

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 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 Generatoren von interaktiven Aufgaben(-instanzen) eines Aufgabentyps. Den Aufbau und die konkrete Funktionsweise von ALADIN legen 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.

2 Didaktische Herausforderungen bei der Anwendung von ALADIN

2.1 Motivierung der Studierenden zur Nutzung von ALADIN

2.1.1 Motivierung zur initialen Nutzung von ALADIN

Eine zentrale Herausforderung ist die Motivierung der Studierenden zur Nutzung von ALADIN. Es wird zwischen der initialen und der fortführenden Nutzung unterschieden. Um die Studierenden initial mit dem System vertraut zu machen, sind in der Regel Schulungen im Umgang mit der Software und ein Support-System für nachträglich anfallende Fragen nötig, was mit hohem Aufwand verbunden ist (Chi et al., 2014).

2.1.2 Motivierung 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 erlangen. ALADIN unterstützt bei der Wiederholung durch seine dynamische Generierung von Aufgaben und Lösungshilfen. In der Praxis ist jedoch eine zusätzliche Motivierung der Studierenden erforderlich, damit sie ALADIN fortführend als unterstützendes Werkzeug in ihrem Lernprozess einsetzen.

2.2 Motivierung der Lehrenden zur Nutzung von ALADIN

2.2.1 Motivierung zur Nutzung in Lehrveranstaltungen

ALADIN fokussiert auf die Generierung und Darbietung von Aufgaben, die Studierende zum selbstständigen Üben nutzen können. Um ALADIN sinnvoll in Lehrveranstaltungen zu integrieren, ist es nötig, möglichst viele weitere Anwendungsszenarien zu erlauben, wie z. B. ein kollaboratives Lösen von Aufgaben während der Präsenzveranstaltungen oder die Anwendung in Prüfungen und Prüfungsvorleistungen.

2.2.2 Motivierung zur Anwendung in weiteren Disziplinen

Die Erstellung neuer Aufgabentypen in ALADIN erfordert bisher grundlegende Programmierkenntnisse. Da Lehrende aus nicht technischen Disziplinen in der Regel nicht über diese Kenntnisse verfügen, besteht die Gefahr, dass ALADIN lediglich in technischen Disziplinen angewendet wird.

2.3 Messbarkeit von Kompetenzen in ALADIN

2.3.1 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 nicht

auf einen Ergebnisabgleich reduzieren, sondern erfordert das Nachvollziehen des kompletten Lösungswegs.

2.3.2 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 jedoch ähnlichen Verzerrungseffekten wie der direkte 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 und profitieren im Vergleich zu weniger engagierten 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 multivariater Analysen und facettierter Visualisierungen anstatt reiner Aggregationen, da letztere möglicherweise zu falschen Rückschlüssen über die tatsächliche Verteilung führen (Matejka & Fitzmaurice, 2017).

Die in 2.3.1 und 2.3.2 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.

2.4 Fehlende Vernetzung der Studierenden untereinander

Im Vergleich zur Präsenzlehre liegt die Schwelle für die Studierenden, sich bei der selbständigen Bearbeitung von Aufgaben in ALADIN miteinander zu vernetzen, deutlich höher. 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).

2.5 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.

2.6 Fehlende Variabilität der Aufgabenrepräsentation

2.6.1 Zu wenige Lösungsmethoden für Aufgabentypen

Zur Maximierung des potentiellen Lerneffekts durch die Bearbeitung von Aufgaben eignet sich der Einsatz unterschiedlicher Lösungsmethoden, die zur Aufgabenbewältigung möglich sind (Lampinen & McClelland, 2018). Beispielsweise kann eine Stücklistenauflösung sowohl als eine Reihe von Interaktionen auf einem Graphen aufgefasst werden, als auch als eine Reihe von Matrixoperationen auf seiner Adjazenzmatrix und dem Bedarfsvektor (Toll, 2010). Zu wenige Lösungsmethoden verringern den Lerneffekt und erschweren den Zugang für einen Teil der Studierenden, die andere Lösungsmethoden präferieren.

2.6.2 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 erzielen, 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 begünstigt hingegen den Lernerfolg der Studierenden, sofern diese Methodik auf die Domäne der Aufgabe anwendbar ist (Kokkonen & Schalk, 2020).

Zudem erfordern gewisse Aufgabengebiete, wie beispielsweise Geschäftsprozessmodellierung oder Datenbankabfragen, die Übertragung auf konkrete Szenarien mit einer fachlichen Semantik, um eine Messung der Kompetenz zu erlauben, welche über die bloße syntaktische Vertrautheit mit der Aufgabe hinausgeht.

2.7 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 Metamodells, aus dem der Kontext der Aufgabe abgeleitet werden kann. Je nach Qualität des Metamodells 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. Gute Aufgaben müssen nicht nur semantisch plausibel sein, sondern auch eine ausreichende Typikalität aufweisen, was ihre Generierung erschwert.

