

Prof. Dr. Torsten Munkelt, Fakultät Informatik/Mathematik,
 Prof. Dr. Torsten Gonschorek, Fakultät Wirtschaftswissenschaften und
 Prof. Dr. Frank Schwarzbach, Fakultät Geoinformation:

Antrag auf Förderung der innovativen Lehrform „ALADIN: Generator für Aufgaben und Lösung(shilf)en aus der Informatik und angrenzenden Disziplinen“

1 Ausgangssituation/Probleme vor der Lehrform ALADIN

Studierenden der Informatik und angrenzender Disziplinen, wie z. B. Ingenieur- und Wirtschaftswissenschaften, stehen derzeit oft **nur wenige Übungsaufgaben** zur Verfügung, die sie oft auch noch während der Übung oder der Vorlesung gemeinsam lösen, so dass **kaum unbekannte Aufgaben zum selbstständigen Üben** verbleiben. Zudem skalieren die verfügbaren Aufgaben nicht hinsichtlich ihres Schwierigkeitsgrades und ihres Umfangs. Die Lehrenden der Informatik und angrenzender Disziplinen stellen oft **keine Musterklausuren zu Prüfungsvorbereitung** bereit, da sie Klausuraufgaben aus einem stark begrenzten Reservoir von Aufgaben auswählen, Klausuraufgaben somit oft wiederverwendet werden und nicht durch Probeklausuren bekanntgemacht werden sollen. Selbst dann, wenn Aufgaben in ausreichendem Umfang existieren, haben die Studenten oft Fragen zum Lösungsweg oder kommen bei der Lösung der Aufgabe nicht weiter, so dass **die Lehrenden persönlich Lösungshilfen geben müssen, was eine Verfügbarkeit der Lehrenden voraussetzt**, ihnen erheblichen Aufwand verursacht und außerhalb der Lehrveranstaltungen nur begrenzt möglich ist. Die soeben geschilderte Situation betrifft z. B. die Module Grundlagen der Wirtschaftsinformatik, Betriebliche Informationssysteme, Geschäftsprozessmodellierung, Datenbanksysteme, Datenbankmanagement, Produktionsmanagement, Produktionswirtschaft, Geodatenbanksysteme und Geoinformationssysteme, die meist sogar Studierende mehrerer Studiengänge aus unterschiedlichen Fakultäten besuchen.

2 Lösungsansatz für die Lehrform ALADIN

Viele Aufgaben der Informatik und angrenzender Disziplinen sowie ihre jeweiligen Lösungswege und Lösungen können als Graphen und Algorithmen über Graphen aufgefasst und modelliert werden. Abbildung 1 zeigt einige der besagten Graphen.



Abbildung 1: v. l. n. r.: Ereignisgesteuerte Prozesskette, Programmablaufplan, Datenflussdiagramm, Gozintograph und physisches Datenmodell als Beispiele für Graphen in der Informatik und angrenzenden Disziplinen

Algorithmen und Frameworks zum Erzeugen, Speichern, Traversieren, Modifizieren und Analysieren von Graphen sind allgemein bekannt bzw. liegen vor, so dass die **technischen Voraussetzungen für ALADIN gegeben** sind. Die Algorithmen und Frameworks müssen aber an die speziellen Graphen und die speziellen Anwendungsfälle der Informatik und angrenzender Disziplinen angepasst werden.

Es wird ein „Generator für Aufgaben und Lösung(shilf)en aus der Informatik und angrenzenden Disziplinen“ (ALADIN) entwickelt, der **Aufgaben automatisch** und zufallsbasiert **erstellt** und sie **den Studierenden digital darbietet**. Man kann den Generator so parametrisieren, dass er **in Abhängigkeit von der Kompetenz und den Wünschen der Studierenden leichte und schwere, umfangreiche und weniger umfangreiche Aufgaben** generiert. Zudem kann der Generator die Aufgaben auch **schrittweise automatisch lösen**, den Studierenden **Hinweise zur Lösung der Aufgaben** geben und somit **bei der Lösung der Aufgaben digital helfen**. Die Studierenden müssen die Übungsaufgaben nicht während der Lehrveranstaltungen, sondern können sie in ihrer Selbststudienzeit (**zu beliebiger Zeit**), auch **ohne Hilfe durch Lehrende** und **an einem beliebigen Ort** und auch während ihrer Prüfungen lösen.

