ALADIN II – Generator für Aufgaben und Lösung(shilf)en aus der Informatik und angrenzenden Disziplinen II

|  |  |
| --- | --- |
| Torsten Munkelt  HTW Dresden torsten.munkelt@htw-dresden.de | Paul Christ  HTW Dresden paul.christ@htw-dresden.de |

Zusammenfassung

Der Beitrag stellt das Framework ALADIN (“Generator für **A**ufgaben und **L**ösung(shilf)en **a**us **d**er **I**nformatik und angrenzenden Diszipline**n**”) 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 sich aus der Entwicklung von ALADIN ergeben und während der Nutzung von ALADIN observiert worden sind, und wie ALADIN II diesen entgegentritt, werden ebenfalls beleuchtet. So führt ALADIN II neue Aufgabentypen ein und ergänzt ALADIN u. a. um eine didaktische Integration in die Lehrveranstaltungen: ALADIN II verzahnt durch die Lehrenden gesteuerte Präsenzlehre und durch die Studierenden gesteuertes E-Learning.

* 1. Anwendungsbereich von ALADIN (II)

ALADIN ist ein System welches Studierenden kompetenzorientiertes E-Learning ermöglicht und graphenbasierte Aufgaben mitsamt Lösungen und Lösungshilfen automatisch, zufallsbasiert und entsprechend der Anforderungen der Studierenden generiert. ALADIN erlaubt es Studierenden, selbständig Aufgaben zeit- und ortsunabhängig zu lösen. ALADIN befähigt Lehrende zur deklarativen Erstellung von Aufgabentypen und interaktiven Lösungspfaden. Den Aufbau und die konkrete Funktionsweise von ALADIN legt bereits (Christ, Paul et al., 2022) dar.

Aus der bisherigen Nutzung von ALADIN sind einige didaktische Herausforderungen abgeleitet worden. Der Beitrag erläutert besagte Herausforderungen und legt dar, wie ALADIN II ihnen begegnet.

* 1. Didaktische Herausforderungen bei der Anwendung von ALADIN
     1. Motivation der Studierenden zur Nutzung von ALADIN
        1. Motivation der Studierenden zur initialen Nutzung von ALADIN

Eine zentrale Herausforderung ist die Motivation der Studierenden zur Nutzung von ALADIN. Es kann zwischen der initialen und der fortführenden Nutzung unterschieden werden. Um die Studierenden 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).

* + - 1. Motivation der Studierenden zur fortführenden Nutzung von ALADIN

Unabhängig von der bevorzugten Lernstrategie der Studierenden müssen sie Aufgaben meist wiederholt bearbeiten, um die benötigte Kompetenz zum Lösen der Aufgaben und ggf. für das Bestehen einer entsprechenden Prüfung zu erreichen. ALADIN unterstützt bei der Wiederholung durch seine dynamische Generierung von Aufgaben und Lösungshilfen. In der Praxis ist jedoch eine zusätzliche Motivation der Studierenden erforderlich, damit sie ALADIN fortführend als unterstützendes Werkzeug in ihrem Lernprozess einsetzen.

* + 1. Fehlende Rückmeldungen von Studierenden und ALADIN gegenüber Lehrenden
       1. Generelle Messbarkeit von Kompetenzen einer Person

Zur Messung der 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.

