

# ALADIN II – GENERATOR FÜR AUFGABEN UND LÖSUNG(SHILF)EN AUS DER INFORMATIK UND ANGRENZENDEN DISZIPLINEN II

Torsten Munkelt

HTW Dresden  
torsten.munkelt@htw-dres-  
den.de

Paul Christ

HTW Dresden  
paul.christ@htw-dresden.de

## Zusammenfassung

Der Beitrag stellt das Framework ALADIN ("Generator für Aufgaben und Lösung(shilf)en aus der Informatik und angrenzenden Disziplinen") und seine Erweiterung, ALADIN II, vor. ALADIN ist ein Web-basiertes Framework, in welchem Lehrende graphenbasierte Aufgabentypen deklarieren können und welches zufallsbasiert und entsprechend einer Parametrisierung den Studierenden Aufgaben in beliebiger Anzahl generiert. Die Studierenden lösen die generierten Aufgaben interaktiv, wobei ALADIN ihnen bei Bedarf abgestufte automatische Lösungshilfen leistet. ALADIN II erlaubt das selbstgesteuerte, orts- und zeitunabhängige E-Learning, gibt Rückmeldungen bezüglich des Fortschritts und kann um Aufgabentypen erweitert werden. Bisher nutzen noch nicht alle Studierenden ALADIN, was der fehlenden didaktischen Integration in die Lehrveranstaltungen und der geringen Durchdringung der Lehrveranstaltungen mit den bereits existierenden Aufgabentypen geschuldet ist. Die didaktischen Herausforderungen die der Entwicklung von ALADIN zu Grunde liegen und in der Nutzung von ALADIN observiert wurden und wie ALADIN II diesen entgegentritt wird ebenfalls beleuchtet. So ergänzt ALADIN II das System um neue Aufgabentypen und eine didaktische Integration in die Lehrveranstaltungen: ALADIN II verzahnt durch die Lehrenden gesteuerte Präsenzlehre und durch die Studierenden gesteuertes E-Learning.

## 1 Aufbau und Funktionsweise von ALADIN

ALADIN ist ein System welches Studierenden kompetenzorientiertes E-Learning ermöglicht und wurde mit dem Ziel entwickelt computergenerierte Aufgaben mitsamt Lösungen und Lösungshilfen zu generieren. ALADIN erlaubt es Studierenden selbständig Aufgaben zeit- und ortsunabhängig zu lösen. Der Entwicklung von ALADIN lag die Beobachtung zugrunde, dass sich viele Aufgaben der Informatik und angrenzender Disziplinen sowie ihre Lösungswege und Lösungen als Graphen und Algorithmen über Graphen auffassen lassen. Algorithmen und Softwarebibliotheken zum Erzeugen, Speichern, Traversieren, Modifizieren und Analysieren von Graphen sind allgemein bekannt bzw. liegen vor (Michail et al., 2020). Die bei der Entwicklung von ALADIN verfolgten Anforderungen waren, dass

- graphenbasierte Aufgaben automatisch und zufallsbasiert erzeugt werden,
- dabei je nach Vorkenntnissen oder Wünschen umfangreiche und weniger umfangreiche Aufgaben generiert werden,
- diese Aufgaben online zeit- und ortsunabhängig bereitgestellt werden,
- das System beim Lösen der Aufgaben Unterstützung in Form von Hinweisen oder schrittweisen Teillösungen liefert.

**Kommentiert [PC1]:** Mehr oder weniger 1 zu 1 aus MoHoL-Beitrag kopiert

Abwandlung nötig?

## 2 Didaktische Herausforderungen

### 2.1 Motivation der Studierenden zur Nutzung von ALADIN

#### 2.1.1 Initiale Motivation der Studierenden zur Nutzung von ALADIN

Eine zentrale Herausforderung ist die Motivation der Studierenden zur Nutzung von ALADIN. Dabei kann zwischen der initialen und der weiterführenden Nutzung unterschieden werden. Um die Nutzer initial mit dem System vertraut zu machen, sind in der Regel sowohl Schulungen im Umgang mit der Software, als auch ein Support-System für nachträglich anfallende Fragen nötig, was mit hohem Aufwand verbunden ist (Chi et al., 2014).

