Embedded Systems Exams With True/False Questions: A Case Study

Bettina Weiss, Günther Gridling, Christian Trödhandl
Institute of Computer Engineering
Embedded Computing Systems Group E182-2
Vienna University of Technology
Treitlstrasse 3/182-2
Vienna, Austria
{*bw,gg,troedhandl*}*@ecs.tuwien.ac.at*

Wilfried Elmenreich
Institute of Computer Engineering
Real-Time Systems Group E182-1
Vienna University of Technology
Treitlstrasse 3/182-1
Vienna, Austria
wil@vmars.tuwien.ac.at

ABSTRACT
With increasing student numbers and decreasing budgets, undergraduate embedded systems education needs to turn to Computer-aided Assessment (CAA) techniques to keep down the time and effort invested into grading. When setting up the mandatory embedded systems courses for the bachelor study "Computer Engineering" at our university, we thus payed particular attention to employ CAA-compatible exam styles which ultimately allow courses to be conducted in a distance learning setting. In this paper, we present our theory exam style, which mainly consists of true/false questions, describe our experiences with the exams, motivate our changes to improve student acceptance, and discuss possible improvements of our current exam style.

KEY WORDS
Computer-aided assessment, multiple-choice, true/false questions, embedded systems education

1 Introduction

Some years ago, the Vienna University of Technology introduced a new bachelor study "Computer Engineering", which consists of several courses in the field of embedded systems. Since most of these courses were new and had to be set up, the project SCDL "Seamless Campus: Distance Labs"[1] was initiated to develop a remote teaching concept for embedded systems laboratory courses. The long-term goal of the project is to provide a remote learning environment that can be used for most of the embedded systems courses at the Vienna University of Technology. This will enable us to increase our training capacity and to better support working or handicapped students.

Since we expected at least 100 students per course, we decided to incorporate computer-aided assessment from the beginning to support distance learning. Although the majority of computer science courses at our university use (manually graded) free-answer exams, we did not see any way to reliably grade text answers automatically. Although there are efforts to solve this problem, e.g. [8, 1], we nevertheless were reluctant to use these approaches in our courses. Therefore, and encouraged by work like [5], we decided to focus on multiple-choice exams. Discussions with the course instructors revealed that most could live with multiple-choice exams instead of free-answer exams, especially in the light of the following advantages:

- The exam papers can be graded automatically.
- Papers are graded consistently (manually graded papers may depend on the instructor).
- Apart from errors in the database or problematic phrasing, there is no room for debate about grades.
- The exams can easily be conducted on the computer and at remote locations.
- Once the question database has been set up satisfactorily, only a small amount of time needs to be invested into the exams.

The main problem we identified is with questions that should encourage the student to find a creative solution, like finding a design to a problem statement or do a complex calculation. However, these kind of tasks are anyway difficult to integrate into a

[1]This work is part of the SCDL "Seamless Campus: Distance Labs" project, which received support from the Austrian "FIT-IT Embedded Systems" initiative, funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) and managed by the Austrian Research Promotion Agency (FFG) under grant 808210. See http://www.ecs.tuwien.ac.at/Projects/SCDL/ for further information.

test of limited time. Therefore we decided to check this kind of knowledge in separate lab exercises.

From the students' side, the switch from the traditional free-answer exams to multiple-choice exams is often a difficult one, since the students are typically used to several amenities of free-answer exams:

- Usually no penalty for wrong answers and hence no penalty for guessing.
- Writing whatever springs to mind about the topic in question in the hope of getting at least partial credit.
- Being able to later argue about the interpretation of the answers, e.g. when asked to explain how something works.

With multiple-choice exams, at least the last two items are not possible anymore, and we strongly feel that the first item should be curtailed as well. Therefore, students used to the amenities of essay exams may have trouble adapting to the different rules of multiple-choice exams, and the transition has to be done with some care. In fact, we found that students did not really warm up to the multiple-choice questions, but were a lot more satisfied after we switched to true/false questions, even though these introduce new problems.