* + - 1. Messbarkeit von Kompetenzen in heterogenen Gruppen

Präsenzlehre vermittelt den Lehrenden zumindest einen groben Eindruck von der Kompetenz der Studierenden hinsichtlich der Lösung einer Aufgabe. ALADIN verhindert diesen Eindruck durch die fehlende persönliche Rückkopplung zunächst. Zusätzlich erfasste Interaktionen aller Studierenden mit ALADIN bei der Lösung der Aufgaben unterliegen dabei jedoch ähnlichen Verzerrungseffekten wie dem direkten Eindruck in der Präsenzlehre. So weisen Studierende, welche engagiert an den Lehrveranstaltungen teilnehmen, eine höhere Kompetenz und erhöhte Meldebereitschaft auf. Analog dazu nutzen engagierte Studierende ALADIN stärker, profitieren im Vergleich zu unengagierten Studierenden weniger davon, schneiden bei der Bearbeitung der Aufgaben aber besser ab. Die Heterogenität der Gruppe und die Ungleichgewichtung der erfassten Ergebnisse im Vergleich zur zugrunde liegenden Verteilung der Studierenden erschweren das Erstellen repräsentativer Statistiken. Eine Auswertung erfasster Interaktionen erfordert daher stets auch die Möglichkeit für facettierte Visualisierungen anstatt reiner Aggregationen, da letztere möglicherweise zu falschen Rückschlüssen über die tatsächliche Verteilung führen (Matejka & Fitzmaurice, 2017).

Die in 2.2.1 und 2.2.2 beschriebenen Herausforderungen resultieren zwar nicht nur aus der Nutzung eines elektronischen Systems, treten allerdings bei ihr durch die verminderte Interaktion mit den Lehrenden stärker hervor. Elektronische Systeme erlauben jedoch die effiziente Betrachtung vieler einzelner Nutzerinteraktionen in automatisch aufbereiteter Form und ermöglichen es so überhaupt erst, Verzerrungseffekten bei der Kompetenzmessung in heterogenen Gruppen entgegenzutreten.

* + 1. Fehlende Vernetzung der Studierenden untereinander

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

* + 1. Fehlende Rückmeldung an Studierende

Unzureichende oder fehlende Rückmeldung an Studierende 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.

* + 1. Fehlende Variabilität der Aufgabenrepräsentation
       1. Zu hoher Abstraktionsgrad und zu geringe fachliche Semantik der generierten Aufgaben

Der Abstraktionsgrad, in der eine Aufgabe gestellt werden sollte, um den größten Lernerfolg zu ermöglichen, ist umstritten (Trninic et al., 2020). Eine graduelle Abstufung des Kontexts und der Beschreibung der Aufgabe von anschaulichen Beispielen zu einer Reduktion auf eine formale Beschreibung hingegen 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 bei der Geschäftsprozessmodellierung oder der Abfrage von Datenbanken, um eine Messung der Kompetenz zu erlauben, welche über die bloße syntaktische Vertrautheit mit der Aufgabe hinausgeht.

* + - 1. Zu wenige unterschiedliche Lösungsmethoden eines Aufgabentyps

Zur Maximierung des potentiellen Lerneffekts durch das Lösen einer Aufgabe eignet sich der Einsatz unterschiedlicher Konzepte, die zur Aufgabenbewältigung möglich sind (Lampinen & McClelland, 2018). Beispielsweise kann eine Stücklistenauflösung in der Materialbedarfsplanung sowohl als eine Reihe von Interaktionen auf einem Graphen aufgefasst werden, als auch als eine Reihe von Matrixoperationen auf der Adjazenzmatrix des Graphen und des Bedarfsvektors (Toll, 2010).

* + 1. Unzureichende semantische Plausibilität und Typikalität der generierten Aufgaben

|  |  |  |
| --- | --- | --- |
| **Aussage (Subjekt - Prädikat – Objekt)** | **Semantisch plausibel** | **Selektionsrestriktion eingehalten (ausreichende Typikalität)** |
| 1. Mensch schluckt Süßigkeit | √ | √ |
| 2. Mensch schluckt Wattebausch | √ | X |
| 3. Mensch schluckt Tisch | X | X |

Tabelle 1 Semantische Plausibilität und Typikalität von Aussagen

Eine automatische Generierung spezifischer Aufgaben erfordert das zu Grunde legen eines Metamodels, aus dem der Kontext der Aufgabe abgeleitet werden kann. Je nach Qualität des Metamodels und der Anwendbarkeit von kontextuellen Beschränkungen kann es zu Brüchen in der semantischen Plausibilität oder der Typikalität der generierten Aufgabe kommen (Wang et al., 2018). Wie in Tabelle 1 ersichtlich, sind die Aussagen eins und zwei zwar beide semantisch plausibel, jedoch verletzt lediglich Aussage eins nicht die Selektionsrestriktion des Verbs „schluckt“ in Kombination mit dem Subjekt „Mensch“ und weist somit eine ausreichende Typikalität auf.



