

aus

Oliver J. Bott, Peter Fricke, Uta Priss, Michael Striewe (Hrsg.)

Automatisierte Bewertung in der Programmierausbildung

Digitale Medien in der Hochschullehre Band 6

2017, 420 Seiten, br., 42,90 €, ISBN 978-3-8309-3606-0



Waxmann Verlag GmbH

www.waxmann.com info@waxmann.com

22 Integration mithilfe der Middleware ProFormA-Server

Oliver Rod

Zusammenfassung

An den Hochschulen gibt es häufig verschiedene Lernmanagementsysteme und spezialisierte Bewertungssysteme. Die ProFormA-Middleware verbindet diese beiden Typen von Systemen miteinander. Im folgenden Kapitel wird die Middleware vorgestellt. Der Fokus liegt auf den Schnittstellen und den Abläufen in und mit der Middleware. Dabei wird auf die Praktiken bei der Implementierung an der Ostfalia Hochschule verwiesen.

22.1 Einleitung

In diesem Buch werden in den Kapiteln 19, 20 und 21 verschiedene Lernmanagementsysteme mit ihren jeweiligen Besonderheiten und Schwerpunkten vorgestellt. Die Systeme haben unter anderem folgende Funktionen:

- Kursverwaltung (Inhaltsverwaltung, Kurse),
- verschiedene Kommunikationsmöglichkeiten zwischen Lehrenden und Lernenden (Forum, Chat, Nachrichten, Kalender),
- Unterstützung gängiger Aufgabentypen (Multiple Choice, Lückentext, Drag and Drop) und
- Verwaltung von Lernobjekten, Aufgaben, Tests.

Teile dieses Kapitels entstanden im Rahmen des Projekts eCULT, Teilvorhaben eAssessment, gefördert durch das Bundesministerium für Bildung und Forschung unter dem Förderkennzeichen 01PL16066H. Die Verantwortung für den Inhalt dieses Kapitels liegt bei dem Autor.

Die genannten Aufgabentypen der Lernmanagementsysteme können die Anforderungen der technischen Studiengänge möglicherweise nicht abdecken. Die Herausforderung besteht darin, hochgeladene Programmcodes oder Modellierungssprachen zu analysieren und zu testen. In den vorangegangenen Kapiteln dieses Buchs (siehe Abschnitt II) wurden bereits einige Systeme zur automatischen Programmbewertung vorgestellt. Diese Bewertungssysteme legen ihren Schwerpunkt hingegen auf:

- automatisierte Bewertung von Einreichungen,
- Sicherheit in der Ausführung der Einreichung für das ausführende System,
- angepasste Tests an die jeweiligen Aufgaben und Sprachen,
- Verwendung von Werkzeugen in der Softwareentwicklung wie Compiler, Unit-Tests, Versionierung und Style-Tester,
- zusätzliches Feedback, welches über die Kompilerausgabe hinaus geht.

Trotz der Menge an Bewertungssystemen ist es schwierig ein Werkzeug zu finden, das allen Anforderungen eines Lehrenden gerecht wird. Diese Anforderungen bestehen in den verwendeten Programmiersprachen, den jeweiligen Testarten und den verschiedenen Lehrstilen.

Die ProFormA-Middleware ermöglicht eine einfache Koppelung der Lernmanagementsysteme und der Bewertungssysteme miteinander und schafft somit die Voraussetzungen, um die genannten Vorteile der beiden Systeme zu verbinden. Die vorgeschlagene Middleware soll mehrere Bewertungssysteme mit jeweils einem Lernmanagementsystem koppeln, um den Lehrenden mehr Freiheiten in ihren jeweiligen Aufgabenstellungen zu gewährleisten.

Ein Fokus bei der Interoperabilität der Bewertungssysteme liegt im Austausch der Aufgaben. Diese sind in ihrer Erstellung viel komplexer und zeitintensiver im Vergleich zu gängigen Aufgabentypen (siehe [PJR12]). Als universelles Aufgabenformat für die Programmieraufgaben nutzt die Middleware das ProFormA-Aufgabenformat (vgl. Kapitel 24). Ein weiterer Vorteil bei der Verbindung von Lernmanagementsystem und Bewertungssystem für Studierende und Lehrende, basiert auf der Verwendung einer Benutzeroberfläche.

