Platform Architecture for the Diagram Assessment Domain

Meike Ullrich, Martin Forell, Constantin Houy, Peter Pfeiffer, Selina Schüler, Tobias Stottrop, Brian Willems, Peter Fettke, Andreas Oberweis
Authors …

- Martin Forell
- Selina Schüler
- Meike Ullrich
- Andreas Oberweis
- Constantin Houy
- Peter Pfeiffer
- Brian Willems
- Peter Fettke
- Tobias Stottrop

Implementation and domain-specific background

... and KEA-Mod project

KEA-Mod: Kompetenzorientiertes E-Assessment für die grafische Modellierung
(Competence-oriented e-assessment for graphical modeling)

Didactical aspects and evaluation

Transfer into community
Outline

Introduction
• Background, challenges, project, related work

Platform Architecture
• Component functionality, requirements

Outlook
• Platform dissemination, research opportunities
In software engineering, modeling in the form of graph-based diagrams is an essential core component of the curriculum.
Introduction » Challenges

- Student solutions tend to be complex and possibly confusing
- No unique solution for a single modeling exercise
- Several important quality aspects (syntax, semantics and pragmatics)
- Lack of consistency and transparency in case of multiple „correctors"

Manual marking is a tedious, error-prone and time consuming task!
Introduction » Project KEA-Mod

- Development of an e-assessment platform for the diagram domain
- Key features:
  - Methodically grounded approach
  - Competence-oriented exercise types
  - Supports various diagram types (UML-AD, UML-Class, BPMN, EPC, Petri nets for a start)
  - Adaptable and expandable
  - Merges existing solutions into a uniform concept
Introduction » Related Work

Publication of systematic literature review in progress

- High level approaches
  - Design universal solutions for various diagram types (i.e. modeling languages)
  - Integration of assessment functionality into a larger e-learning context
  - Currently not widely distributed and lack of support of a comprehensive set of diagram types

- Low level approaches
  - Solutions geared towards a specific use case (e.g., identifying patterns in UML class diagrams)
  - Assessment yields individual results (varying feedback detail, total score quantification is rare)

- Coverage of diagram types
  - Mostly UML class diagrams and ER diagrams
  - Rather few approaches cover other UML diagram types, EPCs and Petri nets
  - No known approach for BPMN (even though its popularity in both academia and industry)

Need for a comprehensive and uniform diagram assessment platform
Platform Architecture » Overview

- Microservice architecture
  - Different inter-operable components and systems

- Interaction with end user through Frontend components (web applications)

- Internal communication between Backend components and Grading Services
Assignments are collections of exercises (like an exercise sheet or exam)

Exercises can have different types. Main types supported up to now:

Diagram understanding

Exercise 1: Analyze the given ER diagram and select only statements (A)-(E) which are actually true according to this diagram.

Diagram creation

Exercise 2: Use BPMN to create a business process model fragment that depicts the following situation description:

When registering for a course, students have to form a team and complete a test. Students can complete this test either online or at the university, depending on whether they are locally available.

Automatic assessment is trivial, but covers only basic competences

Addresses higher levels of competences, but automatic assessment is challenging
Grading schemas (tree-based) can be assigned to exercises to configure the automatic assessment.

- Calculation of a total score (= root) based on weighted results of individual grading services (= leaves).
- Nodes can be used to group (and re-use) specific combinations.
Platform Architecture » Author UI

- Web application for authors (e.g. lecturers, course instructors, tutors)
- Create, configure and review exercises and grading schemas for the automatic assessment
- Public exercise pool with established exercises
- Optional manual annotation and correction of automatic assessment during review
Platform Architecture » Student UI

- Web application for students
- Work on assignments and view assessment results
- Usage of modeling editor
- Transparent total score and additional textual feedback
- Incorrect diagram elements are highlighted as such directly inside the diagram and linked to textual feedback
Platform Architecture » Modeling Editor

- Stand-alone web application which is integrated in both the Student and Author UI
- Offers a vast amount of graphical diagram elements of various modeling languages
- Can be configured per exercise to select the available graphical diagram elements and enable or disable live syntax checking

Core Application

Frontend

“component” : Modeling Editor

“component” : Author UI

“component” : Student UI

Identity Provider

“component” : Backend

“component” : Request Management

“component” : Assessment Module

“component” : Grading Service n

“substitute”

