

jSCAPE – Java Self-assessment Center of Adaptive Programming Exercises

Alexis Chantreau

Introduction

- Programming can be a useful skill to have in today's society.
 - University CompSci courses and MOOCs enrolment → Definite interest in programming.
 - Programming can be difficult to learn. (different programming paradigms and specific skills needed)
 - Students are limited by the number of exercises.
 - Teachers receive a limited amount of feedback.
 - Creating exercises can be time consuming and inefficient.
-

Motivation

- More practice for students. (self-assessment)
 - Better feedback for teachers.
 - Easier way for teachers to supply exercises.
-

jSCAPE System

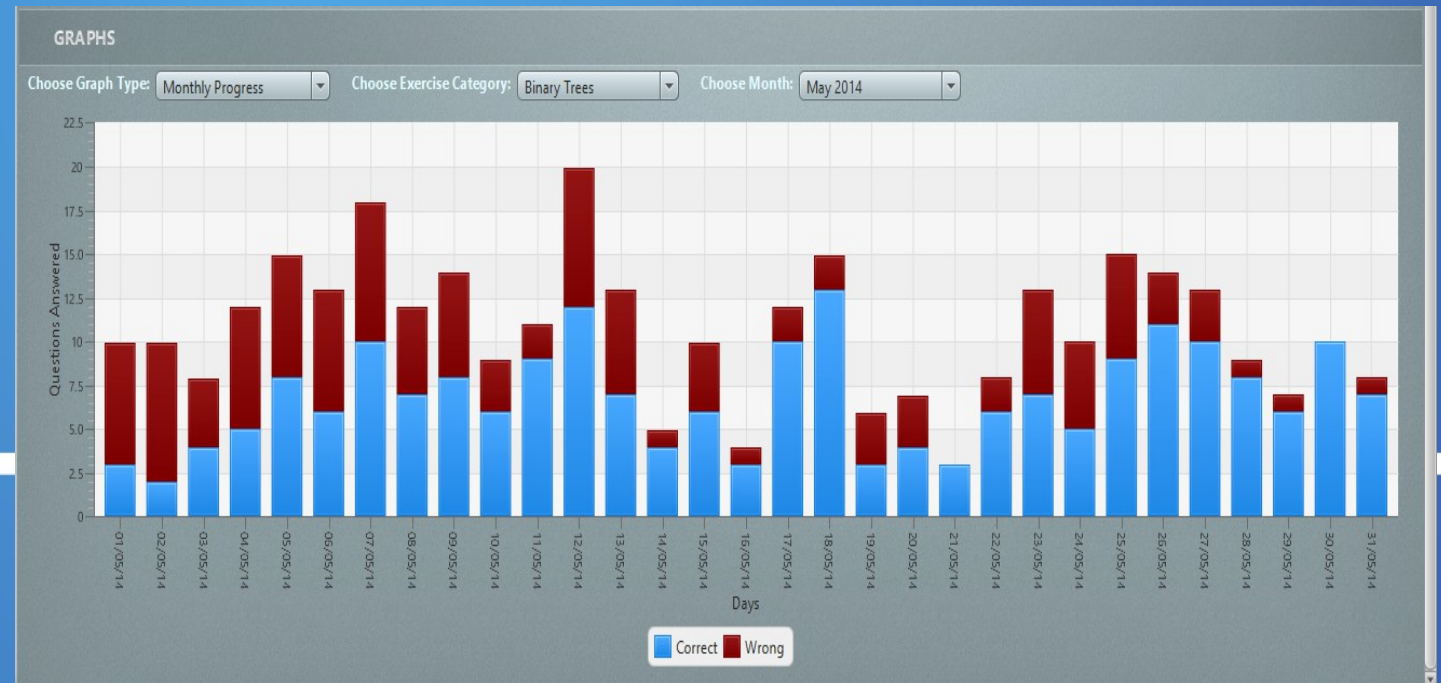
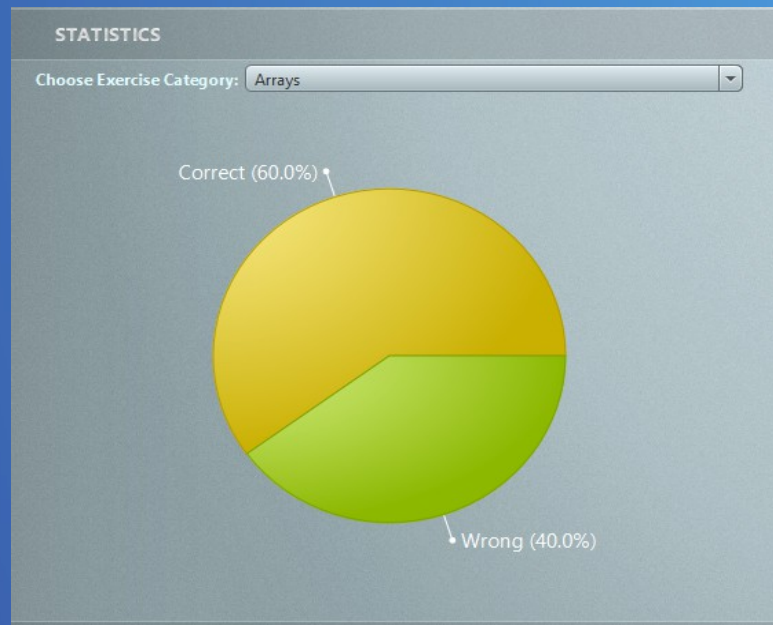
- Provide programming exercises for practice.
 - Display statistics on student progress/performance.
 - Adapt the difficulty of exercises to the student's ability.
 - Automatically generate exercises.
-