#### 2.1.2 Fortführende Motivation der Studierenden zur Nutzung von ALADIN

Unabhängig von der bevorzugten Lernstrategie der Studierenden müssen die Studierenden Aufgaben meist wiederholt bearbeiten um das benötigte Kompetenzlevel für das Bestehen einer Prüfung zu erreichen. ALADIN unterstützt dabei durch seine dynamische Generierung von Aufgaben und Lösungshilfen. In der Praxis ist jedoch eine zusätzliche Motivation der Studierenden erforderlich, damit diese ALADIN fortgehend als unterstützendes Werkzeug in ihrem Lernprozess einsetzen.

### 2.2 Fehlende Rückmeldung gegenüber Lehrenden

#### 2.2.1 Generelle Messbarkeit von Kompetenzen einer Person

Die Messbarkeit der Kompetenz einer Person bezüglich eines Konzepts erfordert zumindest ein iteratives verbales Abtasten dieses Konzepts. Häufig ist jedoch die Betrachtung der Bearbeitung einer zu dieser Kompetenz validen Aufgabe erforderlich (Längrich et al., 2013), da sich die tatsächliche Kompetenz nicht aus der kommunizierten Schlussfolgerung selbst, sondern aus der Abfolge an Schritten welche zu der Schlussfolgerung führen, ergibt und nicht weiter reduzierbar ist.

**Kommentiert [PC2]:**  
Etwas verschwurbelt formuliert?

Evtl. eher:  
Zur Messung einer Kompetenz einer Person genügt es nicht, auf die Selbsteinschätzung der Person zu vertrauen. Erst durch die Bearbeitung einer Aufgabe, welche die Kompetenz erfordert, erschließt sich die Kompetenz der Person (Längrich et al., 2013). Häufig lässt sich die Messung der Kompetenz dabei nicht auf einen Ergebnisabgleich reduzieren, sondern erfordert das Nachvollziehen des kompletten Lösungswegs.

Vielleicht hast du ja noch eine bessere Formulierung :-/

#### 2.2.2 Skalierbare Messbarkeit von Kompetenzen in heterogenen Gruppen

Wohingegen in der Präsenzlehre zumindest ein grober Überblick über die Kompetenz von Studierenden hinsichtlich einer Aufgabe erhalten werden kann, wird dies durch die fehlende persönliche Rückkopplung in ALADIN zunächst verhindert.

Eine zusätzliche Erfassung der Interaktionen zur Lösung der Aufgabe aller Studierenden unterliegt dabei jedoch ähnlichen Verzerrungseffekten wie dem direkten Austausch in der Präsenzlehre. So weisen Studierende, welche engagiert an den Vorlesungen teilnehmen, zum einen eine höhere Kompetenz und zum anderen eine erhöhte Meldebereitschaft auf. Analog dazu ist zu erwarten, dass engagierte Studierende ALADIN vergleichsweise stärker nutzen, welche davon im Vergleich zu unengagierten Studierenden weniger profitieren, in der Bearbeitung der Aufgaben aber besser abschneiden. Die Heterogenität der Gruppe und die Ungleichgewichtung der erfassten Ergebnisse im Vergleich zur zugrunde liegenden Verteilung der Studierenden, erschwert das Erfassen repräsentativer Statistiken. Eine Auswertung dieser Daten erfordert daher stets auch die Möglichkeit für facettierte Visualisierungen anstatt reiner Aggregationen, da diese möglicherweise zu falschen Rückschlüssen über die tatsächliche Verteilung führen können (Matejka & Fitzmaurice, 2017).

Diese Herausforderungen resultieren zwar nicht aus der Nutzung eines elektronischen Systems, werden jedoch durch die verminderte Interaktion mit Lehrkräften verstärkt sichtbar gemacht. Elektronische Systeme erlauben jedoch die effiziente Betrachtung von vielen einzelnen Nutzerlösungen in automatisch aufbereiteter Form, um dem grundsätzlichen Problem überhaupt erst skalierbar entgegenzutreten zu können.

