Automation of Homologation

Verhoeven, Chris J. M.; Montagne, Anton J. M.

DOI
10.23919/EAEEIE55804.2023.10181626

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 32nd Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)

Automation of Homologation

Chris.J.M.Verhoeven
Department of Micro Electronics
Delft University of Technology
Delft, The Netherlands
c.j.m.verhoeven@tudelft.nl

Anton.J.M.Montagne
Department of Micro Electronics
Delft University of Technology
Delft, The Netherlands
anton@montagne.nl

***Abstract*—We are developing content and software to automate the homologation for our Electronics courses to deal with the increasing diversity in knowledge and skills of the master students who have completed their bachelor's at other universities. The courses' content has been structured such that dependencies between topics become explicit. A parameterized database was created with questions characterized with, e.g., subject, Bloom level, and many other tags. It enables the generation of questions and personalized quizzes for individual students at any time. This allows teachers and students to gain insight into the student's level of knowledge and skills and provide adequate feedback at any time. The questions and complete quizzes can also be exported to learning platforms like Brightspace.**

Keywords—personalized homologation, question database, automated test generation, grade prediction, Brightspace interface.

I. Introduction

When students enter a master's program after completing a bachelor's abroad, the diversity in the knowledge and skills requires homologation. This would be most efficient if personalized, but that is logistically not feasible. Instead, master's courses tend to have a "homologation phase," with a few lectures covering the content the educator suspects is not at the required level for those students. In most cases, these lectures are boring for some and still not enough for others. But it is always at the expense of valuable lecture time that could have been spent on the content of the course itself.

So, we are developing a homologation platform to automate and personalize the homologation process for our Electronics courses such that it can be done before the start of the course itself. The content and prerequisites of our courses have been structured such that dependencies between topics become explicit. A database was created with questions and their characteristics like, e.g., topic, Bloom level, and type. The homologation platform can be used to (automatically) generate questions and individualized quizzes for teachers and students to gain insight into the student's level of knowledge and skills. A student can do this anytime to prepare for a lecture, a course, or a workshop and get advice on what to study when the knowledge and skills requirements are unmet. It is similar to exploring a world in a computer game (e.g., "World of Warcraft"), where areas can be explored only after all necessary challenges are completed and the level is sufficient. Students gain much freedom to follow their study path at their own pace without becoming lost or inefficient. This frees up time for the educators to spend on their course content. It also enables the educator to shift their time on, e.g., workshops and mentor classes to work on the higher Bloom levels with a much more knowledgeable and content-synchronized group of students.

II. Content Generation

A. Structured Electronics Design

Our first step to automating the homologation process was structuring the content of our courses in almost orthogonal modules of which a consistent set of learning goals could be defined, and the required entry-level was clear. The topic of our lectures is analog circuit design. Although many still consider this an "art," over the years, it has proven to be possible to clearly describe an orthogonal design methodology, with a minimal amount of iteration loops and well-defined design steps; a system engineering approach for analog electronics has resulted. This "Structured Electronics Design" methodology, in which well-defined orthogonal design steps lead to well-understood optimal designs concerning performance and cost factors, is highly appreciated by students and many expert designers in the industry. It also makes it suitable for automating the design process and automated personalized education. The required knowledge, skills, and their relation to the complete design procedure are explicit. This forms a solid basis for defining a complete set of educational modules and the accompanying question database.

The Structured Electronics Design methodology has been used in education at the TU-Delft since 1980, when Ernst Nordholt laid the fundamentals of the methodology [1], and is constantly extended and updated. The course book "Structured Electronics Design"[2], used at the TU-Delft currently, contains the latest developments. It is also used as the basis for our automated homologation project.

B. The automated homologation platform

Fig. 1 shows the complete setup of the proposed automated homologation platform.

