possibility to test and evaluate your own programs

Disturbances during the exam

I can show my best side in theoretical questions

I can show my best side in practial questions

The examination form is not gender-biased

The exam time in relation to the number of problems

Stress level before the exam

Stress level during the exam

Stress level after the exam

Unsure as to if you have correctly answered a problem or not

Unsure about what grade you have received

The exam environment is similar to a real situation

The exam generally reflects the course content

About computer-based exam: Disagree - Agree (grade 1-5)

The exam form made it easy to ask questions during the exam

I received answers to my questions quickly

The result from my solution submission was returned quickly

I could see immediately whether or not I had passed the exam

I learned something about the subject during the course

I felt my anonymity was ensured

Testing my program helped me in solving the exam questions

The responses I received after asking a question and/or submitting a solution helped me understand the problem better

The exam rules: Disagree - Agree (grade 1-5)

I felt relaxed before the exam

I felt relaxed during the exam

I felt relaxed after the exam

It was helpful to be allowed to correct rejected solutions during the exam

It was helpful to get my test result back immediately

The cutoff for a 3 (1 correct solution, 4 h) is acceptable

The cutoff for a 4 (1 correct solution, 1.5h / 2 correct solutions, 3 h) is acceptable

The cutoff for a 5 (2 correct solutions, 2h / 3 correct solutions, 3 h) is acceptable

It was helpful to have access to the course literature during the exam

The interface: Hard - Easy (grade 1-5)

What was it like to communicate using the interface?

How did you like the presentation of grades etc.?

What was it like to ask a question?

What was it like to submit a solution?

Stability (classify within the following intervals):

How many times did you need help in understanding how the system works? >9 4-9 0-3

How many times did a system-interaction window accidently get lost? >9 4-9 0-3

How many times did the system crash? >2 1-2 0

Miscellaneous (Free text answers)

Is there any information you think is missing from the exam system? Please explain.

Other comments