Imperial College London

BEng Individual Project - Final Report

jSCAPE - Java Self-assessment Center of Adaptive Programming Exercises

Author:
Alexis Chantreau

Supervisor:
Dr. Tristan Allwood

June 4, 2014

Contents

1 Introduction **2**
- 1.1 Motivation . . . 2
- 1.2 Objectives . . . 2
- 1.3 Contributions . . . 4
- 1.4 Report Structure . . . 4

2 Background **5**
- 2.1 Computer Based Tests . . . 5
- 2.2 Computerized Adaptive Testing . . . 5
- 2.3 Probabilistic Test Theory . . . 8
  - 2.3.1 Maximum Likelihood . . . 8
  - 2.3.2 Bayes probability theory . . . 8
  - 2.3.3 Item Response Theory . . . 8

3 Related Work **10**
- 3.1 Environment for Learning to Program . . . 10
- 3.2 CourseMarker . . . 10
- 3.3 Automatic Exercise Generator with Tagged Documents based on the Intelligence of Students . . . 10
- 3.4 Adaptive Self-Assessment Master . . . 11
- 3.5 System of Intelligent Evaluation Using Tests for Tele-education 11

4 Evaluation **12**
- 4.1 Qualitative . . . 12
- 4.2 Quantitative . . . 12

5 Conclusion **13**
- 5.1 Future Work . . . 13

A User Manual **15**

List of Figures

2.1 Flowchart of an adaptive test. Adapted from [1]. 7

Chapter 1

Introduction

Mention three tier architecture

1.1 Motivation

- The rise in MOOCs such as Coursera, Udacity shows that there is a real interest in learning programming.
- Programming is considered "difficult" to learn and it can only be learnt effectively through lots and lots of practice.
- This is even more apparent when students are introduced to other programming languages with different paradigms such as Haskell and Prolog, or more low level programming languages such as C.
- Hence, the need to provide a platform for students to practice their programming skills and understanding of programming concepts.
- Having a lecturer come up with all the exercises by himself can be both time consuming and ineffective: some exercises may not be challenging enough for certain top students, or on the contrary too difficult for struggling students, which can be discouraging.
- Lecturers can only rely on a few homework assignments to get an idea of how students are doing. Having a way for lecturers to gather large amounts of data about students' performances would be beneficial. Allows for supplementary material, exercises, etc...

1.2 Objectives

Having identified the problems associated with teaching and learning programming, we were lead to formulating objectives in order to make the

project successful and useful to the parties involved.

The main objective of the project was to produce a web-based teaching infrastructure to complement the introductory first year programming classes. Four key features were identified:

- **Programming questions/exercises -** The web platform should allow students to practice their programming skills and understanding of programming concepts. There should be no limit to the number of questions a student can answer, so that if a student desires more practice, then he should be able to do that. Additionally, it should be possible for a specific set of people, such as lecturers and tutors, to add questions/exercises to the system.
- **Progress tracking -** Designated people, such as lecturers and tutors, should have access to detailed statistics about the students performances. This will provide them with useful information about difficulties particular students, or the entire class, may be facing. In addition, the system should give feedback to the students, in the form of simple statistics, allowing them to identify their weak areas and thus improve on them.
- **Adaptive difficulty -** The questions or exercises presented to the students should be suited to their ability. Not only will this stimulate the learning process, but it will also give a better indication of a student's understanding of the programming concepts being tested.
- **Automated generation -** There should be a mechanism to allow for some degree of automated or semi-automated generation of exercises. This will provide a large supply of "fresh" questions, so that students don't end up answering the same questions and memorizing the answers to them.

While investigating existing solutions (Section 2.4 - Related Work) we found out that some of these features were less common than others. The availability of programming exercises and progress tracking are very essential in such systems, therefore all of the tools implemented these features. On the other hand, relatively few tools integrated some form of adapting the questions to the students' ability. Finally, almost none of the tools featured automated generation of questions, opting instead to allow exercises to be added manually to the system.

1.3 Contributions

1.4 Report Structure

Chapter 2

Background

2.1 Computer Based Tests

CBT abbreviation - offers advantages such as being able to display higher quality visuals such as pictures, videos, graphs, etc... - low paperwork, everything is stored on the computer - automatic grading, less work for teachers - generation of statistics is made easier by the fact that the data can be processed by the computer - nowadays a lot of learning is done on computer systems, and children are used to dealing with computers so assessment through this medium is advantageous.

CBTs are typically "fixed-item" tests where all the students answer the same set of questions, usually provided by the person responsible for the assessment. This isn't ideal since students can be presented with questions that are too easy or too difficult for them to answer. Consequently, the results of the test won't give a very accurate representation of a student's ability, and for this reason, these types of tests aren't extremely useful. This problem lead to research and the development of computerized adaptive testing (CAT).