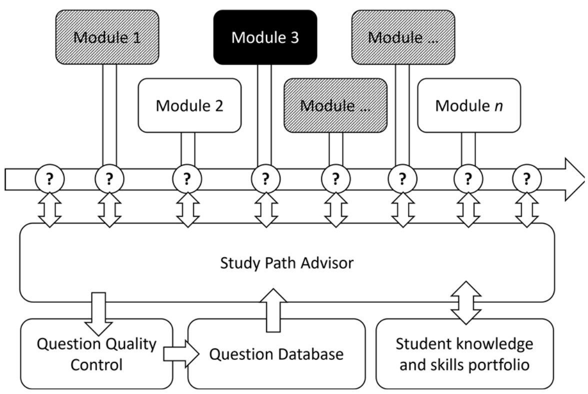

Fig.1. The homologation platform for an individual student. White modules are completed, grey modules are advised, black modules not advised (yet)

978-90-386-5690-8/23/$31.00 ©2023 TU/e

The course content is organized into modules, each covering a different topic. Inside are subtopics of diverse Bloom levels. Each module has its specific advised requirements for entry. Each student has a portfolio describing their present level of knowledge and skills. Based on this portfolio, the (Virtual) Study Path Advisor gives the student the insight to select the most efficient and feasible study path. The student can do formative and summative tests at will. The test's content depends on the student's portfolio and the study path the student has decided to follow. Each quiz updates the profile and provides feedback on which topic is best to improve.

Fig.1 shows a typical student portfolio where Module 3 is not recommended (yet), the gray modules are recommended and feasible but not completed with a sufficient level, and the white modules are completed with an adequate level. Of course, a student can enter a not recommended module anytime, maybe out of curiosity. It might provide the student with helpful feedback.

The questions are in the Question Database with characteristics describing their content and level.

There also is "Question Quality Control," which is used to tune, e.g., the Bloom-level characterization of a question to the appropriate level. This makes that, over time, the discrimination index of the questions in the database improves. With that, the test matrix of an exam also becomes a better predictor of the quality of this exam.

It is up to the educator what profiles are advised before joining a course.

III. TEST GENERATION WITH THE QUESTION DATABASE

A. Questions Characteristics

The questions are stored in the Question database with several characteristics:

- **Question Type:** True-false, Multiple Choice, and Multi-Select (for now).
- **Difficulty level:** expressed in Bloom Levels (Remember, Understand, Apply, Analyze, Evaluate, and Create.)
- **Courses:** indicates for which courses the question is applicable
- **Chapters:** indicates for which book chapters the question is applicable
- **Categories:** indicates to which collections of questions this question belongs. For example, a category may contain all questions about a specific topic in a module or lecture.
- **Pools:** indicates if and to what question pools the question belongs. A question pool consists of similar interchangeable questions and can be used to create randomized quizzes.
- **Tags:** contain specific, user-defined keywords. For example, all questions related to "noise" could be tagged with this keyword.

Two added characteristics not used for selection are the expected **Time** a student will take to answer the question and its value in **Points**.

The characteristics "**Difficulty level**" and "**Time**" are instrumental in creating a test in which the test matrix meets the standards concerning time pressure and discrimination index.

B. Impact of formative and summative tests

During the Corona pandemic, we were forced to let students take exams from home. Proctoring was expensive, still not fail-safe, and also put much stress on the students. It was one of our main drivers in boosting the creation and extension of the question database. Creating an individual exam for each student using question pools instead of single questions relaxed the dependency on proctoring. It also made it very easy to generate sample exams since they are just alternative instances of the same test matrix. Brightspace was chosen as the exam platform since the students were already familiar with quizzes. The grades are automatically added to the grade book in which other course test results were already included. A drawback at the time was that we had to reverse engineer the interface for importing complex questions with, e.g., Latex equations in Brightspace since this is done via an archaic CSV file. The import via this file was needed because we could easily create hundreds of instances of questions and create huge question pools. Although we found that a CSV file should contain no more than approximately 500 questions, we have not found a limit to the number of questions in a question pool in Brightspace.

We have too few exam results yet to draw a well-substantiated conclusion on the effect of using the question characteristics to design an exam with an acceptable test matrix, but the first results are promising. We randomly selected students that passed the exams for an oral test to verify their results. We did not find any indication that the relaxed proctoring, in combination with the individualized tests, led to exam fraud.