* 1. Umgang mit den 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.

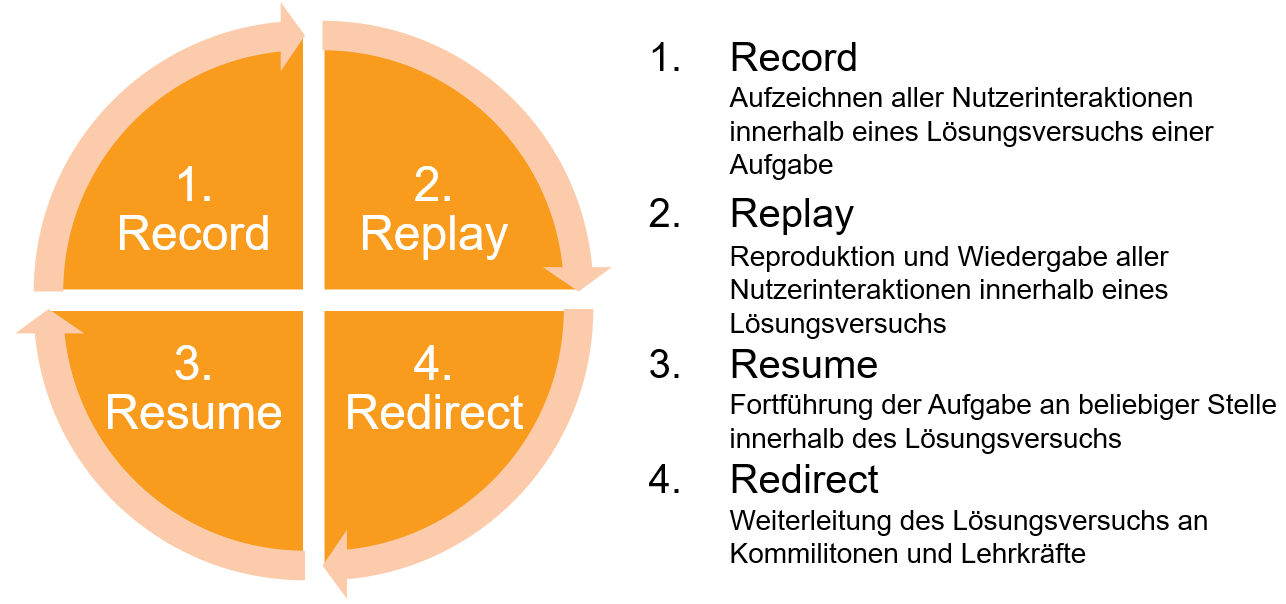
ALADIN II setzt weitere Gamification-Elemente ein um Herausforderung 2.1.2 zu bewältigen: a) Abgeleitet von Stackoverflow[[1]](#footnote-1) können Studierende Hilfegesuche eröffnen, für deren Lösungen Punkte andere Studierende Punkte erhalten. Mit den erworbenen Punkten können „Bounties“ auf Hilfegesuche ausgeschrieben werden, um anderen Studierenden einen stärkeren Anreiz zur Hilfestellung zu geben. Mit steigender Punktezahl der Studierenden werden zusätzliche Privilegien freigeschaltet und „Badges“ für den Rang der Studierenden 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 sichern. ALADIN II incentiviert regelmäßiges Üben durch eine Kopplung weiterer Gamification-Elemente, wie z.  B. einer zusätzlichen Punktebelohnung für das Lösen einer Serie von Aufgaben. Der Einsatz von Spaced-Repetition-Algorithmen erlaubt zudem eine automatische Anpassung des Schwierigkeitsgrades der generierten Aufgaben an das Niveau der Studierenden, basierend auf ihrer Nutzungshistorie.