Exercises

- Exercises to test understanding of programming concepts (in Java).
 - Currently jSCAPE supports multiple choice questions only.
 - Format is exercise data on the left + question and choices on the right.
 - Exercises divided into different exercise categories.
 - Simple feedback after exercise, i.e. show correct/incorrect + solution.
 - Difficulty of exercises is adapted to the student's ability.
-

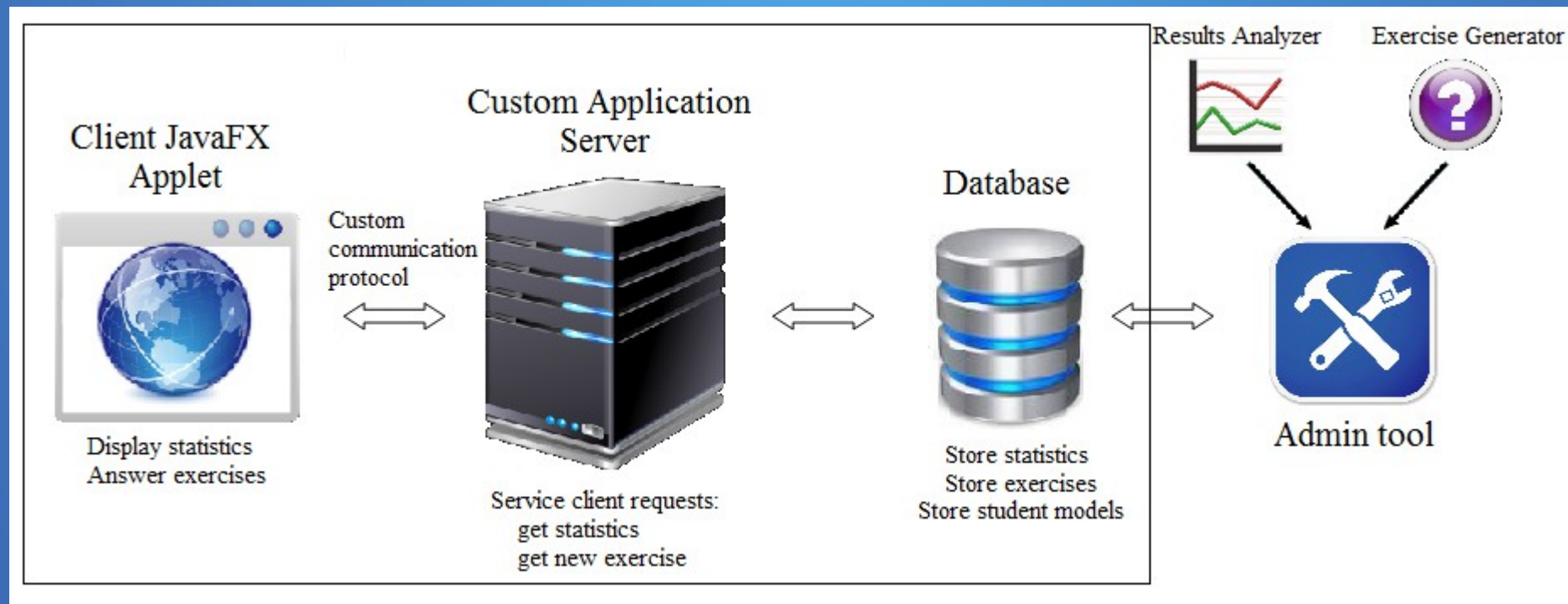
Statistical data