In fig.2, the results of two different exams are shown. The number of students was about the same. The level of the pre-corona exam of 2018 was "estimated" by the educator. The exam of 2021 was the first *online* exam where a test

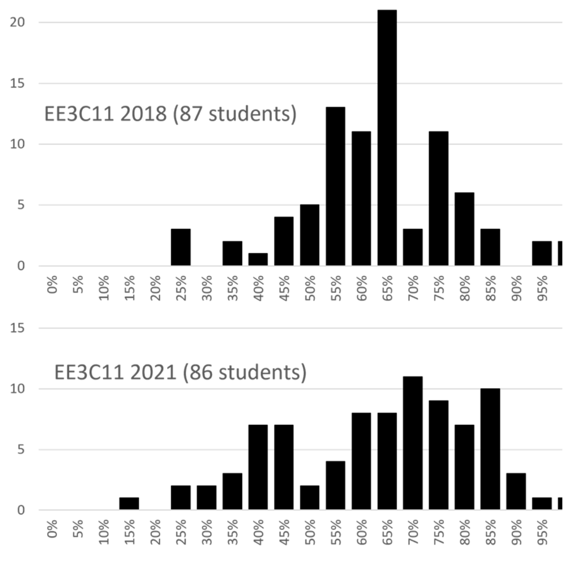

Fig.2 The results of two exams. For the 2021 exam, a test matrix was used to select questions from the Question Database.

matrix was used to select questions from the Question Database.
There is not enough data for any conclusion, but the 2021 exam seems to separate more clearly the students who pass from those who fail.

C. Tests with allowed student interaction

A Question Database has also been used for a summative bonus test for another course in 2022 and 2023. Each lecture of this course starts with a demonstration of an oscillator. Oscillators are electronic or mechanical systems that produce a periodic signal. An example of a mechanic system would be a metronome, and an example of an electronic system would be a tone generator. To give the students insight into the fundamental behavior of oscillators, so they can make a substantiated decision on which one to use in an application later in their career, the students are challenged to classify them. Different meaningful classes of these oscillators can be defined. Essential aspects for determination are the mechanisms that control the frequency and the amplitude, the stability of this control, and the question of whether the oscillator can become chaotic. The students have a workbook with images and diagrams of each demonstration in which they can take notes during the demonstration, and on the web, there are accompanying videos. Traditionally, at the end of the quarter, there was a session where the students could work together in a lab where all demonstrations were available for experimenting. The debates among the students about the qualification were instrumental in gaining insight. After this, the workbooks were individually evaluated. The corona pandemic and the rising number of students forced a format change. It was decided to complete the exercise with a summative test in a computer room, where each student had to complete an individual randomized test generated using the Question Database.

In 2022, the nature of the computer room was such that proctoring was virtually impossible. It was decided to allow the students to talk with each other and have debates about the

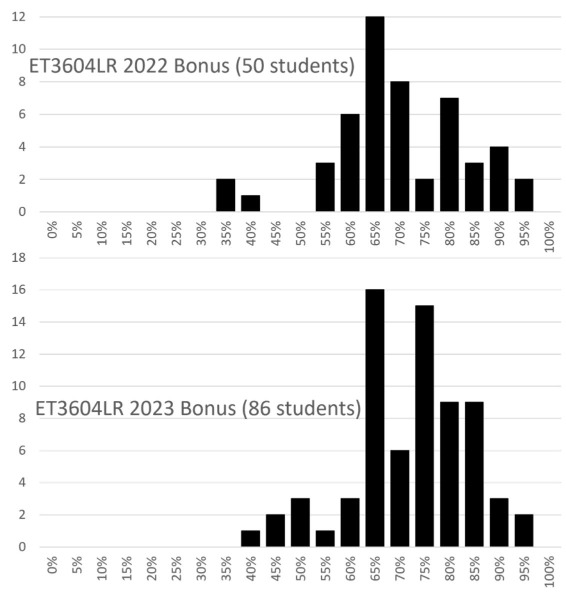

Fig.3 Results of two formative bonus tests where students were allowed to talk with each other. In 2022, the students did not know talking was allowed; in 2023, they did.

classification that used to happen during the lab session. The procedure was announced just before the test, and 50 out of a group of 131 students who did the final exam decided to do the bonus test. The students had to get used to the fact that they were allowed to talk with each other during an exam, but after some time, intense debates started, mainly amongst the students working on the topic during the lectures. The test results are shown in fig.3.

Again, there are not enough results to draw any conclusions, but the distribution of the grades does not suggest that students followed a “knowledgeable leader.” Afterward, the students were excited about the format, and many considered the intense debates one of the best learning experiences.

