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

Paul Christ[1], Ralf Laue[2], Torsten Munkelt[3]

Abstract: Das Erlernen von Fähigkeiten zur Modellbildung ist eine grundlegende Zielstellung in vielen Studiengängen. Insbesondere in der Informatik und angrenzenden Disziplinen lassen sich viele Modellierungsaufgaben mittels Graphen repräsentieren, was das computergestützte Generieren solcher Graphen und entsprechender Aufgaben und Lösung(shilf)en auf Grundlage bestehender Graphenalgorithmen erlaubt. Dieser Beitrag stellt das Framework ALADIN vor, welches graphenbasierte Modelle und Aufgaben für Probleme aus verschiedenen Fachbereichen generiert und Studenten bei der Lösung der Probleme unterstützt. Die Generierung erfolgt parametrisiert, um dem Anforderungsprofil der Bearbeiter zu entsprechen. ALADIN ermöglicht eine zeit- und ortsunabhängige Bearbeitung von Übungsaufgaben. Zudem prüft ALADIN die Lösungen direkt auf Korrektheit, ohne Lehrpersonal zu binden. Aufzeichnungs- und Wiedergabefunktionalität erhöht den Nutzen von ALADIN in Blended-Learning-Szenarien.

Keywords: E-Learning, Aufgabengenerator, Modellierungsaufgaben, Blended Learning

1 Herausforderungen in der Modellierungslehre

Für Modellierungsaufgaben in der Informatik und angrenzenden Disziplinen stehen oft nur wenige Übungsaufgaben zur Verfügung. Diese werden zudem häufig während der Übung oder Vorlesung gemeinsam gelöst, so dass kaum unbekannte Aufgaben zum selbständigen Üben verbleiben. Da Klausuraufgaben aus einem begrenzten Reservoir von Aufgaben stammen, stellt das Lehrpersonal oft keine Musterklausuren bereit, um die vorhandenen Aufgaben nicht durch Probeklausuren bekanntzumachen. Selbst wenn Aufgaben in ausreichendem Umfange existieren, stellen die Studenten oft Fragen zum Lösungsweg oder benötigen Lösungshilfen, was die Verfügbarkeit des Lehrpersonals voraussetzt, erheblichen Aufwand bereitet und außerhalb der Lehrveranstaltungen nur begrenzt möglich ist. Der geschilderte Sachverhalt wurde u. a. in den Modulen „Grundlagen der Wirtschaftsinformatik“, „Geschäftsprozessmanagement“, „Datenbanksysteme“, „Produktionswirtschaft“ und „Geoinformationssysteme“ beobachtet.

[1] Hochschule für Technik und Wirtschaft Dresden, Fakultät Informatik/Mathematik paul.christ@htw-dresden.de
[2] Westsächsische Hochschule Zwickau, Fachgruppe Informatik ralf.laue@fh-zwickau.de
[3] Hochschule für Technik und Wirtschaft Dresden, Fakultät Informatik/Mathematik, torsten.munkelt@htw-dresden.de

2 Ansatz und Ziele von ALADIN

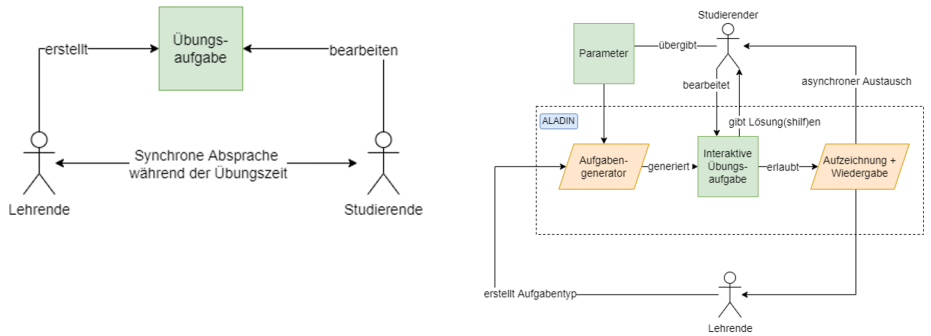
Das in diesem Beitrag vorgestellte Werkzeug ALADIN wurde mit dem Ziel entwickelt, computergenerierte Modellierungsaufgaben mitsamt Lösungen und Lösungshilfen zu generieren. Die Studenten können mittels ALADIN Aufgaben zeit- und ortsunabhängig selbständig lösen. Das Lehrpersonal kann ALADIN auch dazu verwenden, (individuelle) Prüfungsaufgaben zu generieren.