- jSCAPE collects various statistical data, e.g. correct/incorrect, progress over time, ...
- Students can view statistics on their own performance/progress.
- Teachers can view statistics for individual students or a class as a whole.



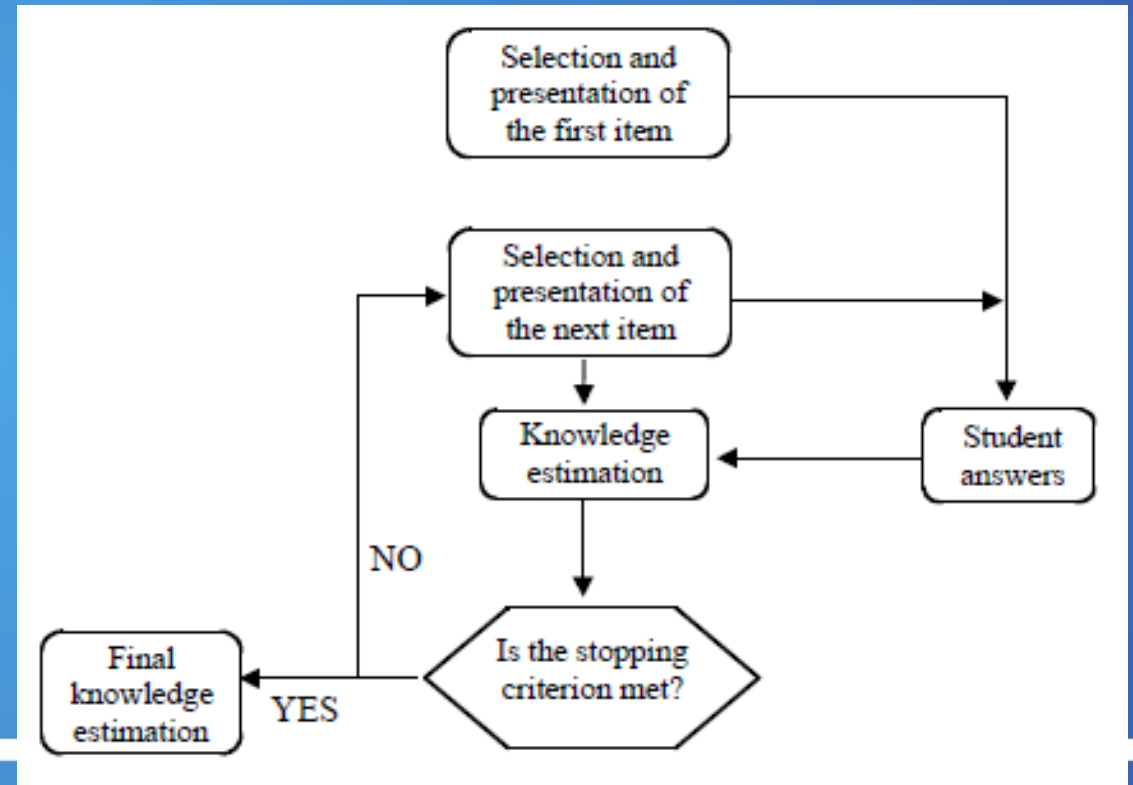
Implementation - jSCAPE

- Three tier architecture: web client, server and database.
- Web client is a JavaFX applet.
- Custom written Java server and communication protocol.



