Automatic Evaluation of Tasks for Instantaneous Diagnostics in Computer Science Lessons

Seminar - USI - Faculty of Informatics

Mike Barkmin ■ 26 February 2020
Outline

1. Introduction
2. Background
3. Considerations
4. The Online-Assessment-System
5. Summary
6. Next Steps

Figure: Picture of TeroVesalainen under Pixabay License via Pixabay
Introduction
Who am I?

Mike Barkmin

Computer Science Education Research Group
University of Duisburg-Essen, Germany
What is my main research area?
What will I show you today?

OpenPatch

26 February 2020
Background
Background

In studies about the structure of programming knowledge we encountered some problems.

Figure: Picture of Ag Ku under Pixabay License via Pixabay
In studies about the structure of programming knowledge we encountered some problems.
Background

In studies about the structure of programming knowledge we encountered some problems

- Digitalisation and following analysis is very time-consuming

Figure: Picture of Ag Ku under Pixabay License via Pixabay

mike.barkmin@uni-due.de

Automatic Evaluation and Visualization of Assessments
In studies about the structure of programming knowledge we encountered some problems

- Digitalisation and following analysis is very time-consuming
- A bigger sample would be hard to manage

Figure: Picture of Ag Ku under Pixabay License via Pixabay
Background

In studies about the structure of programming knowledge we encountered some problems

- Digitalisation and following analysis is very time-consuming
- A bigger sample would be hard to manage
- Complex task formats are difficult to realize

Figure: Picture of Ag Ku under Pixabay License via Pixabay
In studies about the structure of programming knowledge we encountered some problems:

- Digitalisation and following analysis is very time-consuming
- A bigger sample would be hard to manage
- Complex task formats are difficult to realize
- Feedback for teachers is staggered

Figure: Picture of Ag Ku under Pixabay License via Pixabay
Considerations
Considerations I

- Webapplication (WA)
- No need for a user account ⇒ Access to the test with a token (NUA)
- Analysis of the problem-solving-capabilities through capturing the interactions (UIT)
- GDPR: partly encrypted submissions and self-hostable (DS)
- Ability to create items and tests (ITE)
- Ability to create new task formats (CE)
- Ability to download all data for further analysis or provided analysis
### Considerations II

<table>
<thead>
<tr>
<th></th>
<th>WA</th>
<th>NUA</th>
<th>UIT</th>
<th>DS</th>
<th>ITE</th>
<th>CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>JACK (Goedicke and Striewe, 2017)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>VILLE (Rajala et al., 2016)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TRAKLA2 (Laakso et al., 2004)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>BOSS2 (Joy et al., 2005)</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>ProGoSS (Gluga et al., 2011)</td>
<td>✓</td>
<td></td>
<td></td>
<td>x</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>QuizJET (Hsiao et al., 2008)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>x</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

- Additionally, we analyzed other systems (Mooshak, Bottlenose, CourseMarker, WeBWorK and more) as well, but none fitted our needs

- ⇒ Custom development was necessary
The Online-Assessment-System
1. Introduction

2. Background

3. Considerations

4. The Online-Assessment-System
   4.1 Technical Realization
   4.2 Conceptual Realization
   4.3 Item-Layer

5. Summary

6. Next Steps

Figure: Picture of TeroVesalainen under Pixabay License via Pixabay
Technical Realization

Automatic Evaluation and Visualization of Assessments

26 February 2020
4. The Online-Assessment-System

4.1 Technical Realization

4.2 Conceptual Realization

4.3 Item-Layer

4.3.1 Analog to Digital

4.3.2 Authentic Task Formats

4.3.3 Examination of the Process
Analog to Digital I

- **Scale**
  - Digitize analog task formats
  - Makes faster evaluation possible
  - Instantaneous visualization
  - **Evaluation**: Choice
  - **Diagnostic Visualization**: Barchart
Analog to Digital II

- Fill-in
  - No “handwriting recognition”
  - **Evaluation**: Regular expressions e.g.:
    "[li]nterface|[Cc]lass"
  - **Diagnostic Visualization**: Word-cloud
  - For use in an empirical study see Striewe et al. (2017)
Desirable to use more authentic task formats

We implemented a source code runner for this

Source code will be compiled and tested on our servers

Evaluation: Unit tests

Diagnostic Visualization: Currently Missing (Percentage of correct unit tests, average time for execution)
Comparatively simple task format, but authentic

Was already used by Hauswirth and Adamoli (2013)

Connects conceptual knowledge with representation of the concepts in a formal language

Idea: Highlight all spots of `<Concept>` in the given source code
Development of Complex Task Formats: Highlighting II

- **Evaluation**: Calculate Cohens Kappa and compare to a cutoff score
- **Diagnostic Visualization**: Heatmap
- For use in an empirical study see Kramer, Barkmin, and Brinda (2019)

```java
public class Clock {
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private timer thread t;

    private void start()
    {
        clockRunning = true;
    }

    private void stop()
    {
        clockRunning = false;
    }
}
```