22.2 System und Ablauf

Die ProFormA-Middleware wurde an der Ostfalia Hochschule entwickelt und verbindet ein LMS, LON-CAPA (siehe Kapitel 19), mit einem Bewertungssystem, Praktomat (vgl. Kapitel 10). Die Middleware ist hierbei nach dem KISS-Prinzip¹ aufgebaut. Es wurde versucht die Komplexität und den Aufwand bei der Implementierung der Schnittstellen für das LMS und die jeweiligen Bewertungssysteme so gering wie möglich zu halten. Nur wenn viele Systeme die Schnittstellen unterstützen, kann die Middleware ihre Vorteile nutzbar machen. Denn auch das LMS und die Bewertungssysteme müssen eine Schnittstelle bereitstellen, welche die Middleware unterstützt.

Die Abbildung 22.1 zeigt exemplarisch die verbundenen Systeme und die jeweiligen Schnittstellen. Der Aufgabenpool (auch Repository) ist in der Abbildung als eigenständiges System dargestellt, wird aber an der Ostfalia derzeit mittels dem Aufgabenpool von LON-CAPA realisiert. Die zentralen Aufgaben der Middleware sind die Verwaltung der angeschlossenen Systeme und, bei Bedarf, eine Übersetzung der jeweiligen Antwortformate ineinander. Studierende und Lehrende verwenden ihren Browser für die Nutzung oder die Konfiguration des Lernmanagementsystems. Zusätzliche Zugänge zu den verwendeten Bewertungssystemen werden nicht benötigt. Die Middleware verwaltet die angeschlossenen Bewertungssysteme und bietet für das LMS eine definierte Schnittstelle (siehe Kapitel 22.3.1). Die Konfiguration der angeschlossenen Bewertungssysteme wird in einer Konfigurationsdatei auf der Middleware verwaltet. Die Bewertungssysteme

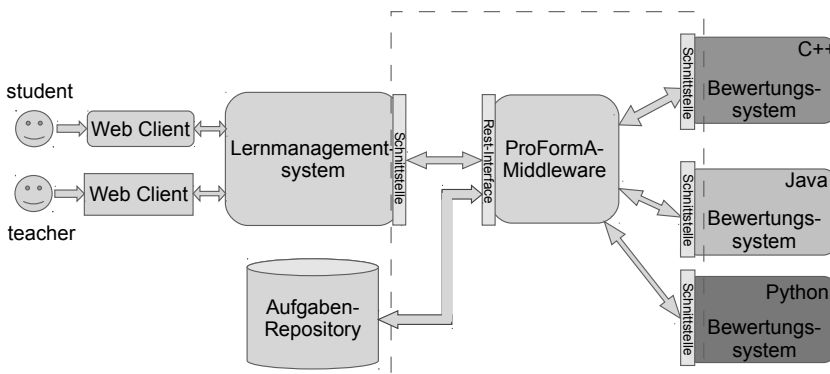


Abbildung 22.1: Architektur der Middleware

¹ Keep it short and simple

können unterschiedliche Programmiersprachen, wie zum Beispiel C++, Python oder Java, aber auch unterschiedliche Testverfahren, wie Unit-Tests oder Blackboxtests unterstützen. Für die Bewertung einer Aufgabe sind nur die studentische Einreichung und die Programmieraufgabe im XML-basierten Austauschformat (ProFormA-XML) notwendig. Die Kommunikation über die Middleware verläuft derzeit immer synchron. Das Ergebnis einer studentischen Einreichung kann somit nicht zu einem späteren Zeitpunkt abgerufen werden. In Kapitel 22.3 folgt die Beschreibung der Schnittstellen.

22.3 Schnittstellen

Es gibt für die vorgeschlagene Architektur drei wichtige Schnittstellen, welche definiert werden müssen.

LMS verwendet ein einfaches Web-Interface, welches Daten versendet und über ein definiertes Antwortformat Ergebnisse auswerten kann.

Middleware REST-Schnittstelle für den definierten Informationsaustausch zwischen Bewertungssystemen und Lernmanagementsystemen.

Bewertungssystem benötigt ebenso eine REST-Schnittstelle für die Erstellung und Bewertung von Programmieraufgaben.