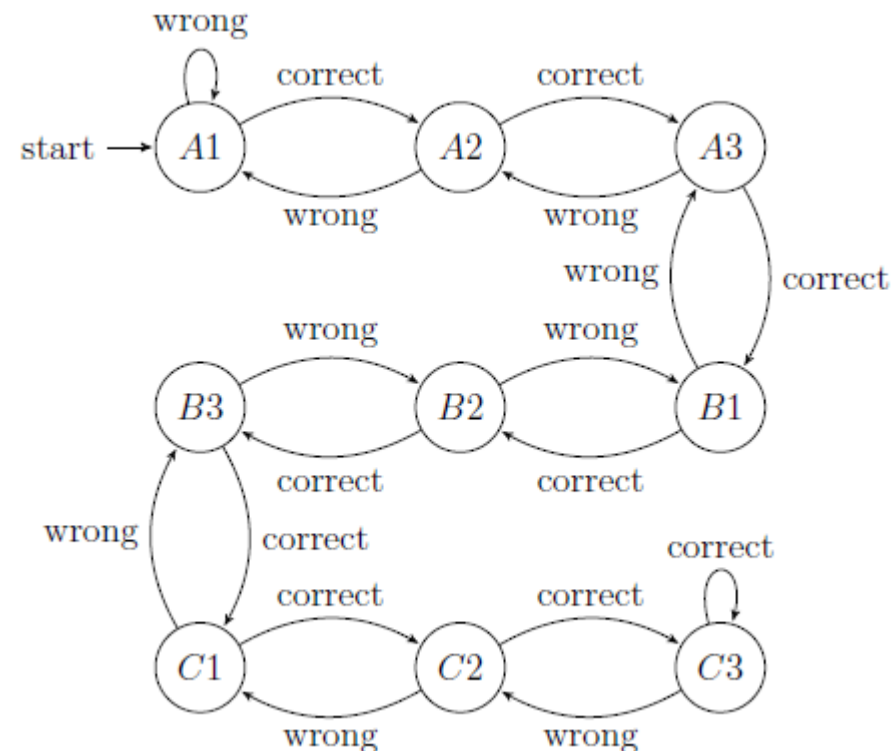
Adaptive Difficulty - CAT

- Based on computerized adaptive testing (CAT).
- Calibrated item pool.
- Starting point.
- Item selection algorithm.
- Scoring algorithm.
- Termination criterion.



Adaptive Difficulty – Simple Algorithm

- Simple algorithm inspired from Programming Adaptive Testing (PAT) software.
- Exercises split into 3 difficulty categories: A (easy), B (intermediate), C (difficult).



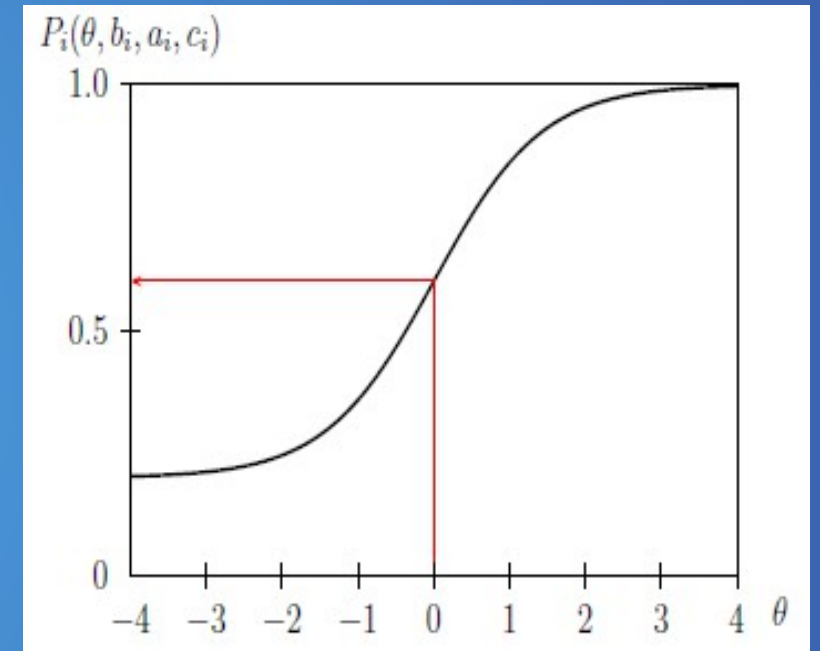
Adaptive Difficulty – More Complex Algorithm

