EFFICIENT AND RELIABLE ONLINE HOMOLOGATION RECOMMENDATION

Bas Luttik and Sander Stuijk

January 8, 2017

1. INTRODUCTION

The intake of many of the master programmes offered within the TU/e Graduate School is very diverse. Students may come from a major within the TU/e Bachelor College, or they may have a bachelor degree of some other Dutch university or from a university outside of the Netherlands.

The bachelor-level prerequisite knowledge and skills for the master programme are usually derived from the final qualifications of a pre-Bachelor College TU/e bachelor programme that gave direct admission to the master programme. Although some admitted students may still satisfy all these prerequisite knowledge and skills, many of them do not. Master programmes allow students to repair deficiencies to some extent in the form of *homologation modules*, bachelor-level courses that may be included in the master programme and count towards the required 120 EC.

Current practice is that virtually all students are recommended (or required), upon admission, to include some homologation modules in their master programme. The admission officer determines which homologation modules a student needs solely based on admission data (transcripts of records, course descriptions, etc.). This way of working is cumbersome and unreliable. It is cumbersome for the admission officer to read and understand all course descriptions, and hard to distil from this information to what extent the student indeed satisfies all prerequisite knowledge skills on the required level. Furthermore, the student often does not understand why he or she should include a certain homologation module, and feels it as a burden. It even happens that a master student still needs to pass a required homologation course as last course before being allowed to start a graduation project, when the master courses for which the knowledge and skills taught in the homologation course was supposedly prerequisite has already been passed.

The goal of the project *Efficient and Reliable Online Homologation Recommendation*, sponsored by the *Innovation Fund 2016* at TU/e, is to improve the method to assess to what extent an admitted master student satisfies the prerequisite knowledge and skills of a master programme, and, based on that assessment, provide a qualitative and reliable recommendation on how to repair deficiencies. To this end, we have developed, in the context of the Embedded Systems master programme, a *homologation recommendation tool*. The tool allows an admitted Embedded Systems student to self-evaluate to which extent he or she satisfies the prerequisite knowledge and skills of the programme through a series of digital tests, without any interference of lecturers. It then automatically provides, based on the outcomes of these digital tests, a recommendation which homologation modules to follow. A special concern of the tool has been to explain to students for which courses of the master program a certain piece of knowledge or skill is a prerequisite.

In this report, we shall briefly describe the steps taken to develop the tool, illustrate the working of the tool from a student perspective, and present some lessons learnt while developing the tool and some plans for future improvements.

2. DEVELOPING A HOMOLOGATION RECOMMENDATION TOOL

In this section, we shall present the steps we took when developing an online homologation recommendation tool for the Embedded Systems master programme.

The tool is provided as a service to the student. Its main purpose is

- to explain to the student for which parts of the Embedded Systems programme these bachelor-level knowledge and skills are relevant,
- to make the student aware of deficiencies in his or her bachelor-level background,
- to give recommendations on how to repair those deficiencies, and
- to motivate students to repair deficiencies in time.

The Embedded Systems programme offers several 2.5 EC *homologation modules*[1] to facilitate that students repair deficiencies. Students may include at four such homologation modules as electives in their study programme. Including homologation modules is never mandatory in the Embedded Systems programme. Recommendations provided by the tool will, hence, never be considered binding, and students do not get credits for performing well on the quizzes of the tool. The advantage of this stance is that fraud is not a concern, and this gives some flexibility when designing quizzes.

We distinguish four phases in the development of the tool.

PHASE 1: DEFINE PREREQUISITE KNOWLEDGE AND SKILLS

The first phase consists of defining and formulating which bachelor-level knowledge and skills should be considered a prerequisite for the master programme. The prerequisite knowledge and skills of the Embedded Systems programme are defined as bachelor-level knowledge and skills that are considered prerequisite for the mandatory parts of the programme. The Embedded Systems programme consists of a mandatory common core of five courses, and students must choose one of four streams which then adds three more mandatory courses to their study programme.

To make an inventory of the prerequisite knowledge and skills, we proceeded as follows:

1. We started with asking the lecturers of mandatory courses to describe the knowledge and skills they expected students to have at the start of their course. These prerequisites were often stated in general terms, and referring to the contents of bachelor courses.
2. We then looked up course descriptions of relevant bachelor courses and distilled concrete intended learning outcomes.
3. The concrete intended learning outcomes thus obtained were scrutinized in another round of interviewing lecturers of mandatory courses, leading to a finalised description of the

[1] 2.5 EC courses derived from 5 EC bachelor courses, with compressed and focused content, preferably offered in the form of online self-study courses.

prerequisite knowledge and skills of the Embedded Systems programme, differentiated per stream.

PHASE 2: COLLECT SUITABLE ASSESSMENT MATERIAL

To assess whether a student satisfies a certain piece of prerequisite knowledge or a certain skill, we needed to create a repository of suitable test questions. To this end, we have asked lecturers of relevant bachelor courses for old written examinations together with grading models. These old examinations were discussed with the lecturers of the mandatory courses of the Embedded Systems master programme, to determine a collection of questions deemed relevant and at the appropriate level.