22.3.1 REST-API der Middleware

Die URI für die Einreichung ist detailliert in Abbildung 22.2 veranschaulicht. Es wird über die URI-Parameter *fw* und *fw-version* das benötigte Framework und die Version übermittelt. Hiermit ist die benötigte Programmiersprache gemeint. Die Middleware weist der Einreichung somit das benötigte Bewertungssystem zu. Grundsätzlich benötigt jede Einreichung drei allgemeine Informationen:

- die eingereichte Aufgabe im ProFormA-Aufgabenformat,
- das erwartete Antwortformat,
- die studentische Lösung.

Um die zu übermittelnden Daten so gering wie möglich zu halten, wird nur der Ort der Aufgabenressource mittels *task-repo* und *task-path* versendet. Eine direkte Übermittlung der Aufgaben erfolgt im XML-Format. Das LMS hat bestimm-

te Formvorgaben, in der es die Antwort erwartet. Das verwendete LMS, LON-CAPA, hat ein definiertes XML-Antwortformat². Sollten die Lernmanagementsysteme keine externe Schnittstelle anbieten, könnte es sinnvoll sein, ein standardisiertes Antwortformat³ für das LMS vorzugeben. Die studentische Einreichung kann entweder direkt im HTTP-POST-Parameter *submission* übergeben werden oder sie wird mittels *submission-uri* zusammen mit dem Ort der Einreichung übermittelt. Die Interaktionen zwischen LMS und Middleware und zwischen Middle-

POST
api/v1/grading/prog-languages/:fw:fw-version/submissions

- Einreichung der Problemlösung

URI-Parameter:

Feld	Typ	Beschreibung
fw	String	Programmiersprache bzw. Framework welches unterstützt werden soll
fw-version	String	Version des Frameworks

HTTP-Parameter:

Feld	Typ	Beschreibung
task-repo	String	IP bzw Domain des Repositories
task-path	String	Name der Aufgabe mit Pfad
answer-format	String	Name und Version des Antwortformates
submission	String	Inhalt der Einreichung
submission-uri	String	URI wo die Einreichung liegt

Abbildung 22.2: URI und Parameter für die Einreichung

ware und Bewertungssystemen sollten immer über HTTPS verlaufen, um eine zusätzliche Sicherheit zu bieten. Des Weiteren müssen die verbundenen Systeme über ihre IP-Adresse beziehungsweise IP-Bereiche freigegeben werden. Es wurde von einer Schnittstelle zur Administration der Middleware abgesehen, zumal diese ein unnötiges Sicherheitsrisiko bei zu geringem Nutzen darstellen würde. Eine Entkopplung von IP-Freigabe und Freigabe neuer Bewertungssysteme sollte direkt auf dem Server der Middleware vorgenommen werden. Eine zusätzliche

² https://vita.ostfalia.de/adm/help/Authoring_ExternalResponse.hlp

³ <https://github.com/ProFormA/responsexml>

Authentifizierung über Token oder HTTP Authentication nach RFC 2617⁴ ist geplant.

22.3.2 Schnittstelle von Bewertungssystemen

Bewertungssysteme laufen grundsätzlich autark und bieten meist keine externe Schnittstelle an. Der Vorschlag ist hier, eine externe Webschnittstelle zu implementieren. Über diese Schnittstelle kann die Middleware die Aufgabe mittels ProFormA-Aufgabenformates erstellen und anschließend die studentische Einreichung bewerten lassen. An der Ostfalia wurde das Bewertungssystem Praktomat⁵ um diese vorgeschlagene REST-Schnittstelle mit zwei URIs erweitert. Einen Vorschlag für diese beiden URIs veranschaulicht Abbildung 22.3a.