- More sophisticated algorithm based on Item Response Theory (IRT).
 - IRT is a psychometric model, an improvement over Classical Test Theory.
 - IRT assumes that one can model the probability of a correct answer to an item, given item parameters and examinee ability level.
 - Several IRT models exist to address the different types of tests, e.g. multiple choice, agreement questionnaires (Likert scale), etc...
-

Adaptive Difficulty – Item Response Theory

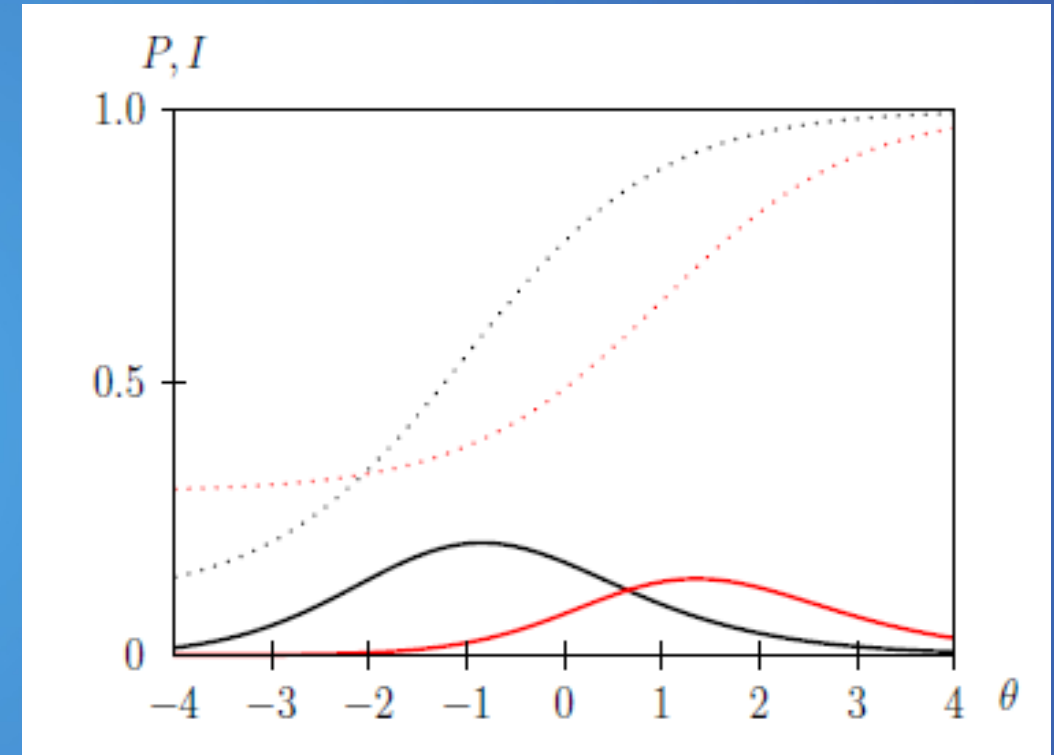
- Item response theory 3PL model used in jSCAPE.
- Theta (θ)= trait being measured, e.g. student ability/knowledge.
- a = discrimination; b = difficulty; c = pseudo-chance/guessing.
- $P_i(\theta)$ =probability of correct answer for item i at ability θ .
- In the graph: $a = 1.4$; $b = 0$; $c = 0.2$

$$P_i(\theta) = c_i + (1 - c_i) \frac{1}{1 + e^{-1.7a_i(\theta - b_i)}}$$



Adaptive Difficulty – Item Information

- Precision in the ability estimate that the item provides, at all ability levels.
- Indication of quality of item in terms of how well it discriminates between respondents.
- Concept used in item selection algorithm
e.g. maximum information method.



$$I(\theta) = a_i^2 \cdot \frac{(P_i(\theta) - c_i)^2}{(1 - c_i)^2} \cdot \frac{Q_i(\theta)}{P_i(\theta)}, \text{ where } Q_i(\theta) = 1 - P_i(\theta).$$