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.

Die bei der Entwicklung von ALADIN verfolgten Anforderungen waren, dass

- graphbasierte Aufgaben automatisch und zufallsbasiert erzeugt werden,
- dabei je nach Vorkenntnissen oder Wünschen umfangreiche und weniger umfangreiche Aufgaben generiert werden,
- diese Aufgaben über Internet zeit- und ortsunabhängig bereitgestellt werden,
- das Werkzeug beim Lösen der Aufgaben Unterstützung in Form von Hinweisen oder schrittweisen Teillösungen liefert,
- Lösungsversuche untereinander ausgetauscht oder den Lehrkräften mit Bitte um Rückmeldungen weitergegeben werden können.

Mit ALADIN als Hilfsmittel kann die in Abschnitt 1 beschriebene Bindung von Lehrkräften reduziert werden, wie Abbildung 1 darstellt.



(a) Szenario ohne ALADIN (b) Szenario mit ALADIN

Abb. 1: Interaktion zwischen Studenten und Lehrpersonal in Übungen

3 Erstellen von Aufgabentypen, Generieren und Lösen von Aufgaben und Leisten von Lösungshilfen in ALADIN

ALADIN verfügt im aktuellen Ausbau über eine Bibliothek von Funktionen, die beim Erstellen neuer Aufgabentypen genutzt werden kann. Die Bibliothek umfasst zum einen gängige Graphalgorithmen. Zum anderen stellt sie generische graphische Bedienelemente bereit, etwa zum Generieren einer Schaltfläche, einer Matrix oder einer Tabelle. Durch Komposition der Funktionen und Bedienelemente können unterschiedliche Aufgabentypen deklarativ beschrieben werden. Da die Beschreibung eines Aufgabentyps die Anordnung der Bedienelemente in Layouts in drei Größen erlaubt, unterstützt ALADIN sowohl Desktopauflösungen als auch Auflösungen mobiler Endgeräte, wie z. B. Tablets oder Smartphones.

Ein neuer Aufgabentyp kann so mittels der vorhandenen Funktionen und Bedienelemente konfiguriert werden. Sollte das Erstellen des neuen Aufgabentyps Elemente erfordern, welche nicht mit bereits vorhandener Funktionalität umsetzbar sind, kann zusätzlich benutzerdefinierter Code ausgeführt werden. So können für die Ausführung aufgabenspezifischer Algorithmen auch externe Programme eingebunden werden, deren Eingaben und zurückgelieferte Ergebnisse einer wohldefinierten Schnittstelle folgen müssen. Die Konfiguration der Logik zur Generierung des Graphen, der Lösungen und der Lösungshilfen sowie der interaktiven Benutzeroberfläche erfolgt in einer JSON-Datei, welche durch ALADIN interpretiert wird. Mit steigender Anzahl an wiederverwendbaren Funktionen in der Bibliothek wird der Bedarf an benutzerdefiniertem Code verringert und die Modellierungsmächtigkeit des Systems erhöht.