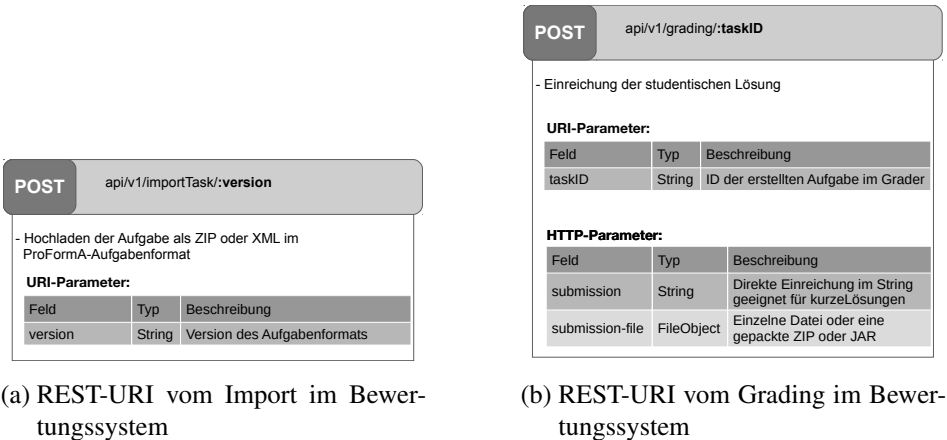


Abbildung 22.3: Übersicht der REST-URIs des ProFormA-Servers

Für den Import muss die Version der verwendeten ProFormA-XML angegeben werden. Bei erfolgreicher Erstellung wird die Identifikation des Tasks zurückgegeben (TaskID), welche für die eigentliche studentische Einreichung benötigt wird. Für die Antwort wird ein einfaches JSON-Schema mit *TaskID* und *message* empfohlen.

⁴ <https://www.ietf.org/rfc/rfc2617.txt>

⁵ <https://github.com/KITPraktomatTeam/Praktomat>

Die erhaltene *TaskID* wird für die *Grading-URI* benötigt und die eigentliche Einreichung wird entweder direkt als String im Feld *submission* übergeben oder als Datei gesendet. Nach dem erfolgten Testen der Einreichung wird auch hier die Antwort in einem JSON-Format erwartet. Die Antwort sollte sich hier an die Vorgaben des Antwortformates halten. Im Optimalfall erspart dies der Middleware eine Umwandlung verschiedener Antwortformate.

22.4 Aufbau der Middleware

Die Middleware wurde mit dem Webframework django⁶ entwickelt und folgt dem Model-View-Controller-Schema. Der einfache Aufbau der Middleware wird in Abbildung 22.4 veranschaulicht. Alle Aufrufe über das Rest-Interface laufen über den Controller, welcher bei Bedarf mit den einzelnen Modulen interagiert. Zum derzeitigen Stand (Juli 2016) wurde zunächst ein Bewertungssystem (*grader_praktomat*) mit der Middleware verbunden. Dieses Modul behandelt die Interaktion mit dem Praktomaten sowie die Erstellung (*create_task*) und das Einreichen der studentischen Lösung (*grade_Task*). LON-CAPA dient übergangsweise auch als Repository und wird im *repo_handler* bearbeitet. Die Einreichungen können auch einen Verweis auf den Ort (*submissions-uri*) enthalten. Dieser externe Verweis wird innerhalb des Moduls *extern_submission* abgehandelt. Derzeit wird nur LON-CAPA für externe Verweise unterstützt.

Das Modul *answerformat_converter* soll sich um die Umwandlung der Antwortformate kümmern. Oftmals ist dies nicht notwendig, aber sobald mehrere Lernmanagementsysteme unterstützt werden, wird dies unausweichlich sein.

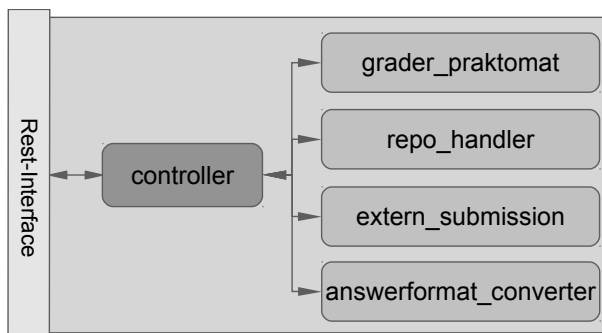


Abbildung 22.4: Systemübersicht der Middleware

⁶ <https://www.djangoproject.com>

22.5 Ablauf

In diesem Abschnitt soll auf einzelne Abläufe in der Verwendung der Middleware eingegangen werden. Die folgenden Phasen beschreiben den Prozess den ein Lehrender bei der Erstellung einer Programmieraufgabe durchläuft. Der Zyklus einer Programmieraufgabe wird in folgende Phasen unterteilt:

1. allgemeine Aufgabenstellung und Tests im eigenen System,
2. Formulierung der Aufgabe im ProFormA-XML-Format unter Zuhilfenahme des Editors⁷,
3. Upload der XML- oder der ZIP-Datei in das Repository
4. Einrichtung der Aufgabe im LMS
5. Testen der Aufgabe im LMS