PHASE 3: COMPILE DIGITAL QUIZZES

The homologation recommendation tool needs to evaluate student answers automatically. Most learning management systems offer a quiz facility allowing automatically graded quizzes. The supported question types do vary considerably. We decided to base our tool on the free and open-source learning management system Moodle (www.moodle.org), which also underlies the OnCourse learning management system in use at TU/e (oncourse.tue.nl). Moodle has a broad range of question types, facilitates typesetting mathematical formulas through a LaTeX plugin, and within both the departments of Mathematics and Computer Science and Electrical Engineering there is ample experience with using Moodle.

Since written examinations are generally hand-graded by a lecturer, only part of the collection of test questions could directly be used for compiling digital tests. Roughly, we could directly use multiple-choice questions and questions that asked students to compute a number. Some open questions could be translated to multiple-choice questions in collaboration with the lectures of the originating bachelor courses.[2] Questions that ask students to write a computer program solving a computational problem can also be graded automatically, using a suitable set of test cases.

The type of questions hardest to digitalise are questions to which the answer necessarily is free-form text (e.g., a mathematical proof), and which serves to assess whether the student has a certain skill. To automatically grade such questions, we have decided to use a two-step approach where the student first enters his answer to the question, and, after the answer has been submitted, ask the student to grade his answer using a grading model. We could safely resort to this approach because the purpose of the tool is self-assessment; students would not benefit in any way from not grading their answer honestly.

PHASE 4: IMPLEMENT CONTROL FLOW AND RECOMMENDATION

The previous phase resulted in a large question bank of digital quiz questions suitable to assess whether a student satisfies a certain piece of prerequisite knowledge or skill needed for the Embedded Systems programme. Using questions from this question bank, we compiled one quiz per bachelor course from which the questions originated, resulting in seven digital quizzes. If a student passes such a quiz, then this results in the feedback that the student satisfies the

[2] The main difficulty of constructing suitable multiple-choice questions is to invent reasonable distractors (incorrect answers). A method that proved fruitful was to harvest suitable distractors from piles of earlier student solutions to the old examination.

prerequisite knowledge and skills on the topic. If a student fails such a quiz, then this results in the feedback that the student does not satisfy the prerequisite knowledge and skills on the topic together with a recommendation how to repair the deficiency. Usually, the recommendation is to follow a homologation module. In some cases, when a suitable homologation module is not (yet) available, the recommendation is to read a certain book or study certain material.

In a Moodle course it is possible to offer activities conditionally: depending on the completion of a certain activity (a quiz), or depending on the result of an activity (e.g., some quiz is passed or failed), or even a Boolean combination of such conditions, it is possible to control whether some other activity is offered to the student. An activity can also be a simple label, just showing some text, with the option to allow students to manually mark such an activity as completed.

We used the aforementioned features of Moodle to implement a diversifying flow in the tool:

1. Which of the quizzes a student depends, first, on which of the four streams a student intends to follow. The first activity a student needs to do is select a stream.
2. It is not useful to let students do a quiz on a topic if the student already knows that he or she does not possess the knowledge and skills assessed in that quiz. Therefore, we added, per topic, so-called *preknowledge checks*. A preknowledge check first asks the student to indicate, per intended learning outcome on a certain topic, whether he or she already has obtained it. Only if the student indicates that he or she sufficiently satisfies the intended learning outcomes, then he or she will be asked to do the quiz on that topic (to confirm his or her own assessment).

Recommendation is also generated using the aforementioned features of Moodle. If the preknowledge check or the quiz on a certain topic is not passed, then a recommendation on how to repair the corresponding deficiencies is shown; otherwise it is not shown.

4. CONCLUSIONS AND LESSONS LEARNT

In the summer of 2016 we have invited 38 newly admitted Embedded Systems students to use our homologation recommendation tool. Of these 38 students, 21 students have used the tool, doing the preknowledge checks on the two topics most relevant for the Embedded Systems program (*Computer Architecture and C programming* and *Logic and Set Theory*).

On the topic of Computer Architecture and C programming, 13 of the 21 participants passed the preknowledge check; of these 13, only 4 completed the associated quiz, and only 1 of these 4 participants passed the quiz. For the quizzes on the topic of Logic and Set Theory, a similar the phenomenon was observed: 3 out of 16 participants passed the preknowledge check and 1 out of 3 participants passed the quiz.

On the one hand, one may conclude from the low passing percentage that it is difficult for students to give a good estimation whether they possess the required knowledge and skills on a topic. The quizzes are apparently necessary to provide a more reliably confirmation. On the other hand, students also indicated that the quizzes currently perhaps rely too much on specific terminology that is introduced in the corresponding bachelor courses. For a homologation recommendation tool, it is particularly important to either use generally accepted terminology, or include an explanation of terminology used in the formulation of the question.

We are under the impression that it still takes students too long to use the tool. Not many students complete all the offered quizzes. In future versions of the tool, we would therefore like to experiment more extensively with adaptive testing, and with more refined recommendations. To this end, we should establish, within the tool, a more refined correspondence between intended learning outcomes, quiz questions and recommendation.

For some of the quizzes, we should still confirm their validity with the lecturers of the mandatory courses relying on the knowledge and skills assessed.

Constructing digital tests is a lot of work. Unfortunately, a considerable part of this work is hard to delegate to student assistants, but student assistants can alleviate the task of entering digital quiz questions into the system.

With the current infrastructure, it is rather cumbersome to implement the necessary control flow (Phase 4 above). In 2017, we will run a Software Engineering project in which a group of Software Science bachelor students are supposed to improve the underlying infrastructure of the tool. This should make it easier for other master programmes to use the technology.