Die folgenden Schritte beschreiben die interaktive Bearbeitung einer Aufgabe in ALADIN: Die Studenten wählen einen Aufgabentyp aus und parametrisieren optional die Generierung einer Aufgabe. Auf Grundlage der Parametrisierung erzeugt ALADIN zufallsbasiert eine Aufgabe des gewählten Aufgabentyps. Die Parametrisierung erfolgt manuell durch die Studenten, oder ALADIN nimmt sie aufgrund der Nutzungshistorie automatisch vor, um die Komplexität der generierten Aufgabe an die Bedürfnisse bzw. an den Fähigkeitsstand der Studenten anzupassen. Während der Bearbeitung der Aufgabe können die Studenten Lösungshinweise anfordern. Zudem besteht die Möglichkeit, den Ablauf des Lösungsversuchs aufzuzeichnen und ihn zu einem späteren Zeitpunkt wieder abzuspielen. So aufgezeichnete Lösungsversuche können die Studenten darüber hinaus mit Kommilitonen und dem Lehrpersonal teilen, welche die bisher durchgeführten Interaktionen ebenfalls schrittweise abspielen können. Das Aufzeichnen und Teilen der Lösungsversuche ermöglicht einen asynchronen Austausch mit Anderen und individuelle Hilfestellungen, ohne Lehrpersonal (zeitlich) zu binden. Nach erfolgreicher Bearbeitung der Aufgabe wird der Nutzer entsprechend benachrichtigt.

4 Aufgabenklassen und ihre Beziehung zu Aufgabentypen und Aufgaben in ALADIN

Die Bibliothek von ALADIN erlaubt die Modellierung verschiedener Klassen von Aufgaben, welche im Folgenden vorgestellt werden:

Für Aufgaben der Klasse „Fehler im Graphen finden“ modifiziert ALADIN einen zufallsbasiert generierten Graphen im Anschluss so, dass er syntaktische oder semantische Fehler aufweist. Die Aufgabe besteht danach darin, diese Fehler zu entdecken oder zu beheben. Da ALADIN den Graphen selbst modifiziert, sind die Modifikationen bekannt, sodass ALADIN entsprechende Lösungshilfen geben kann.

Für Aufgaben der Klasse „Berechnungen in Graphen“ generiert ALADIN ebenfalls zufallsbasiert Graphen und nutzt dann vorliegende Graphalgorithmen zum Finden der Lösung. Bei mehrschrittigen Lösungswegen können die Zwischenlösungen gespeichert und auf Anforderung der Studenten als Lösungshilfen zur Verfügung gestellt werden oder ALADIN löst die Aufgabe parallel zu den Studenten.

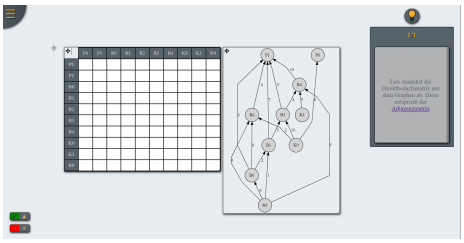
Für Aufgaben der Klasse „Graphen modellieren“ traversiert ALADIN einen zufällig generierten Graphen und erzeugt einen zu ihm äquivalenten Text. Die Aufgabe besteht nun darin, diesen Text zurück in einen Graphen zu überführen, der dem Originalgraphen isomorph ist oder mit ihm in den für die Modellierung wesentlichen Eigenschaften übereinstimmt. ALADIN identifiziert sukzessive Unterschiede zwischen dem Originalgraphen und dem von den Studenten modellierten Graphen und erstellt Lösungshilfen, die auf den Unterschieden basieren.

Für Aufgaben der Klasse „Graphen ergänzen“ entfernt ALADIN Bestandteile des Graphen, Knoten oder Kanten oder deren Bezeichnungen aus einem zufällig generierten Graphen. Die Aufgabe besteht darin, die fehlenden Bestandteile wieder zu ergänzen. Da ALADIN die entfernten Bestandteile und ihre Position im Graphen speichert, kann es auf ihrer Basis Lösungshilfen leisten.

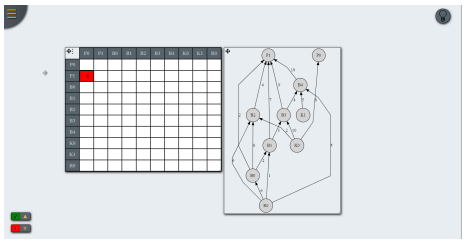
Von einer Aufgabenklasse kann es mehrere Aufgabentypen geben, wie z. B. von der Aufgabenklasse „Graphen modellieren“ die Aufgabentypen „Datenflussdiagramm modellieren“ und „Klassendiagramm modellieren“. Von einem Aufgabentyp kann ALADIN zufallsbasiert beliebig viele konkrete Aufgaben generieren.