“component” : Grading Service 1

“system” : Grading Queue

“system” : Database

Ex-/Import

Ex-/Import

Ex-/Import

Ex-/Import
Platform Architecture » Request Management

- Processes incoming requests via the Frontend components
  - Basic CRUD operations
  - Communication with Assessment Module

- Authentication and authorization performed by querying external identity providers

- Only pseudonymized identifier stored in the database
Platform Architecture » Assessment Module

- Triggered when exercise submissions have to be graded
- Input: Exercise grading schema and student submission
- Hands over grading requests to the grading queue and processes the corresponding results
- Output: Total score and generated feedback
Platform Architecture » Grading Queue and Grading Services

- The Grading Queue stores both incoming requests for grading and grading results.

- A Grading Service application runs a specific analysis on a student submission and returns a numerical score and feedback list as result.

- Grading Services can be language-specific or general.

Examples:

- Syntax checker
  - basic petri net
  - workflow net

- Validity of diagram facts
- Completeness of diagram facts
- Edge crossings
- Label consistency
- Flow direction consistency
## Platform Architecture » Non-functional requirements

<table>
<thead>
<tr>
<th>Frontend usability design</th>
<th>Data protection and security</th>
<th>Technical expandability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Easy to use</strong></td>
<td><strong>Safe to use</strong></td>
<td><strong>Easy to extend</strong></td>
</tr>
<tr>
<td>• Simple, intuitive and consistent graphical design</td>
<td>• Does not store any personal data, only a pseudonymized identifier</td>
<td>• Geared towards further expandability and re-use</td>
</tr>
<tr>
<td>• Media-pedagogic principles and usability guidelines are considered</td>
<td>• Communication is encrypted on each communication channel</td>
<td>• Modeling Editor supports flexible configuration of new diagram types</td>
</tr>
<tr>
<td>• Usability evaluation will be performed</td>
<td></td>
<td>• Due to microservice architecture, assessment solutions can be integrated as grading services</td>
</tr>
</tbody>
</table>
Platform Architecture » Technical realization

Ex-/Import

Core Application

Frontend

«component» :Modeling Editor

«component» :Author UI

«component» :Student UI

Identity Provider

«component» :Request Management

«component» :Assessment Module

«system» :Database

Ex-/Import

Ex-/Import

«component» :Grading Service n

«component» :Grading Service 1

«system» :Grading Queue

«component» :Grading Queue

Image sources: see respective project homepages, accessed 02/2021
Outlook » Platform dissemination

Pilot phase
Universität Duisburg-Essen
Universität des Saarlandes
Karlsruher Institut für Technologie

Starting Q3/2021

Transfer phase
Universität Kiel
HTW Saar
HTWG Konstanz
Universität Regensburg
Hochschule Pforzheim
Hochschule Kaiserslautern
DHBW Karlsruhe
...

Interested? Sign up to our newsletter at: https://keamod.gi.de/kontakt

Image source: https://www.intermap.de/html/deutschlandkarte/dospol.html
Outlook » Research opportunities

- During the diagram creation, user actions are logged (completely anonymously) by the Modeling Editor component.
- Log data can be analyzed to gain insights about the modeling process itself (Process Mining).

- Automatically generated assessments that have been manually reviewed and approved might serve as training data for the development of innovative machine-learning-based grading services.

- Exercises of the type diagram understanding can be generated automatically by external exercise generators.
- Here, answer options for single/multiple choice questions corresponding to a given diagram are generated automatically.
Thank you for your attention!
Any questions?

Meike Ullrich, Martin Forell, Constantin Houy, Peter Pfeiffer, Selina Schüler, Tobias Stottrop, Brian Willems, Peter Fettke, Andreas Oberweis
Platform Architecture » Overview

Core Application

Frontend

- «component»: Modeling Editor
- «component»: Author UI
- «component»: Student UI

Identity Provider

«component»: Request Management

Backend

- «component»: Grading Service n
- «substitute» «component»: Grading Service 1
- «system»: Grading Queue

Ex-/Import

Ex-/Import

Ex-/Import

«system»: Database
Platform Architecture » Backup