Bei den Aufgabentypen, die ALADIN stellen, lösen und bei denen er Lösungshilfe geben kann, handelt es sich z. B. um die nachfolgenden, wobei viele weitere Aufgabentypen denkbar sind:

- Finden von Fehlern in Datenflussdiagrammen (Informatik),
- Zeichnen von Geschäftsprozessdiagrammen (Wirtschaftsinformatik),
- Stücklistenauflösung (Betriebswirtschaft),
- Vorwärts- und Rückwärtsterminierung (Betriebswirtschaft),
- Abfragen raumbezogener Daten in Geodatenbanksystemen (Geoinformatik) und
- Entwickeln von Workflows zur Verarbeitung raumbezogener Daten (Geoinformatik).

Aladin funktioniert im Groben wie folgt: Zu Beginn generiert ALADIN aufgrund von **a) Vorschlagswerten**, **b) Eingaben der Studierenden** oder **c) ihrer bisherigen Leistungen** zufällig einen zum Aufgabentyp passenden speziellen Graphen, dessen Eigenschaften **a), b) oder c)** entsprechen. Beim Aufgabentyp „**Fehler im Graphen finden**“ modifiziert ALADIN den Graphen im Anschluss so, dass er syntaktische oder semantische Fehler aufweist, die von den Studierenden behoben werden. Da ALADIN den Graphen selbst modifiziert, merkt sich ALADIN die Modifikationen und schlägt den Studierenden inverse Modifikationen als Lösungshilfen vor. ALADIN umfasst auch Graphenalgorithmen zum Finden von syntaktischen oder semantischen Fehlern in Graphen. Diese Algorithmen erlauben es, die Aufgaben schrittweise zu lösen und den Studierenden Lösungshilfen zu geben. Beim Aufgabentyp „**Diagramm zeichnen**“ traversiert ALADIN den Graphen und erzeugt einen zu ihm äquivalenten Text, den die Studierenden wieder in einen Graphen überführen, der mit dem Originalgraphen im Idealfall identisch (isomorph) ist. ALADIN erzeugt Lösung(shilf)en aus dem originalen Graphen oder bietet die zur Texterzeugung inversen Operationen als Lösungshilfen an. Beim Aufgabentyp „**Graph ergänzen**“ entfernt ALADIN (Bezeichnungen von) Knoten oder Kanten des Graphen, welche die Studierenden wieder ergänzen. ALADIN merkt sich die entfernten Bestandteile des Graphen und schlägt sie den Studierenden als Lösungshilfen vor. Beim Aufgabentyp „**Berechnung im Graphen**“ stellt ALADIN den generierten Graphen nur dar, und die Studierenden nehmen Berechnungen am Graphen vor. Die Berechnungen am Graphen nimmt ALADIN mittels bekannter Algorithmen selbst schrittweise vor und kann den Studierenden somit ebenfalls Lösungshilfen geben oder Musterlösungen präsentieren. – Bei ALADIN handelt es sich um eine **Client-Server-Anwendung**. Der Server beherbergt die Anwendungslogik. Die **Web- oder Smartphone-Clients** kommunizieren mit dem Server über eine **REST-Schnittstelle**.

3 Nutzen der Lehrform ALADIN

ALADIN **löst die drei in Kapitel 1 aufgeführten Probleme**: Er stellt digital Übungs- und Prüfungsaufgaben beliebigen Umfangs und beliebiger Komplexität – entsprechend der Wünsche und der Kompetenz