Abb. 1 4R-Prinzip

Um den Herausforderungen aus den Abschnitten 2.2 bis 2.4 zu begegnen, ermöglicht ALADIN II, mittels dem in Abb. 1 dargestellten 4R-Prinzip, das Aufzeichnen (Record), Weiterleiten (Redirect), Wiedergeben (Replay) und Wiederaufnehmen (Resume) der Lösungsversuche bzw. der Interaktionen der Studierenden mit einer interaktiven Aufgabe. Das 4R-Prinzip erlaubt Studierenden somit, ihre Lösungsversuche aufzuzeichnen, um diese zu einem späteren Zeitpunkt fortzuführen, aber auch, ihre Lösungsversuche an andere Studierende oder an Lehrende weiterzuleiten und sie somit mit ihnen zu teilen, um Feedback von ihnen einzuholen.

ALADIN II wertet die Aufzeichnung der Lösungsversuche aus und erstellt automatisiert Statistiken und Visualisierungen, die Einblicke in die Lösungsversuche sowohl in aggregierter Form als auch anonymisiert in einzelne Lösungsversuche gewähren. ALADIN II verwendet zudem Methoden des Process Mining um Verklemmungen in den Lösungsversuchen der Studierenden sichtbar zu machen (Hakim et al., 2019).

ALADIN II begegnet den, in den Abschnitten 2.5 und 2.6 beschriebenen, Herausforderungen aufgrund deren Komplexität nicht unmittelbar. Mögliche zukünftige Lösungsvorschläge werden jedoch in Kapitel 5 benannt.

* 1. Aufgabentypen und fachliche Einsatzgebiete von ALADIN und ALADIN II

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

Die Terminplanung kann hinsichtlich ihres Zwecks (produktionsorientiert oder projektorientiert), der Terminierungsrichtung (progressiv oder retrograd), und der Darstellungsmethode (Terminliste, Balkenplan, Netzplan, etc.) unterschieden werden (Bielefeld, 2017). Herkömmlich erfordert die Unterstützung dieser unterschiedlichen Repräsentationen die Entwicklung eigenständiger Systeme (Siepermann et al., 2013). ALADIN hingegen erlaubt aufgrund der Entkopplung der Algorithmen zur Generierung der grundlegenden Datenstruktur der Aufgaben von den zugehörigen Darstellungsformen der Aufgabenstruktur und der Lösungsalgorithmen, die Generierung von Aufgaben hinsichtlich der unterschiedlichen Repräsentationen mithilfe eines Systems. Dabei ist neben der Implementation des Generierungsalgorithmus lediglich die deklarative Konfiguration der interaktiven Nutzerbedienelemente zur Modellierung der Darstellungs- und Lösungsmethode erforderlich.

ALADIN II implementiert einen Algorithmus zur Generierung eines zufälligen Graphen, bestehend aus Vorgängen als Knoten mit den Attributen Dauer, Frühester Anfangszeitpunkt (FAZ), Frühester Endzeitpunkt (FEZ), Spätester Endzeitpunkt (SEZ), Spätester Anfangszeitpunkt (SAZ) Gesamtpuffer (GP) und freier Puffer (FP), und Anordnungsbeziehungen als Kanten mit den möglichen Ausprägungen Ende-Anfang (EA), Anfang-Anfang (AA), Anfang-Ende (AE) und Ende-Ende (EE). Der generierte Graph wird anschließend vorwärts- und rückwärtsterminiert um die Pufferzeiten und den eventuell kritischen Pfad zu bestimmen.

Der Algorithmus erlaubt dabei die Parametrisierung bezüglich der möglichen Werteintervalle der Knotenattribute, der möglichen Anordnungsbeziehungen, der Knotenanzahl und des Knotengrades, um zum einen eine Individualisierung der Aufgabenfokussierung und zum anderen eine Anpassung der Aufgabenkomplexität zu ermöglichen.