3 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 Stackoverflow¹ können Studierende Hilfesuche stellen, für deren Lösungen andere Studierende Punkte erhalten. Mit den erworbenen Punkten können „Bounties“ auf Hilfesuche 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.

ALADIN II begegnet Herausforderung 2.2.2 in Teilen, indem es neue Aufgabentypen implementiert und damit seinen Anwendungsbereich vergrößert.

Um den Herausforderungen aus den Abschnitten 2.3 bis 2.5 zu begegnen, ermöglicht ALADIN II, mittels dem 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, und auch, ihre Lösungsversuche an andere Studierende oder an Lehrende weiterzuleiten und sie somit mit ihnen zu teilen, um Feedback von ihnen einzuholen.

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

ALADIN II wertet die Aufzeichnung der Lösungsversuche aus und erstellt automatisch 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 2.2, 2.6 und 2.7 beschriebenen Herausforderungen aufgrund ihrer Komplexität noch nicht unmittelbar oder nur in Teilen. Mögliche ergänzende Lösungsvorschläge unterbreitet Kapitel 5.

4 Aufgabentypen und fachliche Einsatzgebiete von ALADIN und ALADIN II

4.1 Von ALADIN und ALADIN II unterstützte Aufgabentypen

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“ und „SQL Abfragen“ stellen (Christ, Paul et al., 2022) bereits kurz vor. Im Folgenden wird exemplarisch ein weiterer Aufgabentyp „Terminierung und Netzplantechniken“ vorgestellt:

4.2 Exemplarische Vorstellung des Aufgabentyps „Terminierung und Netzplantechniken“

Bei der 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 unterschiedlicher Terminplanungsarten die Entwicklung eigenständiger Systeme (Siepermann et al., 2013). ALADIN hingegen generiert Aufgaben unterschiedlicher Terminplanungsarten mittels nur eines Algorithmus, da es den Algorithmus zur Generierung der den Aufgaben zugrundeliegenden Datenstruktur, die der Terminplanungsart zugehörigen Darstellungsformen und die jeweiligen Terminplanungsalgorithmen voneinander trennt. Neben der Implementation des Generierungsalgorithmus ist lediglich die deklarative Konfiguration der interaktiven grafischen Bedienelemente 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 ggf. die kritischen Pfade zu bestimmen.

Der Algorithmus erlaubt die Parametrisierung der Intervalle möglicher Werte für Knotenattribute, der möglichen Ordnungsbeziehungen zwischen den Knoten, der Knotenanzahl und der Knotengrade, um eine benutzerspezifische Fokussierung auf bestimmte Terminplanungsaspekte und eine Anpassung der Aufgabenkomplexität zu ermöglichen.

Weiterhin implementiert ALADIN II die benötigten Bedienelemente zur Modellierung von Aufgaben der unterschiedlichen Darstellungsformen 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 Löschoptionen.

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

4.3 Fachliche Einsatzgebiete von ALADIN und ALADIN II

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

5 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 der 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

Paraphennetzwerke zu Rechtsfällen (Burton, 2017), chemische Strukturformeln von Molekülverbindungen (Zheng et al., 2020) und triadische Transformationen in der neo-Riemannschen Theorie (Mamedov, 2019).

Um Herausforderung 2.2.1 zu begegnen, ist die Einführung verschiedener interaktiver Bearbeitungsmodi in ALADIN geplant, wie beispielsweise ein Kollaborations- und Prüfungsmodus. Zudem soll ALADIN mittels einer LTI-Schnittstelle an Lehrmanagementsysteme wie OPAL und Moodle angebunden werden, um die Einbindung von ALADIN in die bestehende Lehre zu vereinfachen und ALADIN hochschulübergreifend nutzbar zu machen.

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 Lösungsmethoden zur Lösung eines Aufgabentyps abzubilden (Christ, Paul et al., 2022). Um das Erstellen neuer Aufgabentypen für Lehrkräfte zu erleichtern und den Herausforderung 2.2.2 und 2.6.1 entgegenzutreten, muss das Erstellen neuer Aufgabentypen vereinfacht werden. Dazu ist geplant, einen Editor zu implementieren, welcher eine grafische Konfiguration der in ALADIN vorhandenen UI-Elemente zur Modellierung eines interaktiven Aufgabentyps ermöglicht.

Zur Bewältigung der Herausforderungen 2.6.2 und 2.7 wird die Tauglichkeit großer vortrainierter Sprachmodelle wie GPT-3 (Brown et al., 2020) zur semi-automatischen Erzeugung semantischer Metamodelle und zur Generierung von textuellen Aufgaben- und Problembeschreibungen untersucht.

6 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/1457c0d6bfc4967418bfb8ac142f64a-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. (in Druck). 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.di.Co 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. *Journal of Educational Psychology*, 110(5), 664–682. <https://doi.org/10.1037/edu0000235>

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>