### 2.3 Fehlende Vernetzung der Studierenden untereinander

Im Gegensatz zur Präsenzlehre sind Studierende bei der selbständigen Bearbeitung von Aufgaben in ALADIN nicht unmittelbar dazu gehalten, sich miteinander zu vernetzen. Die fehlende Vernetzung von Studierenden kann in Extremfällen zur Überforderung der Studierenden, Verlängerungen des Studiums und zu erhöhten Abbruchquoten führen, wie in der Corona-Pandemie beobachtet (Kris-Stephen et al., 2021).

### 2.4 Fehlende Rückmeldung an Studierende, wenn Aufgaben nicht, falsch oder unvollständig gelöst

Unzureichende oder fehlende Rückmeldung gegenüber den Studierenden über den Status oder die Fehler innerhalb der Lösungsversuche in ALADIN kann zu einer Minderung der Motivation der Studierenden führen. ALADIN generiert zwar zusätzliche Lösungshilfen zu den Aufgaben, in einigen Fällen können diese jedoch nicht die persönliche Hilfestellung ersetzen, um das *Warum* eines Fehlers ausreichend zu erklären.

### 2.5 Fehlende Variabilität der Aufgabenrepräsentation

#### 2.5.1 Zu hoher Abstraktionsgrad der generierten Aufgaben / fehlende fachliche Semantik der generierten Aufgaben

Der Abstraktionsgrad, in dem eine Aufgabe gestellt werden sollte, um den größten Lernerfolg zu ermöglichen, ist umstritten (Trninic et al., 2020). Eine graduelle Abstufung von konkreten Beispielen hin zu abstrakteren Formulierungen hingegen

**Kommentiert [PC3]:** Bezieht sich auf 2.2.1 und 2.2.2

Evtl. auf anderen Abschnitt verschieben oder einfach weglassen?

**Kommentiert [PC4]:** Bin mir bezüglich der Überschrift immer noch unschlüssig.

Ich verlasse mich da auf dein Urteil :-)

begünstigt den Lernerfolg der Studierenden, sofern diese Methodik auf die Domäne der Aufgabe anwendbar ist (Kokkonen & Schalk, 2020). Zudem erfordern gewisse Aufgabengebiete die Übertragung auf konkrete Szenarien, wie beispielsweise die Geschäftsprozessmodellierung oder die Abfrage von Datenbanken, um eine Kompetenzmessung, welche über die bloße syntaktische Vertrautheit mit der Aufgabe hinausgeht, zu erlauben.

### 2.5.2 Eindimensionale Repräsentation von Konzepten eines Aufgabentyps

Zur Maximierung des potentiellen Lerneffekts durch das Lösen einer Aufgabe, eignen sich unterschiedliche Repräsentationen der verwendeten Konzepte, die zur Aufgabenbewältigung benötigt werden (Lampinen & McClelland, 2018). Beispielsweise kann eine Stücklistenauflösung in der Materialbedarfsplanung sowohl als eine Reihe von Interaktionen auf einem Graph aufgefasst werden, als auch als eine Reihe von Matrixoperationen auf der Adjazenzmatrix des Graphen und des Bedarfsvektors.

### 2.6 Unzureichende semantische Plausibilität und Typikalität der generierten Aufgaben

Eine automatische Generierung spezifischer Aufgaben erfordert das zu Grunde legen eines Metamodells, aus dem der Kontext der Aufgabe abgeleitet werden kann. Je nach Qualität des Metamodells und der Anwendbarkeit von kontextuellen Beschränkungen kann es jedoch zu Brüchen in der semantischen Plausibilität oder der Typikalität der generierten Aufgabe kommen (Wang et al., 2018). Betrachtet man die Ereignisse „Mensch schluckt Süßigkeit“, „Mensch schluckt Watteball“ und „Mensch schluckt Tisch“ sind zwar Ereignis eins und zwei semantisch plausibel, jedoch verletzt lediglich Ereignis eins nicht die Selektionsrestriktion des Verbs „schluckt“ in Kombination mit dem Subjekt „Mensch“ und weist somit eine ausreichende Typikalität auf.