der Studierenden –, und er hilft und unterstützt die Studierenden bei der Lösung der Übungsaufgaben zu beliebiger Uhrzeit und an beliebigem Ort, ohne das Lehrende anwesend sein oder helfen müssen. **ALADIN bietet Aufgaben entsprechend des Niveaus der Studierenden an und motiviert somit Studierende zur Lösung der Aufgaben.** Aufgaben auf zu hohem oder zu niedrigem Niveau demotivieren eher. Eine weitere Motivation besteht darin, dass die Studierenden ALADIN selbst parametrisieren können, z. B. mit lustigen Beschriftungen für die Knoten und Kanten des jeweiligen Graphen. **ALADIN erhöht die Kompetenz der Studierenden** zur Lösung entsprechender Aufgaben, **steigert ihren Studienerfolg, senkt ihre Abbrecherquote und verringert ihre Studiendauer.** ALADIN **entlastet die Lehrenden** vom aufwendigen Stellen von Übungs- und Prüfungsaufgaben und strategisch vom monotonen Korrigieren der Aufgaben und Prüfungen. ALADIN ist **mehrsprachig** und somit auch **für internationale Studierende geeignet** und wird mit leichteren Aufgaben geringeren Umfangs **zur Vorbereitung aufs Studium** eingesetzt. ALADIN ist **Open-Source**, im **WWW** und perspektivisch als **Smartphone-App für alle sächsischen Hochschulen verfügbar** und kann **von ihnen erweitert** werden. ALADIN wird **bezüglich weiterer Disziplinen ausgebaut**, was aber fachliche Expertise aus den jeweiligen Disziplinen erfordert.

4 Einordnung der Lehrform ALADIN in die Ziele der Förderung innovativer Lehrformen

Offensichtlich **vermittelt ALADIN Wissen und Kompetenz**, praxisrelevante Aufgaben selbständig zu lösen, was auch dem klaren Lernziel entspricht. **ALADIN aktiviert**, indem es den Schwierigkeitsgrad der Aufgaben an die aktuelle Kompetenz der Studierenden anpasst, so dass die Studierenden Erfolgserlebnisse haben und motiviert werden. Zudem **aktiviert ALADIN** insofern, als dass sich die Studenten selbst an der Generierung der Aufgaben beteiligen können. **ALADIN betreut**, indem es den Studierenden Lösungshilfen gibt, ggf. Fehler aufzeigt und Lösungen zum Vergleich anbietet. ALADIN gestattet eine **Vermischung von Präsenz- und digitalen Lehrmethoden**, da die Studierenden die Aufgaben sowohl selbstständig (durch ALADIN geführt) als auch in den Lehrveranstaltungen gemeinsam mit der/dem Lehrenden lösen können. ALADIN unterstützt das **selbstgesteuerte Lernen**, da die Studierenden sich bei der Generierung der Aufgaben ihre Ziele selbst setzen, sich durch interessante Eingaben selbst motivieren, ihre Lernstrategien und Lerntaktiken weitgehend frei wählen, Probleme weitgehend selbst überwinden, und den Lernerfolg mit der generierten Musterlösung kontrollieren. Der **Kompetenzaufbau** erfolgt mittels ALADIN **a) inhaltlich, b) räumlich und c) zeitlich flexibel**, da ALADIN **a)** Aufgaben unterschiedlicher Art generiert und weitere Aufgabenarten integrieren kann, **b)** Studierende ALADIN mit einem internetfähigen Gerät an jedem beliebigen Ort und **c)** zu jeder beliebigen Zeit nutzen können, da ALADIN keine Betreuung durch Lehrende erfordert.

5 Finanzplan

Für 6000,00 € wird eine studentische/wissenschaftliche Hilfskraft (SHK/WHK) für ein Jahr sechs/fünf Tage pro Monat (siehe Tabelle 1) angestellt, was zirka drei Monaten reiner Arbeitszeit entspricht.

Tabelle 1: Zwei alternativen Finanzpläne in zwei Alternativen; Wahl von SHK oder WHK abhängig von Verfügbarkeit

Alternativen	Stundensatz	Stunden pro Monat	Laufzeit in Monaten	Kosten
SHK	10,10 €	49,5	12	6.000,00 €
WHK	11,80 €	42,4	12	6.000,00 €

Drei Monate genügen einer SHK/WHK allerdings nicht, ein so komplexes Projekt wie ALADIN in vollem Umfang umzusetzen. Deshalb unterstützen die Antragsteller bei Konzeption, Entwurf oder Implementierung, und es werden in einem Jahr voraussichtlich nur das technische Framework für ALADIN aufgebaut und die Generierung mindestens zweier Aufgabentypen als Proof of Concept entwickelt.