Adaptive Difficulty – Knowledge Estimation

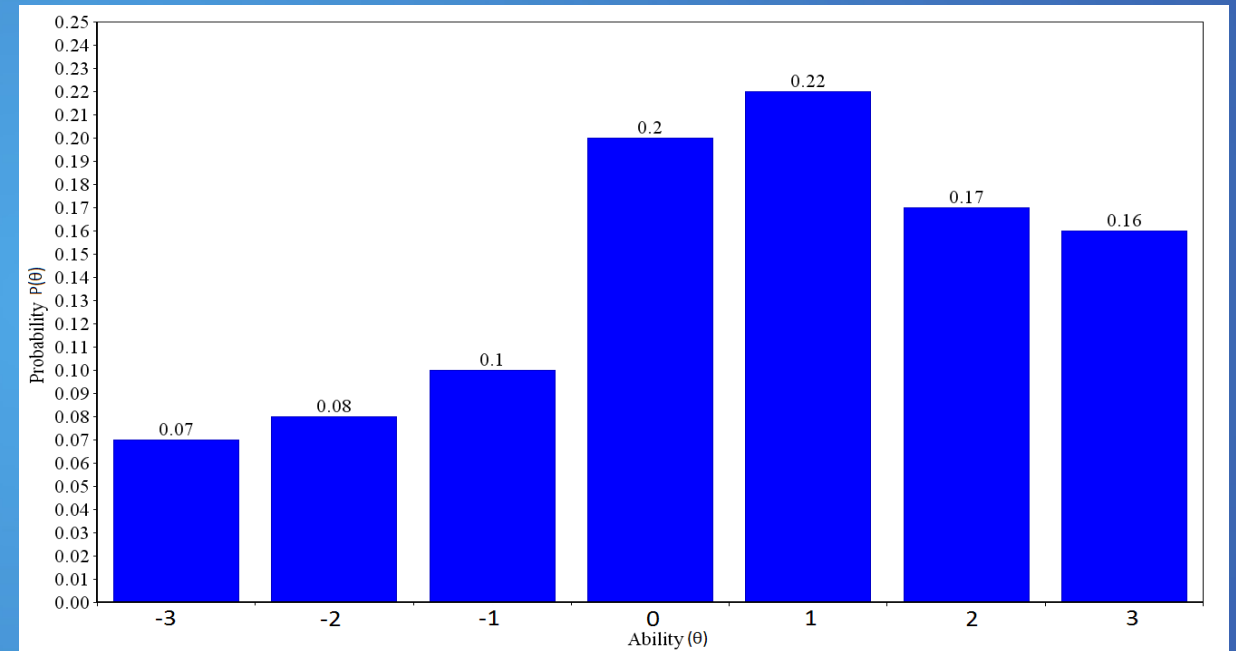
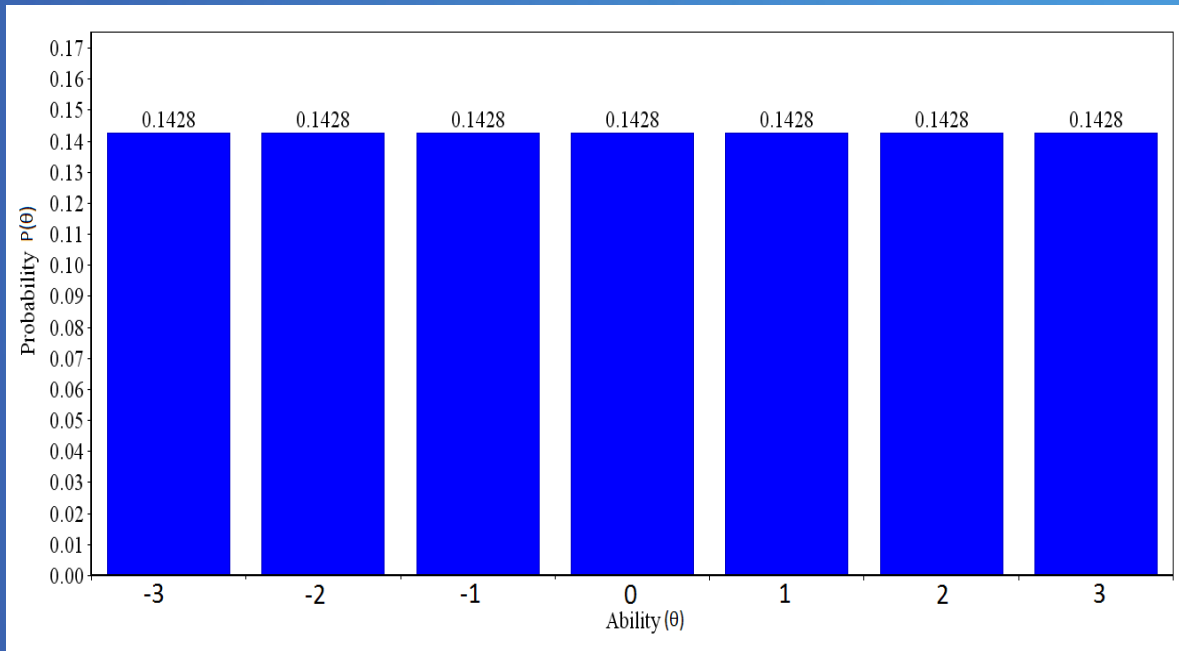
- Knowledge estimation techniques → jSCAPE uses Bayesian estimation.
- CAT maintains a knowledge distribution $P(\theta)$, the probability that the student's ability is θ .
- Initial knowledge distribution is uniform distribution.
- Knowledge distribution updated according to Bayes' rule and Bayesian inference.