Weiterhin implementiert ALADIN II die nötigen Bedienelemente zur Modellierung von Aufgaben der unterschiedlichen Darstellungsmethoden der Terminplanung: Die manuelle Konstruktion von Graphen einer definierten Form durch den Nutzer, die manuelle Beschriftung von Knoten und Kanten des konstruierten Graphen und die Manipulation eines Balkendiagramms anhand von Einfüge- und Löschoperationen.

ALADIN II unterstützt im Aufgabentyp „Terminierung und Netzplantechniken“ die Lösungs- und Darstellungsverfahren: Gantt-Diagramm, Critical-Path-Methode (CPM), Metra-Potential-Methode (MPM) und Program Evaluation Research Task (PERT).

<Kurze Beschreibungen für Aufgaben einfügen>

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.

* 1. Zusammenfassung und Ausblick

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

Zukünftig geplante fachliche Einsatzgebiete umfassen beispielsweise Aufgaben aus der Juristerei, der Chemie und der Musiktheorie, wie Prüfmuster und Paragraphennetzwerke zu Rechtsfällen (Burton, 2017), chemische Strukturformeln von Molekülverbindungen (Zheng et al., 2020) und triadische Transformationen in der neo-Riemannschen Theorie (Mamedov, 2019).

Für Herausforderungen, denen ALADIN II noch nicht begegnet (siehe Abschnitte 2.5 und 2.6), sind folgende Maßnahmen geplant:

Zur Bewältigung der Herausforderungen 2.5.1 und 2.6 wird die Tauglichkeit großer vortrainierter Sprachmodelle wie GPT-3 (Brown et al., 2020) zur semi-automatischen Erzeugung semantischer Metamodelle untersucht.

Wie in Kapitel 1 beschrieben, bietet ALADIN bereits eine deklarative Konfigurationssprache im JSON-Format, um interaktive Aufgaben(typen) und die zugrunde liegenden Aufgabengeneratoren zu modellieren und unterschiedliche Konzepte zur Lösung eines Aufgabentyps abzubilden. Um das Erstellen neuer Aufgabentypen für Lehrkräfte zu erleichtern und Herausforderung 2.5.2 entgegenzutreten, muss das Erstellen neuer Aufgabentypen vereinfacht werden. Es ist geplant, einen Editor zu implementieren, welcher eine grafische Konfiguration der in ALADIN vorhandenen UI-Elemente zur Modellierung eines interaktiven Aufgabentyps ermöglicht.

* 1. Literatur

Bielefeld, B. (2017). Basics Terminplanung. In *Basics Terminplanung*. Birkhäuser. https://doi.org/10.1515/9783035612646

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/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Burton, K. J. (2017). *„Think Like a Lawyer“ Using a Legal Reasoning Grid and Criterion-Referenced Assessment Rubric on IRAC (Issue, Rule, Application, Conclusion).* https://doi.org/10.5204/JLD.V10I2.229

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.

Mamedov, N. (2019). Introducing Neo-Riemannian Theory in AP Curriculum through Liszt’s Liebestraum No. 3. *International Journal of Information and Education Technology*, *9*, 594–598. https://doi.org/10.18178/ijiet.2019.9.9.1273

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

Schimanke, F., Mertens, R., Hallay, F., Enders, A., & Vornberger, O. (2015). *Using a Spaced-Repetition-Based Mobile Learning Game in Database Lectures*.

Siepermann, M., Siepermann, C., & Lackes, R. (2013). *Electronic Exercises for the Metra Potential Method*. 435–442. https://www.scitepress.org/Link.aspx?doi=10.5220/0004421504350442

Toll, C. (2010). *Materialbedarfsermittlung mit graphentheoretischen Verfahren* (S. 95–100).

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

Zheng, H., Hu, B., Sun, Q., Cao, J., & Liu, F. (2020). Applying a Chemical Structure Teaching Method in the Pharmaceutical Analysis Curriculum to Improve Student Engagement and Learning. *Journal of Chemical Education*, *97*(2), 421–426. https://doi.org/10.1021/acs.jchemed.9b00551

1. siehe <https://stackoverflow.com/help/whats-reputation> [↑](#footnote-ref-1)