In the remainder of the text, we will first present and motivate our current exam style in Section 2. Section 3 is dedicated to a description of our exam management software, which plays a part in increasing student satisfaction. Section 4 discusses our experiences, both with our first exams in multiple-select style and with our current true/false questions, and the overall student reactions. Section 5 concludes the paper.

2 Exam Style

When we started out, we intended to use normal multiple-choice questions as described in [3] consisting of a so-called *stem* containing the problem statement followed by several possible answers, whereof only one answer is correct, the others are *distractors* that appear plausible to somebody without the knowledge to be tested.

However, on closer inspection we were hesitant to use this particular layout, because we did not like the fact that in such questions, there is exactly one correct answer. Since students know about this fact, they will pick the choice that sounds most plausible, leaving us with the task to come up with excellent distractors for every single question, which turned out to be quite difficult.

So instead of classical multiple-choice exams, we decided to employ multiple-select exams, where more than one choice may be a correct answer. Students got points for the question if they marked all correct answers, and there was no penalty for wrong answers. In multiple-select exams, students have to think more, it is not sufficient to just select the most plausible choice and be done with it. We actually used these exams on several occasions, but were still not satisfied. Our main point of criticism is that although the exam shows us whether students know which answers are correct, it does not show us whether they know which answers are wrong – after all, a student may leave a choice unchecked due to ignorance, or because he or she knows that the choice is wrong.

To address this problem, we changed the choices to true/false answers, that is, we provided two checkboxes next to each choice, one for yes/true and one for no/false. Students were required to tick off the appropriate checkbox for every single answer. This enabled us to better test the knowledge of students, and it also allowed us to have questions where none of the answers were correct, doing away with the problem that students know there must be one correct answer. So from our point of view, this exam style was already fairly acceptable. However, students criticized that they only got points for a question if they got all the answers right. No partial credit was awarded.

To address these remaining issues, we decided to drop the multiple-choice structure and switched to individual true/false questions. To help students concentrate during the exams, we still group questions into topics, but each question is scored independently. So our exams are now structured into n topics with k independent true/false questions each, which are individually scored. We found that this exam style is more satisfactory to the students, and it also allows us to cover a larger range of course material with less hassle. In the light of studies like [7], which indicate that true/false questions are comparable to open-ended questions with respect to the score ranking, we therefore settled on this exam style.

Yet, this new style does have its share of problems as well, most notably its bad guessing resistance.

According to the Merriam-Webster Dictionary, guessing means to choose a correct answer based on *conjecture, chance, or intuition* instead of knowledge. While it is impossible to draw a sharp distinction between conjecture/intuition and knowledge, a well designed exam should at least eliminate points from randomly chosen answers.

Guessing is a general problem in response-limited exams, and is either handled by deducting points for wrong answers or by accepting that students can and do guess. Although some go so far as to say that penalizing wrong answers is irrational (since there will always be some students who gain by guessing) and who advocate to encourage all students to guess without penalties in an attempt to make

things fair [2], we still argue that penalties, combined with student discussions about guessing and its ethical problems, are fairer and more instructive to students than allowing them to guess without penalties. Furthermore, we feel that computer engineering students, who may in their future career be responsible for safety- and life-critical systems, should not be encouraged to guess.

So we are not averse to penalizing wrong answers, and feel that it solely depends on the exam style whether guessing should be penalized or not. While there are styles where the expectation of points by guessing randomly is positive, but very low [5], our scoring scheme has been engineered to eliminate effects from random guesses on the average by awarding $+x$ points for a correct answer, 0 points for an answered item, and $-x$ points for a wrong answer, where x is a positive number.

Thus, a pure random guesser is expected to achieve a total of 0 points on average, while an intuitive guesser who will choose 75% of answers correctly gets on average 50% of the total achievable score.

3 Exam Environment

To conduct our exams, we use a tool that is currently under development at our university. Of course, there are several tools for computer-aided assessment available, but we wanted a tool that was flexible and could be extended to support additional exam styles, and which could be integrated into our course management software. Existing tools often were limited in their supported exam types. Therefore, we decided to develop our own software. Our exam management tool, called GTF (Generic Test Framework), is currently under development and only rudimentary functionality is available, but it already serves to help us identify our requirements for a good exam management tool.