In 2023 the format was repeated. Now the students knew from the start that the test would be a test where they would be allowed to talk with each other. 86 out of 148 students decided to take part in the test. The results are also shown in fig.3. Again; the students were excited about the learning experience because of the format. One student mentioned that ‘’he had never thought as intensely about a subject as in the debates during the exam. It had been the occasion when finally all became clear.’

The results have strengthened our idea that with an extensive Question Database with well-characterized questions and an adequately defined test matrix, it is possible to (automatically) generate individualized formative and summative tests that give valid results under relaxed proctoring conditions and that can even allow debates amongst the students during the exam. Then these exams become more than just a “stressful level thermometer” for the students, but also the place where they are challenged at the higher Bloom levels. It also makes us believe that automatically generated formative and even summative tests with relaxed proctoring are meaningful when the homologation is automated. The relaxed proctoring for the tests of this course meant that the students had to be in the exam location during the test, and only students registered for the lecture series were allowed in. No contact with the “outside world” (e.g., via social media) was allowed, and tablets, phones, and laptops had to be in flight mode.

D. Question generation on course websites

The availability of an extensive Question Database with well-characterized questions makes it little effort to accompany any study material on a course’s web page with matching questions. So on the course websites of Structured Electronics Design, each page offers the content for a specific lecture, like slides, book references, links to videos, etc. also contains “test buttons.” When a student presses a button, a relevant question is retrieved from the Question Database. When the answer is evaluated, not only is feedback given on the correctness. In many cases, when the answer is incorrect, the kind of error that seems to be made and the material to study is given as feedback. Adding this feedback to a question in the database is much work, but since questions can be re-used, it is worth the effort. A new question is generated every time the student presses the button or refreshes the webpage. Since the formative and summative test questions come from the same database, “pressing the buttons” is an excellent way to practice for an exam. Also, the feedback loop is shorter and far more efficient than when making old exams just before the final exam is due. This even counts for the educator who might notice during the lectures already, instead of after correcting the exams, that students tend to give a specific incorrect answer systematically for some topics. During the lecture period, there is still time to act.

IV. The Software

A. Quiz Server

Currently, the Question Database is just a directory in which each question is stored as a text file. With a “Quiz Server,” the database can be accessed. The Quiz Server is written in Python and interfaces with the user as a local webserver. For building the Question Database, it can create, delete, view, and edit questions. It can also be used to create categories and pools of questions and define tags. It can be used to ask a question randomly selected from a Category. This feature is used to create the test buttons on the web pages of a course. It is possible to use complex Latex expressions in the question statements and the feedback.

For creating a formative or summative test, the Quiz Server can export a Category as a CSV file that can be imported into Brightspace. (Latex expressions are included in this CSV file without any problems!) A Category is composed in the Quiz Server by selecting questions based on their characteristics or questions that are part of a previously created question Pool. An export mechanism for Brightspace was built because it is the learning platform used at the TU-Delft. However, writing an export module for other learning platforms is relatively easy.

Creating questions that use equations to define the question statement and correct and incorrect answers is also possible. They can be directly exported via the CSV file or first expanded, using a user-defined range of numerical parameters, to a huge set of different numerical instances of the same question. During the expansion, care is taken that questions with numerical answers that are so close that they might confuse a student are not included in the output file.

B. Latex, paragraphs, and commas for Brightspace import.

Brightspace does not have a convenient tool to create large quantities of questions in its Question Library, nor is it easy to re-use questions from another course, share questions with colleagues, or share a consistent set of questions on the web. And using the CSV file for importing complex questions, or even questions that contain a comma or a paragraph, let alone Latex expressions, seemed impossible for us. But fortunately, we found during the Corona pandemic that including all this in the CSV file is relatively easy. Brightspace uses MathML to format equations (which could be just a comma), and MathML statements can be included in the different fields of the CSV file.