## 3 Umgang mit didaktischen Herausforderungen in ALADIN II

ALADIN II begegnet Herausforderung 2.1.1 mit mehreren Strategien: a) Die Lehrenden verwenden ALADIN II beim Vorführen der Lösungen und dem gemeinsamen Lösen der Aufgaben in den Vorlesungen, um den Studierenden Berührungsängste zu nehmen. b) Sie setzen ALADIN II zumindest in der Einführungsphase in Präsenzübungen und -praktika ein, um die Studierenden im Umgang mit ALADIN II für die Selbstlernphasen zu trainieren. c) Lehrende können in ALADIN II Ausschreibungen für begleitendes Material, wie Video-Tutorials, zu den Aufgaben anlegen, welches von Studierenden erstellt wird und für das sie Punkte erhalten. Durch die Einführung dieses Gamification-Elements werden Studierende gehalten sich mit der Software und den Aufgaben konzeptionell auseinanderzusetzen und eine Kultur zur Selbsthilfe gefördert (Chi et al., 2014).

Kommentiert [PC5]: Evtl. Ausblick?

## ALADIN II - GENERATOR FÜR AUFGABEN UND LÖSUNG(SHILF)EN AUS DER INFORMATIK UND ANGRENZENDEN DISZIPLINEN II

ALADIN II setzt weitere Gamification-Elemente ein um Herausforderung 2.1.2 zu bewältigen: a) Abgeleitet von Stackoverflow<sup>1</sup> können Hilfesuche eröffnet werden, für deren Lösungen Punkte durch andere Nutzer vergeben werden können. Mit den erworbenen Punkten können „Bounties“ auf Hilfesuche ausgeschrieben werden, um anderen Studierenden einen stärkeren Anreiz zur Hilfestellung zu geben. Mit steigender Punktezahl eines Nutzers werden zusätzliche Privilegien freigeschaltet und „Badges“ für den Rang des Nutzers vergeben. b) ALADIN II verwendet Spaced Repetition Algorithmen (Schimanke et al., 2015), um Studierende zu ermutigen Lernstrategien wie „Bulimielernen“ zu vermeiden. Stattdessen werden die Studierenden gehalten in regelmäßigen Zyklen Aufgaben zu wiederholen, um die aufgebauten Kompetenzen langfristig zu erhalten. ALADIN II incentiviert dies durch eine Kopplung weiterer Gamification-Elemente, wie einer zusätzlichen Punktebelohnung für das Einhalten einer Serie von Aufgaben. Der Einsatz von Spaced Repetition Algorithmen erlaubt zudem eine automatische adaptive Anpassung des Schwierigkeitsgrades der generierten Aufgaben an das Niveau des Studierenden, basierend auf seiner Nutzerhistorie.



**Abb. 1 4R-Prinzip**

Um den Herausforderungen aus den Abschnitten 2.2 bis 2.4 zu begegnen, ermöglicht ALADIN II das Aufzeichnen (Record), Weiterleiten (Redirect), Wiedergeben (Replay) und Wiederaufnehmen (Resume) aller Interaktionen mit einer interaktiven Aufgabe durch einen Studierenden (4R). Das 4R-Prinzip erlaubt Studierenden somit ihre Lösungsversuche aufzuzeichnen um diese zu einem späteren Zeitpunkt fortzuführen, aber auch ihre Interaktionen mit anderen Nutzern zu teilen um Feedback von Menschen einzuholen.

**Kommentiert [PC6]:** Evtl. Ausblick?

**Kommentiert [PC7]:** Graphik bieten keinen Mehrwert gegenüber textueller Beschreibung?

<sup>1</sup> siehe <https://stackoverflow.com/help/whats-reputation>

ALADIN II wertet die, mithilfe der durch das 4R-Prinzip ermöglichten, Aufzeichnung der Lösungsversuche aus und erstellt automatisiert Statistiken und Visualisierungen, wobei diese Einblicke in die Lösungsversuche, sowohl in aggregierter Form, als auch anonymisiert in einzelne Lösungsversuche ermöglicht. ALADIN II verwendet zudem Methoden des Process Mining um Verklemmungen in den Lösungsversuchen der Studierenden sichtbar zu machen (Hakim et al., 2019).