GTF consists of several parts: One generates exams randomly from an ASCII database. Another is a client-server architecture which executes the exams and stores the students' answers and scores. The software is capable of displaying graphics along with the text, so questions can also refer to diagrams or formulas. Finally, an evaluation tool can generate a Latex file from the stored answers of a student. The Latex file contains the questions and the student's answers, as well as the score for each topic and for the complete exam, in an ASCII format, and allows to generate a nicely formatted postscript file. If the grading can be accepted as is, then we simply use a script to extract the students' scores from the files and to assign the grades. However, to be on the safe side, we developed additional tools to help us further process the exams:

First, we abandoned the idea of creating an individual exam for each student in favor of giving the same questions to all students that take the exam at the same time (8-16 students in our case). Since we now have several exam papers containing the same questions, we can extract an exam statistic: For each question, we count the number of correct and incorrect answers as well as the number of abstentions. Furthermore, the tool computes the percentage of correct answers out of all answers to a question, as well as its percentage of abstentions. With this information, it is very easy to spot problematic questions (they have a low percentage of correct answers and/or a high percentage of abstentions), which are then inspected more closely to determine whether there was a problem with the formulation of the text, a problem with conveying the material to the students, or even an error in the database. In the first two cases, we withdraw the question from the exam (this is also done automatically with the help of a script, which adapts the exam result accordingly) and the the evaluation result of the test will be only based on the remaining questions. In the last case, we correct the error in the database and re-evaluate the exams.

With the help of the statistic, which in our case is generated by a simple script and is available as ASCII text, we can spot problematic questions within minutes after starting the processing of the exams. This allows us to eliminate all problems in a couple of minutes, independent of the number of exam papers.

With these tools, we can process any number of exam papers within a few minutes, where most of the time is spent debating what to do with a problematic question. Automating these tasks and using statistics has also prevented us from having to handle individual papers and has thus ensured that all exam papers are graded automatically and in the same way, which was and is our foremost goal. It also makes sure that the grading is error-free with a very high probability, which is another of our goals. Although generating a statistic prior to grading means that students do not get immediate feedback on their exams, we do not perceive this as a drawback, since the grades are generally announced at the same day. And since we weed out problematic questions before students see their results, we in fact increase the students' trust into our questions.

4 Discussion

Although we may have been overenthusiastic when starting to use multiple-choice exams, it still came as a surprise to find that neither we nor our students liked this kind of exam, mainly due to the problem of generating good choices. Perhaps it is a

problem particular to our field, microcontroller and embedded systems, or perhaps it was due to our own inexperience with formulating such questions, but we simply found it way too time-consuming to create 20 or 30 good multiple-choice questions per exam group, especially since we generally need 6-12 exam groups per course. Mainly, the problem lay with the distractors, which we generally perceived as being either too obvious or too ambiguous, so we did not feel comfortable with this exam style.

Surprisingly, neither did the students. They too perceived many questions as ambiguous, and many of them already came into our courses with a bad attitude towards multiple-choice exams, which they had acquired from other exams. Apparently, good phrasing and unambiguity were difficult for the authors of these other exams, too. So we had students who already thought of our exams as "yet another exam where the student simply cannot win" before they had even seen the first question. It is pretty hard to go against that kind of preconceptions, and that we were not satisfied with our questions ourselves did not help.

Interestingly, students also complained about the scoring method of our multiple-select exams, even though we did not penalize wrong answers. The problem here lay with the all-or-nothing scoring of the questions. If there were four choices, three of them correct, and a student got two but not all three, then this student got no credit for the question.

This system introduced a great random factor for students who knew part of the questions. Depending on the distribution of wrong answers, the same number of correct answers could lead to a very different score. And indeed, in discussions with such students, we also got the impression that the exam results did not match the knowledge of the students, who were often quite prepared but happened not to know the particular detail the one choice had tested.

