**OPALADIN – ALADIN goes OPAL**

Dieser Beitrag stellt OPALADIN vor, eine Erweiterung von ALADIN (Generator für Aufgaben und Lösung(shilf)en in der Informatik und angrenzenden Disziplinen).

ALADIN bietet ein Framework zur deklarativen Modellierung von Aufgabentypen und zur automatischen Generierung von Aufgaben und Lösungen. ALADIN verfügt zudem über eine graphische Nutzeroberfläche zur interaktiven Bearbeitung von personalisierten Übungen und für einen asynchronen Austausch über, und ein Nachvollziehen von Lösungsversuchen mittels des 4R-Prinzips (Record, Redirect, Replay, Resume).

ALADIN verringert die hohen Aufwände in der manuellen Erstellung von Aufgaben, passt Übungsaufgaben an die individuellen Bedürfnisse der Lernenden an und ermöglicht orts- und zeitunabhängiges Lernen. ALADIN bietet jedoch keine Möglichkeit Lernenden vor und nach der Bearbeitung der Aufgaben weitere Lehr- und Lehrinhalte zur Vermittlung der, zur Lösung der Aufgabe benötigten, fachlichen Inhalte darzubieten.

OPALADIN ergänzt ALADIN um eine Learning Tools Interoperability (LTI) Schnittstelle und integriert ALADIN damit in Lernmanagementsysteme (LMS) wie OPAL, Moodle oder ILIAS. OPALADIN ermöglicht durch die LTI-Schnittstelle die nahtlose Einbindung in verschiedene Lernszenarien, wie beispielsweise Formative und Self-Assessments oder Blended Learning.

OPALADIN erweitert das fachliche Anwendungsgebiet von ALADIN durch Aufgabentypen aus der Chemie und der Geschäftsprozessmodellierung erstens in die Breite und zweitens in die Tiefe, indem für bestehende Aufgabentypen, wie dem Lösen von SQL-Aufgaben, fachsemantisch sinnvolle Aufgaben generiert werden können.

OPALADIN bereichert ALADIN um Graph-Grammatiken und Computer Algebra Systeme, um auch die Aufgaben- und Lösungsgenerierung deklarativ, modular und wiederverwendbar zu gestalten.

OPALADIN untersucht wie die Implementation der LTI-Schnittstelle in LMS, insb. in OPAL, ausgebaut und verbessert werden kann, beispielsweise um die eine dynamische Übergabe von toolspezifischen Parametern zu ermöglichen und die Rückgabewerte des Tools zur dynamischen Kurssteuerung zu verwenden, um Lernpfade individuell auf den Lernenden abzustimmen. Um das Spektrum an Aufgabentypen zu erweitern soll ALADIN zukünftig um ein Autorentool ergänzt werden, welches eine visuelle Modellierung von Aufgabentypen und Aufgabengeneratoren erlaubt.