**Kommentiert [PC8]:** Evtl. Ausblick?

## 4 Aufgabentypen und fachliche Einsatzgebiete von ALADIN & ALADIN II

ALADIN umfasst bereits die Aufgabentypen „Stücklistenauflösung“, „SQL Abfragen“, „Interpolationsverfahren zur Georeferenzierung“, und „Finden kürzester Pfade“. ALADIN II ergänzt ALADIN zudem um die Aufgabentypen „Spatial SQL-Abfragen“, „Datenfluss-, ERM- und UML-Modellierung“ und „Terminierung und Netzplantechniken“. Die Aufgabentypen „Stücklistenauflösung“, „SQL Abfragen“ und „Terminierung und Netzplantechniken“ wurden bereits vorgestellt (Christ, Paul et al., 2022).

**Kommentiert [PC9]:** Die zwei wurden schon umgesetzt, bisher aber noch nicht in einem Beitrag vorgestellt.

**Kommentiert [PC10]:** Die zwei tauchen noch im Antrag zu ALADIN II auf

**Kommentiert [PC11]:** Wurde bereits im MoHoL-Paper kurz vorgestellt, ist aber aus dem Antrag für ALADIN II. Trotzdem für ALADIN II benennen (in der Kurzzusammenfassung haben wir die Terminierung ebenfalls schon genannt)?

<Kurze Beschreibungen für Aufgaben einfügen>

**Kommentiert [PC12]:** Welche Aufgaben sollten wir deiner Meinung nach vorstellen?

ALADIN II bedient beispielsweise die Studiengänge Wirtschaftsinformatik, Betriebswirtschaftslehre, Wirtschaftsingenieurwesen und Geoinformatik und eignet sich für den Einsatz in den Modulen Betriebliche Informationssysteme, Grundlagen der Wirtschaftsinformatik, Geschäftsprozessmodellierung, Produktionswirtschaft, Geoinformationssysteme, Geodatenmanagement und Datenbanksysteme.

## 5 Zusammenfassung und Ausblick

ALADIN generiert Übungs- und Prüfungsaufgaben und bietet sie Studenten digital dar, so dass sie die Aufgaben selbstständig, zu beliebiger Zeit, an beliebigem Ort und in passendem Schwierigkeitsgrad lösen können. ALADIN reduziert den Aufwand des Lehrpersonal bezüglich der Erzeugung von Übungsaufgaben, der Korrektur der Lösungen und der Betreuung der Studenten während der Lösung der Aufgaben. ALADIN II ergänzt ALADIN mithilfe des 4R-Prinzip um eine asynchrone Interaktionsmöglichkeit zwischen Studenten und Lehrpersonal und erlaubt eine skalierbare statistische Auswertung aufgezeichneter Lösungsversuche. Durch Einführung verschiedener Gamification-Aspekte erhöht ALADIN II die Motivation der Studierenden das System langfristig zu verwenden und fördert eine Selbsthilfekultur. ALADIN II fügt zudem neue Aufgabentypen hinzu, um das Anwendungsspektrum des Systems zu erweitern.

Weitere zukünftig geplante fachliche Einsatzgebiete umschließen beispielsweise Aufgaben aus der Juristerei, Chemie und der Musiktheorie, wie Prüfmuster und Paragraphennetzwerke zu Rechtsfällen, chemische Strukturformeln von Molekülverbindungen und triadische Transformationen in der neo-Riemannschen Theorie.

## ALADIN II - GENERATOR FÜR AUFGABEN UND LÖSUNG(SHILF)EN AUS DER INFORMATIK UND ANGRENZENDE DISZIPLINEN II

Für Herausforderungen welche nicht von ALADIN II bearbeitet werden sind zukünftig folgende Maßnahmen geplant:

Zur Bewältigung der Herausforderungen 2.5.1 und 2.6 soll die Tauglichkeit großer vortrainierter Sprachmodelle wie GPT-3 (Brown et al., 2020) zur semi-automatischen Erzeugung von semantischen Metamodellen untersucht werden.