Automatic Evaluation and Visualization of Assessments

26 February 2020
mike.barkmin@uni-due.de
Examination of the Process

- By just looking at and analyzing the solution, valuable information will be lost
- Idea: Examine the process

Figure: Picture of Bhuvanesh S under Pixabay License via Pixabay
Examination of the Process

- By just looking at and analyzing the solution, valuable information will be lost
- Idea: Examine the process
- **Solution**: Videorecording of the process
  - 8 students approx. 4h ~ 140GB
  - Manual tagging of events

*Figure: Picture of Bhuvanesh S under Pixabay License via Pixabay*
Examination of the Process

- By just looking at and analyzing the solution, valuable information will be lost
- Idea: Examine the process
- **Solution**: Videorecording of the process
  - 8 students approx. 4h ~ 140GB
  - Manual tagging of events
- **Alternative solution**: Recording of the interactions with the Online-Assessment-System
  - approx. 500 students ~ 20MB
  - Auto tagging of events

Figure: Picture of Bhuvanesh S under Pixabay License via Pixabay
Recording of the Process

- **Action**: Is dispatched by the user
- **Reducer**: Constructs a new state based on a dispatched action
- **Store**: Contains the current state
- **UI**: Will be rendered depending on the current state in the store

Figure: Action-Reducer-Store see https://redux.js.org

26 February 2020

mike.barkmin@uni-due.de

Automatic Evaluation and Visualization of Assessments
Recording of the Process

Klassenbezeichner

public class Clock
{
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private TimerThread timerThread;
    public Clock()
    {
        makeFrame();
        clock = new ClockDisplay();
    }
    private void start()
    {
        clockRunning = true;
        timerThread = new TimerThread();
        timerThread.start();
    }
    private void stop()
    {
        clockRunning = false;
    }
    private void step()
    {
        clock.timeTick();
        label.setText(clock.getTime());
    }
    private void makeMenuBar(JFrame frame)
}
Making use of the Recordings
Making use of the Recordings - Visualization

Based on the idea of Parsons and Haden (2006)

Our actions for Parson Puzzles
- MOVE_FROM_SOURCE_TO_USER (sourceld, userld)
- MOVE_FROM_USER_TO_SOURCE (userld, sourceld)
- MOVE_WITHIN_USER (userld1, userld2)

What happens, when the action
MOVE_FROM_SOURCE_TO_USER (1, 1) is dispatched?

Initial State (left: source, right: user)
Making use of the Recordings - Visualization

Initial State (left: source, right: user)

After dispatching the action MOVE_FROM_SOURCE_TO_USER (1, 1)
Making use of the Recordings - Visualization

- Visualization as a directed graph
- Each node represents a state of the parsons puzzle
  - star-shape indicates start state
  - green indicates correct state
- Each edge represents the dispatch of an action
- Number and thickness indicating the frequency

Figure: Visualization of 136 processes based on Helminen et al. (2012)
Making use of the Recordings - Cognitive Structures

class Datei {
    String name;
    String pfad;
    Datei(String name, String pfad) {
        this.name = name;
        this.pfad = pfad;
    }
    void umbenennen(String name) {
        this.name = name;
    }
}
Making use of the Recordings - Cognitive Structures

Actions for task format memorize
- INSERT_CHAR (charId, pos)
- REMOVE_CHAR (pos)
- OPEN_MEMORIZE
- CLOSE_MEMORIZE
Making use of the Recordings - Cognitive Structures

- Every keystroke is recorded
- Many actions are hard to analyze
- Actions must be combined to reduce complexity
- Memorize-Phases (Blue), Write-Phases (Green) and Pause-Phases (Lightblue)
- Empirical study see Barkmin et al. (2017)

![Figure: Timeline of one process](image1)

<table>
<thead>
<tr>
<th>Phases</th>
<th>Zeit in s</th>
<th>LS</th>
<th>G</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>2.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>5.71</td>
<td>-33</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>3.01</td>
<td>-6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.72 (4.88)</td>
<td>-39</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>2.15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2.19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>5.67</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

![Figure: Transcript of one process using combined actions](image2)
public class Clock
{
    private JFrame frame;
    private JLabel label;
    private ClockDisplay clock;
    private boolean clockRunning = false;
    private timerThread t;

    private void start()
    {
        clockRunning = true;
    }

    private void stop()
    {
        clockRunning = false;
    }

    0 %  20 %  40 %  60 %  80 %  100 %
Next Steps
Next Steps

Teacher - Visualization
Study the handling of the visualizations by teachers

Figure: Picture of JESHOOTS-com under Pixabay License via Pixabay

Pattern-Recognition
Automatic Evaluation of the Process

Figure: Picture of GDJ under Pixabay License via Pixabay

26 February 2020
Thank you! Any Questions?

Source Code: https://gitlab.com/openpatch

Website: https://openpatch.app

Contact
Mike Barkmin
Computer Science Education Research Group
Universität Duisburg-Essen
Schützenbahn 70, 45127 Essen
mike.barkmin@uni-due.de
http://udue.de/mba


References IV


mike.barkmin@uni-due.de

CSE