Aus der Praxis ist bekannt, dass diese Sequenz von Lehrenden mehrfach durchlaufen werden muss, um die Tests und die Aufgabenstellung so präzise wie möglich zu gestalten. Häufig werden Probleme erst nach der ersten Nutzung im Kurs sichtbar. Eine Optimierung der Aufgabenstellung ist zu jedem Zeitpunkt möglich, es müssen dann einzelne Schritte der Sequenz wiederholt werden. Für die Erstellung einer Programmieraufgabe wird die Middleware *nicht* benötigt. Es wird zudem zu keinem Zeitpunkt der direkte Zugriff auf das Bewertungssystem von Lehrenden oder Studierenden benötigt.

Ein direkter Zugriff auf die Middleware ist nur für die IP-Freischaltungen der verbundenen Lernmanagementsysteme, Bewertungssysteme und Aufgabenpools notwendig. Dieser Vorgang wird nicht von Lehrenden, sondern einem Administrator vorgenommen. Die Anzahl der angeschlossenen Systeme an die ProFormA-Middleware ist nicht begrenzt. Einzig die Anzahl der gleichzeitig zu bearbeitenden Anfragen ist limitiert. In der Konfigurationsdatei der Middleware erfolgt die Zuordnung der Bewertungssysteme zu einer Grading-Engine.

Die Abbildung 22.5 verdeutlicht den Ablauf einer studentischen Einreichung. In diesem Szenario gibt die Studierende ihren Programmcode in ein Textfeld ein und sendet ihre Einreichung ab. Das Lernmanagementsystem sendet die Einreichung mit dem Verweis auf die ProFormA-XML und dem zu verwendenden Bewertungssystem an die Middleware.

⁷ <https://github.com/ProFormA/formatEditor>

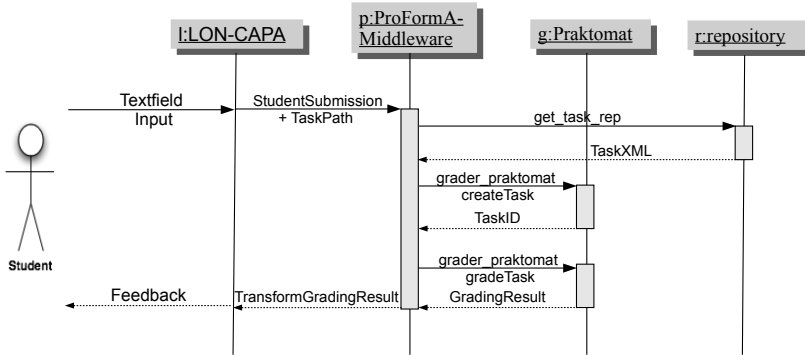


Abbildung 22.5: Sequenzdiagramm einer studentischen Einreichung

Die Middleware lädt die Aufgabe aus dem Aufgabenpool herunter und leitet diese an das gewählte Bewertungssystem, den Praktomaten, weiter. Anschließend erstellt das Bewertungssystem die Aufgabe und gibt die *TaskID* zurück. Jetzt kann die studentische Einreichung mit Hilfe der *TaskID* an die neu erstellte Aufgabe gesendet werden. Nach der automatisierten Bewertung der Einreichung gibt das Bewertungssystem das Ergebnis an die Middleware zurück. Diese übermittelt die Lösung im geforderten Antwortformat an das LMS.

Abschließend bekommt der Studierende das Ergebnis seiner Lösung in LON-CAPA dargestellt. Abbildung 22.6 zeigt beispielhaft das Ergebnis einer fehlerhaften Einreichung in LON-CAPA. Die Anzeige ist in drei Bereiche unterteilt:

- die Aufgabenstellung und die bereits getätigten Einreichungen,
- das Eingabefeld für die Einreichungen mit integriertem Syntax-Highlighting und
- das Ergebnis der Einreichung mit den einzelnen Testergebnissen.

Alle studentischen Daten, wie die Anzahl der Versuche, die getätigten Einreichungen und die Ergebnisse liegen im LMS vor. Die Kursleitung hat hierdurch alle notwendigen Daten innerhalb des Lernmanagementsystem vorliegen und muss keine weiteren Informationen aus den Bewertungssystemen abfragen. Die Kursleitung hat somit die Möglichkeit, die aktuellen Probleme und den aktuellen Kenntnisstand der Studierenden zu erkennen und in den Veranstaltungen gezielt darauf einzugehen.