ALADIN bietet bereits eine deklarative Konfigurationssprache im JSON-Format um interaktive Aufgaben und die zugrunde liegenden Aufgabengeneratoren zu modellieren und Aufgaben in unterschiedlichen Repräsentationen abzubilden (Christ, Paul et al., 2022). Um diese Funktionalität jedoch für Lehrkräfte praktikabel zu machen und Herausforderung 2.5.2 entgegenzutreten, muss der Umgang mit dieser vereinfacht werden. Zukünftig ist geplant einen Editor zu implementieren, welcher eine grafische Konfiguration der in ALADIN vorhandenen UI-Elemente zur Modellierung einer interaktiven Aufgabe ermöglicht.

**Kommentiert [PC13]:** Kann ALADIN ja schon, trotzdem von ALADIN II sprechen?

**Kommentiert [PC14]:** s.o.

## 6 Literatur

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. <https://papers.nips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>

Chi, T.-Y., Olfman, L., & Lin, F. (2014). Exploring the Feasibility of Conducting Software Training in a Peer Learning Context with the Aid of Student-Produced Screencasts. *2014 47th Hawaii International Conference on System Sciences*, 4946–4955. <https://doi.org/10.1109/HICSS.2014.607>

Christ, Paul, Laue, Ralf, & Munkelt, Torsten. (2022). ALADIN – Generator für Aufgaben und Lösung(shilf)en aus der Informatik und angrenzenden Disziplinen. *Workshop zur Modellierung in der Hochschulbildung (MoHoL 2022) co-located with Modellierung 2022, Hamburg, Germany*.

Hakim, A. R., Hasibuan, M. A., & Andreswari, R. (2019). E-learning process analysis to determining student learning patterns using process mining approach. *Journal of Physics: Conference Series*, 1193, 012020. <https://doi.org/10.1088/1742-6596/1193/1/012020>

Kokkonen, T., & Schalk, L. (2020). *One Instructional Sequence Fits all? A Conceptual Analysis of the Applicability of Concreteness Fading in Mathematics, Physics, Chemistry, and Biology Education*. <https://doi.org/10.1007/s10648-020-09581-7>

Kris-Stephen, B., Kochskämper, D., Lips, A., Schröer, W., & Thomas, S. (2021). *Stu.diCo II – Die Corona Pandemie aus der Perspektive von Studierenden*. <https://doi.org/10.18442/194>

- Lampinen, A. K., & McClelland, J. L. (2018). Different Presentations of a Mathematical Concept Can Support Learning in Complementary Ways. *Undefined*. <https://www.semanticscholar.org/paper/Different-Presentations-of-a-Mathematical-Concept-Lampinen-McClelland/b1c046104990ca01c230a88db712afa29b354f99>
- Längrich, M., Schulze, J., & Ghanbari, S. (2013). Anwendung eines allgemeinen Aufgabenbeschreibungsformates auf die Imperative Programmierung. *grkg Humankybernetik*, 64–76.
- Matejka, J., & Fitzmaurice, G. (2017). *Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing* (S. 1294). <https://doi.org/10.1145/3025453.3025912>
- Michail, D., Kinable, J., Naveh, B., & Sichi, J. V. (2020). JGraphT—A Java Library for Graph Data Structures and Algorithms. *ACM Trans. Math. Softw.* <https://doi.org/10.1145/3381449>
- Schimanke, F., Mertens, R., Hallay, F., Enders, A., & Vornberger, O. (2015). *Using a Spaced-Repetition-Based Mobile Learning Game in Database Lectures*.
- Trninic, D., Kapur, M., & Sinha, T. (2020). The Disappearing “Advantage of Abstract Examples in Learning Math”. *Cognitive Science*, 44. <https://doi.org/10.1111/cogs.12851>
- Wang, S., Durrett, G., & Erk, K. (2018). Modeling Semantic Plausibility by Injecting World Knowledge. *NAACL*. <https://doi.org/10.18653/v1/N18-2049>

**Kommentiert [PC15]:** Fehlende Einträge für unser MoHoL Paper (ISBN, Seitenzahl, etc. noch nicht bekannt)