$$P(\theta|\mathbf{u}) = \frac{P(\mathbf{u}|\theta)P(\theta)}{P(\mathbf{u})}$$

- \mathbf{u} is the student's response vector.
 - The mode (highest value) of the knowledge distribution is the new ability estimate.
-

Adaptive Difficulty – Knowledge Dist. Update

- Knowledge distribution update after student answers correctly a medium difficulty exercise.



Adaptive Difficulty in jSCAPE

- Implementation details based on SIETTE.
 - $\theta = 0$ to 10, 11 discrete knowledge levels.
 - Calibrated item pool, parameters assigned manually, $0.5 < a < 1.5$ and $0 \leq b \leq 10$, $c=0.25$
 - Starting point: assume student is of average ability, i.e. $\theta = 5$
 - Exercise selection algorithm: *maximum information method*.
 - Scoring algorithm: *Bayesian estimation method*.
 - Termination criterion: exiting jSCAPE and changing exercise category.
-

Exercises and Exercise Generation

```
1 <?xml version="1.0"?>
2 <exercise>
3     <display>
4         <view>.....</view>
5         <value>.....</value>
6     </display>
7     <display>
8         <view>.....</view>
9         <value>.....</value>
10        <choice0>.....</choice0>
11        <choice1>.....</choice1>
12        <choice2>.....</choice2>
13        <choice3>.....</choice3>
14        <solution>....</solution>
15    </display>
16    <display>
17        <difficulty>.....</difficulty>
18    </display>
19 </exercise>
```

Listing 5.9: Exercise format.

```

1 <?xml version="1.0"?>
2 <exercise>
3   <display>
4     <view>CodeEditor</view>
5     <value>public class ConditionalsExercise {
6       public static void main(String[] args) {
7         boolean var1;
8         boolean var2;
9         int var3;
10        int var4;
11
12        var1 = true;
13        var2 = false;
14        var3 = 150;
15        var4 = 300;
16
17        if (var2) {
18          var2 = var1;
19          var1 = true;
20        } else {
21          var3 = var4;
22        }
23
24      }
25    }</value>
26  </display>
27  <display>
28    <view>Multiple Choice</view>
29    <value>What is the correct combination of final values?</value>
30    <choice0>var1 = true; var4 = 266</choice0>
31    <choice1>var1 = false; var4 = 348</choice1>
32    <choice2>var1 = true; var4 = 300</choice2>
33    <choice3>var1 = false; var4 = 300</choice3>
34    <solution>var1 = true; var4 = 300</solution>
35  </display>
36  <display>
37    <difficulty>A</difficulty>
38  </display>
39 </exercise>

```

Listing A.3: Example exercise for the Conditionals exercise category

Conclusion

- Result of this project is jSCAPE, a complete system with core features.
 - Exercises and statistics implemented successfully.
 - However, more work needs to be done on the adaptive difficulty and exercise generation components!
-

Future work

- Other programming languages, e.g. cSCAPE, hSCAPE.
 - More exercise types, e.g. fill-in-blank, multiple response, interactive exercises, etc...
 - More feedback after a student answers an exercise.
 - Improve adaptive difficulty component of jSCAPE.
 - More scalable code generator.
 - Different approach to automated exercise generation.
-

Questions ?