Schreiben Sie ein Programm, das "Hello World!" (ohne die Anführungsstriche) ausgibt. Das Programm soll eine Methode enthalten, welche den String "Hello World!" zurückgibt. Die Klasse soll "HelloWorld" und die Methode soll "greet" heißen.

Ihre Einreichungen:

► Einreichung 1

[Hinweise zur Benutzung](#)

Geben Sie im folgenden Feld Ihre Lösungsdatei ein:

```

1 public class HelloWorld {
2
3     public static String hallo() {
4         return "Hello World!";
5     }
6
7     public static void main(String[] args) {
8         System.out.println(hallo());
9     }
10 }
```

Aufgabe: HelloWorld

Mindestens ein Test wurde nicht bestanden.

Java - Compiler ► **bestanden**

JUnit Test: Java JUnit Test ► **nicht bestanden**

```

***** Test Results *****

1 Java user-submitted files found for compilation: HelloWorld.java
Java compiler output:
HelloWorldTest.java:7: error: cannot find symbol
    assertEquals(h.greet(), "Hello World!");
                  ^
    symbol:   method greet()
    location: variable h of type HelloWorld
1 error
1
```

Datei: HelloWorld.java

Antwort einreichen Inkorrekt. Versuche 1 [Bisherige Antworten](#)

Abbildung 22.6: Bildschirmfoto einer fehlerhaften Einreichung

22.6 Eigene Erfahrungen und Ausblick

Die vorgestellte Architektur ist seit dem Wintersemester 2014 an der Ostfalia Hochschule im Einsatz. Sie wurde anfangs nur in der Informatik und der Elektrotechnik für die Einführungsveranstaltungen der Programmierung mit Java genutzt. Mittlerweile werden auch Aufgaben im mathematischen Kontext mit SetlX (vgl. Kapitel 8) verwendet. Die Verwendung von LON-CAPA ermöglicht es auch

den Dozenten an anderen Standorten mit LON-CAPA Zugriff auf bereits erstellte Aufgaben der Ostfalia zuzugreifen. Der vorgeschlagene Austausch von Programmieraufgaben wurde somit in einigen Fällen bereits genutzt. Viele der an der Ostfalia betreuten Dozenten haben sich einverstanden erklärt ihre Aufgaben mit anderen Dozenten zu teilen. Gleichzeitig gab es die Bereitschaft Programmieraufgaben von anderen Lehrenden zu nutzen. Der Austausch von Aufgaben ist einer der zentralen Punkte von LON-CAPA (siehe [KC09]). Gerade in der akademischen Kultur ist das Teilen von Wissen essenziell um Erkenntnisse zu teilen und auf diesen aufzubauen.

Die exemplarische Implementierung der ProFormA-Middleware mit dem Lernmanagementsystem LON-CAPA und dem Bewertungssystem Praktomat ist erfolgreich verlaufen. Für die Zukunft ist die Integration weiterer Bewertungssysteme wie auch weiterer Lernmanagementsysteme angestrebt um so die Nutzerchaft und den Nutzen der Middleware weiter zu erhöhen. Die Themen Caching und Lastverteilung auf mehrere Bewertungssysteme werden somit in Zukunft an Bedeutung gewinnen. Hier bietet es sich an, diese in die weiterführenden Entwicklungen der Middleware zu implementieren.

Weiterhin ist geplant, die Middleware ProFormA-Server (Kapitel 22) und die Middleware Grappa (Kapitel 23) mit einer wechselseitigen Schnittstelle auf der Basis des Aufgaben-Austauschformats (Kapitel 24) auszustatten, so dass am Ende alle von einer Middleware unterstützten LMSe mit allen von der jeweils anderen Middleware unterstützten Gradern kombiniert werden können.

Literatur für dieses Kapitel

- [KC09] Gert Kortemeyer und E. Cruz. „LON-CAPA – An Open-Source Learning Content Management and Assessment System“. In: *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*. 2009, S. 1515–1520.
- [PJR12] Uta Priss, Nils Jensen und Oliver Rod. „Software for Formative Assessment of Programming Exercises“. In: *elearning Baltics (2012)*, S. 63–72.