2.2 Computerized Adaptive Testing

Computerized adaptive testing (CAT), also called *tailored testing*, is a form of computer-based testing which administers questions (referred to as *items* in the psychometrics literature) of the appropriate difficulty by adapting to the examinee's ability. For example, if an examinee answers an item correctly, then the next item presented will higher on the difficulty scale. On the other hand, if they answer incorrectly, they will be presented with an item lower on the difficulty scale.

From an architectural perspective, a computerized adaptive test (CAT) consists of five components [3]:

1. Calibrated item pool

An item pool is needed to store all the items available for inclusion in a test. This item pool must be calibrated with a psychometric model. During this phase, the item parameters are estimated according to the chosen model and scaled to fit with already existing items. Usually, the psychometric model employed in these systems is called Item Response Theory(IRT) (Section 2.3.3). Calibration is a complex process, and to be done accurately it requires a considerable amount of data. Typically, it is performed by psychometricians, aided by expensive and sophisticated calibration software.

2. Starting point

Initially, when zero items have been administered, no information is known about the examinees and so the CAT is unable to estimate their ability. As a result, the item selection algorithm will fail to choose the next item to be administered. If there is previous information available, for example an examinee's ability estimate in a closely related subject, then this can be input into the system to form the starting point configuration. Often this data isn't available or too costly to collect, so the CAT's initial ability estimate for the examinee corresponds to the mean on the ability scale - hence the first item presented will be of average difficulty.

3. Item selection algorithm

The item selection algorithm chooses the next item to present to the examinee based on the ability estimate of the examinee up to that point.
There are several methods available to serve as the item selection algorithm, these will be examined when we consider IRT(section...)

4. Scoring algorithm

After an examinee has answered an item, the CAT will update its estim

5. Termination criterion

The CAT shouldn't be terminated too early, so as to allow enough time to estimate the examinee's ability with reasonable accuracy.

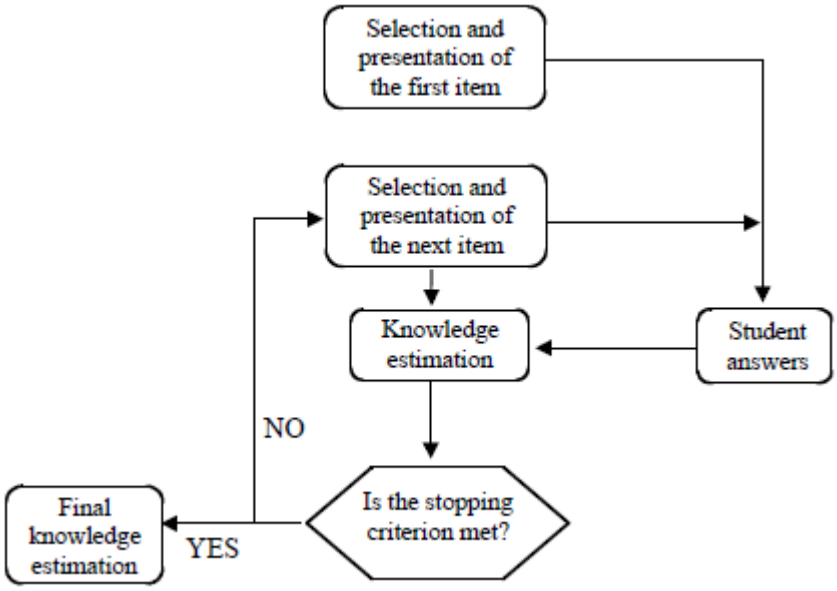

Figure 2.1: Flowchart of an adaptive test. Adapted from [1].

The flowchart in figure 2.1 corresponds to components 2-5, and illustrates the basics of the algorithm implemented in CAT. [2] gives a more detailed description of the procedure:

1. The pool of items that haven't been administered yet is searched to determine the best item to present to the examinee, according to the current estimation of his ability.

2. The chosen item is presented to the examinee, who then answers it correctly or incorrectly.

3. The ability estimate is updated, based upon this new piece of information and the previous ability estimate.

4. Steps 1–3 are repeated until a termination criterion is met.

5. The algorithm returns a final ability estimate for the examinee's performance along with a confidence level: a percentage value indicating how accurate the estimate is.