On the other hand, we have employed exams with true/false questions and individual scoring including penalties for wrong answers for over two years in three courses now and have gained some experience with this exam style during this time. Although especially in the beginning, ambiguous questions still occurred, generating good questions was easier from the start and has become even more so with practice, since one develops a feeling for problematic questions and learns how to formulate more carefully. This has smoothed out many of the initial difficulties both we and our students had with the exams.

Students also appear to have less problems with individual questions to which they simply have to answer yes or no. They feel more confident in their abilities to answer the question, and they seem to be less afraid of running into a trick question. Of course, these are just our subjective impressions gained during the exams, but student feedback during discussions of the exams has also improved after we switched to true/false questions.

However, we cannot deny that there are still some problems left, especially from the students' point of view. Some of these are home-made, others are inherent to the exam style.

First of all, students tend to believe that "multiple-choice is easy". Since the answers are given and their sole task is to find out whether the statements are true or false, they apparently believe that a cursory knowledge of the material should be enough, and are surprised when they belatedly find out that this assumption is wrong. We try to lessen the culture shock by telling them in advance that our exams are difficult, by telling them that we have seen a lot of their colleagues make these erroneous assumptions and fail, and by showing them sample questions. Once they know what to expect and prepare well, acceptance of the exams rises, and so does the students' performance on the exams.

Some students criticize multiple-choice exams as a "lottery". In part, this seems to stem from experience with multiple-choice exams where questions were perceived as ambiguous. Therefore, we stress that our statistic prevents such ambiguous questions from slipping by unnoticed, and we also explain that questions may appear ambiguous if the student is not sufficiently prepared. The second reason for the perception of the exam as a lottery is the possibility to gamble and win points. Here, we can only tell students that of course they can degrade the exam to a lottery, where they win or lose by chance, but they can also prepare for the exam, answer the questions they know, and remove the element of chance from the exam. It is their choice.

Finally, students do not like the penalty for wrong answers. Here, we point out that it would be unfair to prepared students if we did not penalize unprepared students for guessing. Furthermore, for courses that solely use true/false questions for the exam, we employ a fairly nice grading scheme which –as is usual at our university– fails people with a score below 50%, but already assigns an A for 80% and more. Since we can expect that students do not get more than one or two questions wrong by mistake, an assumption that is supported by our discussions with students, our grading scheme softens the impact of the penalties on students who only make honest mistakes.

In the beginning, we also had trouble settling on the right amount of time we should give students for our exams. We finally settled on the rule of thumb of 1 minute per question for an open-book exam and $20 - 30$ seconds per question for a closed-book exam, which seems to work quite well for our top-

ics (most students finish in time). Of course, more time is always appreciated, but since we are generally under tight time constraints due to the large number of students and the small number of workstations available for the exams, we had to settle on these values.

Now that we have gained some experience with our exams, we certainly do not consider switching back to free-answer or even oral exams, for several reasons. First, grading is fair, since it is independent of the identity of the student, the time of the day, and other such influences. This is of utmost importance to us. Also, grading is fast and scales well. We grade up to 150 exam papers per exam, and the grading itself takes two minutes. Even if the statistic reveals errors in the question database, these are corrected instantly and then all current exams are re-evaluated automatically. So the time complexity of correcting the exam is $O(n)$, where n is the number of exam questions. If you reuse exam questions after some time, the complexity even goes down to $O(1)$. If you compare that to a manually corrected essay exam, the time complexity is $O(n \cdot S)$, where S is the number of students. Taking into account that thinking up questions is more fun than correcting exams (at least to us), automatically graded exams soon outperform manually graded exams.

We also would not switch back to the classical multiple-choice exams. We did, however, convert the old multiple-select questions into a set of true/false choices if the topic lends itself to this type of question (e.g., *which pins of a microcontroller can be used by the timer module*, with a selection of pins to choose from). All choices are scored individually, though.

If asked what makes our exams with true/false questions better than multiple-choice exams, we would first mention the individual scoring. Of course, this brings the problem of penalties for errors, which we try to soften by telling students what to expect (and not expect) from the exams, by spreading information to increase the acceptance of the scoring scheme, and by discouraging students to guess in order to reduce the variance of their score, but still we feel that the advantages prevail.