To import a question text in a question in Brightspace via the CSV file that has only “a_b” in the question statement, in the Quiz Server, the question field would contain the Latex statement a_b. The expansion in the CSV file then looks as shown in fig.4. It shows the three entries for the definition of

QuestionText	<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><msub><mi>a</mi><mi>b</mi></msub></mrow></math> Parameters: <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><msub><mi>R</mi><mi>s</mi></msub></mrow></math> = A*10^(B+1) <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><mrow><mo stretchy="true" fence="true" form="prefix">[</mo><mi>Ω</mi><mo stretchy="true" fence="true" form="postfix">]</mo></mrow></mrow></math> : Source resistance <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><msub><mi>f</mi><mi>l</mi></msub></mrow></math> = <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><mrow><mo stretchy="true" fence="true" form="prefix">[</mo><mi>H</mi><mi>z</mi><mo stretchy="true" fence="true" form="postfix">]</mo></mrow></mrow></math> : Lower limit of the frequency range of interest <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><msub><mi>f</mi><mi>h</mi></msub></mrow></math> = <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><mrow><mo stretchy="true" fence="true" form="prefix">[</mo><mi>H</mi><mi>z</mi><mo stretchy="true" fence="true" form="postfix">]</mo></mrow></mrow></math> : Upper limit of the frequency range of interest <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><msub><mi>R</mi><mi>i</mi></msub></mrow></math> = R_s*A <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><mrow><mo stretchy="true" fence="true" form="prefix">[</mo><mi>Ω</mi><mo stretchy="true" fence="true" form="postfix">]</mo></mrow></mrow></math> : Input resistance of the amplifier	HTML

Fig.4 Latex statement a_b *translated into a* MathML *statement in the CSV file for Brightspace import*

the question text. The first contains the keyword “QuestionText,” and the third contains the keyword “HTML.” The second one includes the MathML translation of the original Latex statement. It is beyond the scope of this paper to elaborate on this further, but we hope this gives a clue on how to import complex questions into Brightspace via the CSV file. The Quiz Server does this translation automatically, so the fact that the CSV file becomes illegible is not seen as a problem by us (nor by Brightspace).

C. Future developments: SLiCAP

Over the years, a Symbolic Linear Circuit Analysis tool (SLiCAP) [3] has been developed to support electronic circuits' symbolic and numerical analysis and synthesis. Both students and professional designers use it to formulate and evaluate design equations for all performance aspects during the different design steps defined in the Structured Electronic Design methodology. The discussion of SLiCAP itself is beyond the scope of this paper, but it is the intention to link it to the Quiz Server. This will make it possible to evaluate symbolic answers given by students, compare them with the correct answer, assess differences, and detect “common mistakes.” This can lead to appropriate feedback, including references to topics to study, and also give clues to the educator for improving the course material.

Students have been using SLiCAP for several years in our electronics courses, especially the more talented students who aspire to become expert designers who use SLiCAP to its full potential.

V. Conclusions

In this paper, we have discussed our progress toward automated homologation. The content, lecture material, and self-test environment for the Systematic Electronics Design Methodology courses were optimized for automated homologation and self-study, leaving more time for the educators to interact with the students at the higher Bloom levels. A Question Database was generated in which the questions are accompanied by characteristics that enable the creation of tests with a meaningful test matrix. Evaluating the test results is used to improve the discrimination index of the questions in the database.

Different tools have been developed to create and use the database in practice. Practical results with formative and summative tests give promising results concerning test quality.

The possibility of creating individualized tests seems to allow relaxed proctoring. The first experiments with allowing students to talk with each other during a test seem to enhance the student's learning experience, especially concerning the higher Bloom levels. So, automated homologation with relaxed proctoring and other automated individual test generation while maintaining quality is possible. It may even add a valuable way to improve a student's knowledge and skills at higher Bloom levels. It also saves time for learning activities like mentor classes, workshops, and design exercises.

Another conclusion is that such automated homologation helps scale up the course, increasing the number of students without increasing the effort in synchronizing their knowledge and content level. Therefore, this additionally encourages diversity in the master's study while eliminating concerns about adequate prerequisites.

ACKNOWLEDGMENT

This research has been and still is made possible by project funding from the TU-Delft and a Comenius project in cooperation with the TU/e.

REFERENCES

[1] E.H.Nordholt, "The design of high-performance negative-feedback amplifiers," doctoral thesis, 1980, downloadable from the TU-Delft repository.http://resolver.tudelft.nl/uuid:54fb267c-7101-4cb7-a4d9-a9ad2da14650

[2] A.J.M. Montagne, "Structured Electronics Design," Free download from https://analog-electronics.tudelft.nl.

[3] SliCAP is free of use and licensed under a `Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. Download from: https://analog-electronics.tudelft.nl/SLiCAP.html