OPALADIN: ALADIN goes OPAL

Dieser Beitrag präsentiert OPALADIN, eine Erweiterung von ALADIN (Generator für Aufgaben und Lösungs(shilf)en in der Informatik und angrenzenden Disziplinen), die ALADIN nahtlos in das Lernmanagementsystem OPAL integriert. ALADIN bietet ein Framework zur deklarativen Modellierung von Aufgabentypen, automatischen Generierung von Aufgaben und Lösungen, interaktiven Bearbeitung von personalisierten Übungen sowie asynchronem Austausch und Nachverfolgung von Lösungsversuchen. Die Motivation für OPALADIN ergibt sich aus dem Mangel an unbekannten Übungsaufgaben, fehlender Skalierbarkeit des Schwierigkeitsgrades, mangelnder Flexibilität in Bezug auf Ort und Zeit des Lernens sowie dem hohen Arbeitsaufwand für Lehrkräfte bei der Erstellung und Bewertung von Aufgaben. OPALADIN zielt darauf ab, diese Herausforderungen anzugehen, indem beliebig viele Aufgaben generiert werden können, die Parametrisierung und Individualisierung der Aufgabengenerierung ermöglicht wird und eine fachliche, zeitliche und institutionelle Flexibilität gegeben ist. Darüber hinaus integriert sich OPALADIN nahtlos in OPAL durch die Authentifizierung über LTI v1.3, die Übermittlung von ALADIN-spezifischen Parametern zur Aufgabengenerierung, das Aufzeichnen und Nachverfolgen von Lösungsversuchen sowie die Übertragung und Verwaltung von Noten. Die Funktionalität von OPALADIN umfasst das 4R-Prinzip für den asynchronen Austausch, die deklarative Erstellung von Aufgabentypen, die Wiederverwendbarkeit von Aufgabenkomponenten (Benutzeroberfläche und Generatorelemente) sowie Funktionen wie Gamification und Spaced Repetition. Darüber hinaus werden semantisch plausible SQL-Aufgaben generiert, indem semantische Bezeichner für Tabellen und Fremdschlüsselbeziehungen der IMDB-Datenbank in einem Wissensgraphen annotiert werden, der Datenbankgraph durchlaufen wird, um Tabellen auszuwählen und zufällig SQL-Abfragebestandteile zu generieren, und schließlich die SQL-Abfrage in natürliche Sprache übersetzt wird. Das Poster schließt mit Vorschlägen für zukünftige Erweiterungen, einschließlich dynamischer Eingabevariablen und Ausgabevariablen zur Steuerung von Kursen, einer verbesserten Benutzeroberfläche für das Notenbuch und der Erweiterung von ALADIN um ein visuelles Autorentool zur vereinfachten Erstellung von Aufgabentypen in nicht-informatischen Bereichen sowie einen virtuellen KI-gestützten Tutor, der Wissensgraphen generiert, Lernerdaten sammelt, anonymisiert und analysiert.

**OPALADIN: ALADIN goes OPAL**

**Einführung**

ALADIN (Generator für **A**ufgaben und **L**ösung-(shilf)en **a**us **d**er

**I**nformatik und angrenzenden Diszipline**n**) ist ein Framework

- zur deklarativen Modellierung von Aufgabentypen,

- zur automatischen Generierung von Aufgaben und

Lösung(shilf)en,

- zur interaktiven Bearbeitung von individualisierten

Übungsaufgaben,

- zum asynchronen Austausch und Nachvollziehen von

Lösungsversuchen.

**Motivation:**

**Problemstellung:**

- Kaum unbekannte Aufgaben zum selbständigen Üben

- Keine Skalierung der Aufgaben hinsichtlich ihres Schwierigkeitsgrads

- Keine orts- und zeitflexible Lehre

- Hoher Aufwand für Lehrkräfte bei der Aufgabenerstellung und -korrektur

- Keine zentraler „Hub“ für Lehre und Lernen

- Annotation semantischer Bezeichner für Tabellen und Fremdschlüsselbeziehungen der IMDB-DB in Wissensgraph

- Traversieren des Datenbankgraphen zur Auswahl der Tabellen und zufällige Generierung von SQL-Abfragebestandteilen

- „Übersetzung“ der SQL-Abfrage in natürliche Sprache

**Zielstellung:**

- Generierung beliebig vieler Aufgaben

- Parametrisierung und Individualisierung der Aufgabengenerierung

- Fachlich, zeitlich und institutionell flexible Nutzbarkeit

- Aufwandsreduktion in der Aufgabenerstellung und -korrektur

- Integration in bestehende Lernmanagementsysteme (LMS) wie OPAL

**LMS-Integration**

- Authentifizierung via LTI v1.3 (OPAL)

- Übermittlung von ALADIN-spezifischen Parametern zur Aufgabengenerierung via „Spezielle Konfiguration“

- Aufzeichnen und Nachvollziehen der Lösungsversuche in ALADIN

- Notenübertragung und -management in OPAL mittels des „Assignment und Grades Service“

**OPALADIN-Funktionalität**

- 4R-Prinzip für asynchronen Austausch

- Deklarative Aufgabentyperstellung

- Wiederverwendbarkeit von Aufgabenbestandteilen (UI- und Generatorenelemente)

- Gamification und Spaced Repetition

**Generierung semantisch plausibler SQL-Aufgaben:**

- Annotation semantischer Bezeichner für Tabellen und Fremdschlüsselbeziehungen der IMDB-DB in Wissensgraph

- Traversieren des Datenbankgraphen zur Auswahl der Tabellen und zufällige Generierung von SQL-Abfragebestandteilen

- „Übersetzung“ der SQL-Abfrage in natürliche Sprache

**Erweiterungsmöglichkeiten**

**OPAL**

− Dynamische Input-Variablen (Expertenregeln?)