CATs offer several advantages when compared to CBTs. As a result CATs have been used in many areas, ranging from education, hospital questionnaires, etc..
Advantages include - precise scores for most test takers - requires less items to be administered before arriving at an equally accurate score, in comparison to static multiple choice tests - like any computer based test adaptive

tests may show results immediately after testing, no grading having to be done by teachers, so saves time and money.
CATs present many attractive features but they do present a few disadvantages. - the need for a large bank of calibrated items to cater to all ability levels. - Items are administered one by one, so once an answer to the item has been given there is no turning back. No skipping questions either.
A brief overview of CATs was given in this section. All of these concepts will make more sense after reading the section on IRT and the implementation of adaptive testing in our system, jSCAPE.

2.3 Probabilistic Test Theory

2.3.1 Maximum Likelihood

IRT is based on the idea that the probability of a correct/keyed response to an item is a mathematical function of person and item parameters.

- Calculates the probability of a particular student answering a specific item correctly.
- Different IRT models: One-Parameter Logistic (1-PL), Two-Parameter Logistic (2-PL), Three-Parameter Logistic (3-PL). Refers to the number of parameters used in the model. Parameters are:
  - item difficulty parameter (b)
  - item discrimination parameter (a)
  - chance/guessing parameter (c)
- Item Characteristic Curve, i.e. probability distribution

2.3.2 Bayes probability theory

2.3.3 Item Response Theory

For these reasons, Item response theory (IRT) has seen frequent usage when it comes to CATs....We present the different models developed to predict the probability of a correct or incorrect response to a particular item.

The one-parameter logistic model

dfsdfsdf

The two-parameter logistic model

dfsdfsdf

The three-parameter logistic model

dfsdfsdf

Chapter 3

Related Work

Web-based/Computer based education and adaptive web-based assessment systems are a "hot" research area, and as a result, numerous tools, environments and infrastructures have emerged. There are common features to all, however some distinguish themselves by having not so common features. Automated exercise generation in these tools is usually non-existent or very limited. Moreover, the tools are more focused on assessing students rather than self-assessment, i.e. students take tests which count towards their final grade on these systems.

3.1 Environment for Learning to Program

Environment for Learning to Program (ELP) is an interactive web based environment for teaching programming to first year Information Technology students at Queensland University of Technology (QUT).

3.2 CourseMarker

CourseMarker is a re-design of Ceilidh, a computer based assessment system used at the University of Nottingham for 13 years. Ceilidh was quite a complete system, providing coursework, the management of modules and the presentation of module content.

3.3 Automatic Exercise Generator with Tagged Documents based on the Intelligence of Students

The Automatic Exercise Generator with Tagged Documents based on the Intelligence of Students (AEGIS)

3.4 Adaptive Self-Assessment Master

Adaptive Self-Assessment Master (ASAM) is an extension to CourseMarker, which improves upon it by administering questions which are suited to the student's ability.

3.5 System of Intelligent Evaluation Using Tests for Tele-education

The System of Intelligent Evaluation Using Tests for Tele-education (SIETTE) is a web based environment for generating and constructing adaptive tests.

Chapter 4

Evaluation

Mention how our developed system performs against the advantages and disadvantages of CAT.
The evaluation stage will address mostly these two aspects:

- The system has correctly modelled the ability of students
- The system is useful in helping students to learn programming and helping lecturers with getting feedback on their teaching, in the form of statistics.

4.1 Qualitative

- Surveys to get feedback from students on interface, usability, etc...
-

4.2 Quantitative

- Statistical analysis to evaluate item calibration and modelling of student's abilities
-

Chapter 5

Conclusion

5.1 Future Work

- Very flexible system so other programming languages could be offered, i.e. cSCAPE, for C and hSCAPE for Haskell.
- Working more extensively on the adaptive component of the system, i.e. improving the algorithm which selects questions for students based on their estimated ability.
- Extend system to allow admins to provide their own question templates, maybe come up with a template grammar which then allows questions to be automatically generated. Or at least allow pluggable function references which will be called to generate the exercise component.
- Add support for more question types. JavaFX is very good in that sense since it can play audio clips, video clips, show animations, the webview component has endless possibilities thanks to the inclusion of Javascript.
- Future Work as a research project vs future work as a commercial product
- JavaFX applications are based on the model-view-controller pattern, so a nice split can be done in the code, however I only learnt about this two weeks into the project, so all the of the GUI components are created in the code as opposed to in the FXML file.

Bibliography

[1] Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-de-la-Cruz, J. L., & Rios, A. (2004). SIETTE: A Web-Based Tool for Adaptive Testing. *International Journal of Artificial Intelligence in Education, 14*, 29-61.

[2] Computerized adaptive testing. `http://en.wikipedia.org/wiki/Computerized_adaptive_testing`. Accessed: June 4, 2014.

[3] Thompson, Nathan A., & Weiss, David A. (2011). A Framework for the Development of Computerized Adaptive Tests. *Practical Assessment, Research & Evaluation*, 16(1).

Appendix A

User Manual