In the context of guessing, there might be merit in the idea of confidence levels, see for example [4, 6], where students are allowed to indicate their confidence in answering a multiple-choice question (prior to seeing the answers in the case of Davies [4]). The idea is simply that the more confidence a student has in his or her ability to answer a question, the higher is the score for a correct answer and the penalty for a wrong answer. This could be used in our exams as well to allow students who are less sure to answer questions without giving them the odor of gamblers. The addition of a confidence level does not change the system for students who are sure, but allows honest students who are less sure to still answer with less penalty in case they are wrong. Similarly, it does not change the system for dishonest students. Although this may encourage some students to gamble for lower stakes, we also do not see too much of an ethical problem here since students admit that they are not too sure and accept a lesser reward in case they do happen to be right. Davies reported good student satisfaction with this system. However, note that such a system adds a second orthogonal examination to each question that is based on the ability of a student to judge his/her knowledge. A student that has sufficient knowledge on the examination subject may therefore fail because he or she was unable to correctly estimate the confidence levels.

A second point in our favor is that we really take the exams seriously, and we care a lot about fairness. This means we invest a lot of time into creating new questions, they are checked by colleagues and debated until we are satisfied with the phrasing. During the exams, we encourage students to ask if they have problems understanding questions, so we can clarify potential ambiguities and also identify ambiguous questions during the exam. After the exams, we extract our statistic which again points out remaining problematic questions. Of course, most of these efforts are only necessary for new questions, once a question has been used and passed all these checkpoints, it can be flagged as okay. But still, we run the statistic after every exam to identify problems the students may have with understanding the material.

As a final remark, we have to mention that students still do not like these type of limited-response exams and obviously would prefer free-answer or oral exams, yet on the whole grant us that our exams are fair and acceptable.

5 Conclusion

Although automatically graded exams, be they multiple-choice or true/false questions, are controversial, we believe that their usefulness outweighs their drawbacks. However, exam questions must be formulated with care. While the question database is being built up, an exam statistic is vital to finding problematic questions. Later, the statistic can be used to determine whether students have understood the material.

We have tried out both multiple-select exams with no penalties for wrong answers and individually scored true/false questions, and came to the conclusion that exams with true/false questions are better than multiple-select exams, from the point of view of both the instructors and the students. True/false questions are easier to generate, and are better understood by the students.

References

[1] L. F. Bachman, N. Carr, G. Kamei, M. Kim, M. J. Pan, C. Salvador, and Y. Sawaki. A reliable approach to automatic assessment of short answer free responses. In *19th International Conference on Computational Linguistics*, pages 1–4, 2002.

[2] M. Bar-Hillel, D. Budescu, and Y. Attali. Scoring and keying multiple choice tests: A case study in irrationality. *Mind and Society*, 1(1), 2005.

[3] A. S. Cohen and J. A. Wollack. *Handbook on Test Development: Helpful Tips for Creating Reliable and Valid Classroom Tests*. University of Wisconsin-Madison, Wisconsin, USA, 2004.

[4] P. Davies. There's no confidence in multiple choice testing. In *6th Annual Computer-Assisted Assessment Conference (CAA'02)*, 2002.

[5] D. W. Farthing, D. M. Jones, and D. McPhee. Permutational multiple-choice questions: An objective and efficient alternative to essay-type examination questions. In *6th Annual Conference on the Teaching of Computing and 3rd Annual Conference on Integrating Technology into Computer Science Education (ITiCSE'98)*, pages 81–85, 1998.

[6] A. Gardner-Medwin and M. Gahan. Formative and summative confidence-based assessment. In *7th Annual Computer-Assisted Assessment Conference (CAA'03)*, 2003.

[7] L. W. Lackey and J. W. Lackey. Influence of true/false tests and first language on engineering students' test scores. *Journal of Engineering Education*, Jan. 2002.

[8] P. Thomas. The evaluation of electronic marking of examinations. In *8th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'03)*, pages 50–54, 2003.