− Output-Variablen zur Steuerung des Kurses (analog zu ONYX-Variablen)

− Key-Value-Paare als Response zur Darstellung im Gradebook (evtl. mit ext[Extension]-Präfix)

− Horizontale Scrollbar im Gradebook

− Implementation aller LTI v1.3 Schnittstellenmodule

**ALADIN**

- Visuelles Autorentool zur vereinfachten Aufgabentyperstellung → Verwendung in Nicht-Informatik-Domänen

- Virtueller KI-gestützter Tutor

− (Semi-)Automatische Generierung von Wissensgraphen

− Sammeln, Anonymisieren, Analysieren und Bereitstellen von Lernerdaten

OPALADIN: ALADIN goes OPAL

Dieser Beitrag präsentiert OPALADIN, eine Erweiterung von ALADIN (Generator für Aufgaben und Lösungs(shilf)en in der Informatik und angrenzenden Disziplinen), die ALADIN nahtlos in Lernmanagementsysteme (LMS), wie beispielsweise OPAL integriert. ALADIN bietet ein Framework zur deklarativen Modellierung von Aufgabentypen, automatischen Generierung von Aufgaben und Lösungen, interaktiven Bearbeitung von personalisierten Übungen sowie asynchronem Austausch und Nachverfolgung von Lösungsversuchen. Die Motivation für OPALADIN ergibt sich aus dem Mangel an unbekannten Übungsaufgaben, fehlender Skalierbarkeit des Schwierigkeitsgrades, mangelnder Flexibilität in Bezug auf Ort und Zeit des Lernens sowie dem hohen Arbeitsaufwand für Lehrkräfte bei der Erstellung und Bewertung von Aufgaben. OPALADIN zielt darauf ab, diese Herausforderungen anzugehen, indem beliebig viele Aufgaben generiert werden können, die Parametrisierung und Individualisierung der Aufgabengenerierung ermöglicht wird und eine fachliche, zeitliche und institutionelle Flexibilität gegeben ist. Darüber hinaus integriert sich OPALADIN nahtlos in OPAL durch die Authentifizierung über LTI v1.3, die Übermittlung von ALADIN-spezifischen Parametern zur Aufgabengenerierung, das Aufzeichnen und Nachverfolgen von Lösungsversuchen sowie die Übertragung und Verwaltung von Noten. Die Funktionalität von OPALADIN umfasst das 4R-Prinzip für den asynchronen Austausch, die deklarative Erstellung von Aufgabentypen, die Wiederverwendbarkeit von Aufgabenkomponenten (Benutzeroberfläche und Generatorelemente) sowie Funktionen wie Gamification und Spaced Repetition. Darüber hinaus werden semantisch plausible SQL-Aufgaben generiert, indem semantische Bezeichner für Tabellen und Fremdschlüsselbeziehungen der IMDB-Datenbank in einem Wissensgraphen annotiert werden, der Datenbankgraph durchlaufen wird, um Tabellen auszuwählen und zufällig SQL-Abfragebestandteile zu generieren, und schließlich die SQL-Abfrage in natürliche Sprache übersetzt wird. Das Poster schließt mit Vorschlägen für zukünftige Erweiterungen, einschließlich dynamischer Eingabevariablen und Ausgabevariablen zur Steuerung von Kursen, einer verbesserten Benutzeroberfläche für das Notenbuch und der Erweiterung von ALADIN um ein visuelles Autorentool zur vereinfachten Erstellung von Aufgabentypen in nicht-informatischen Bereichen sowie einen virtuellen KI-gestützten Tutor, der Wissensgraphen generiert, Lernerdaten sammelt, anonymisiert und analysiert.