5 Ausgewählte Beispiele für in ALADIN realisierte Aufgabentypen

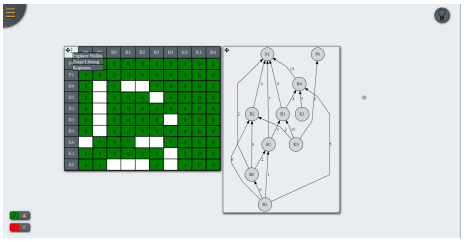
Die in diesem Abschnitt vorgestellten Aufgabentypen stehen u. a. derzeit in ALADIN zur Verfügung.



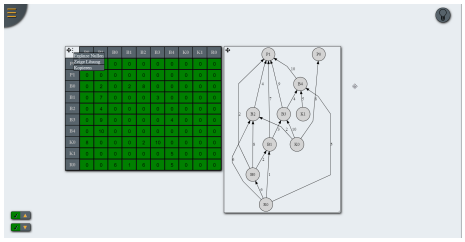
(a) Hinweis zur Aufgabenstellung



(b) Kennzeichnung fehlerhafter Eingaben



(c) Automatische Ergänzung von Null-Werten



(d) Automatisches Lösen

Abb. 2: Lösungshilfen zur Stücklistenauflösung

5.1 Aufgabentyp „Stücklistenauflösung aufgrund von Gozintographen“

Ein Gozintograph stellt den Aufbau von Endprodukten aus Baugruppen, Einzelteilen und Rohstoffen dar. ALADIN unterstützt drei Verfahren zur Stücklistenauflösung. Bei einem überführen die Bearbeiter den Gozintographen G in eine Adjazenzmatrix D (Direktbedarfsmatrix). Alsdann berechnen sie die zu D inverse Matrix I und subtrahieren I von der Einheitsmatrix E, woraus sich die sogenannte Gesamtbedarfsmatrix G ergibt. Zuletzt multiplizieren die Bearbeiter G mit dem Vektor P des ebenfalls generierten Primärbedarfs an Endprodukten und erhalten den Vektor des (Brutto)sekundärbedarfs S an Einzelteilen bzw. Rohstoffen. Beim Üben mit einer zufällig generierten Aufgabe kann ALADIN zu jedem der Schritte Lösungshilfen geben, die Korrektheit von Teillösungen bewerten und korrekte Teillösungen angeben. Abbildung 2 zeigt besagte Lösungshilfen. Der Aufgabentyp „Stücklistenauflösung“ gehört zur Aufgabenklasse „Berechnung in Graphen“, von denen ALADIN auch noch die Aufgabentypen „Geointerpolation von Messwerten“ und „Anwendung des Dijkstra-Algorithmus“ enthält.

5.2 Aufgabentyp „Netzplantechnik“

Ein Netzplan ist ein graphisches Modell zur Abbildung einer Liste an Vorgängen, ihrer jeweiligen Vorgänger und Vorgangsdauern und wird in der Projektplanung verwendet. In ALADIN abgebildete Lösungswege zur Netzplantechnik sind PERT, MPM und CPM sowie

die Darstellung mittels Gantt-Diagramm. Die Bearbeiter modellieren zuerst den Netzplan, einen Graphen, und bestimmen sodann die kritischen Pfade in ihm. Der Aufgabentyp Netzplantechnik ist ein Aufgabentyp der Klasse „Graphen modellieren“.

5.3 Aufgabentyp „SQL-Abfragen“

Die Formulierung von Datenbankabfragen nach dem SQL-Standard ist eine häufige und unverzichtbare Aufgabe in Datenbank-Kursen. Der Aufgabentyp „SQL-Abfragen“ zeigt die Grenzen einfacher computergenerierter Aufgaben auf: Bei Aufgaben zur Stücklistenauflösung und zur Netzplantechnik ist es noch vertretbar, abstrakt von „Produkt A“ oder „Vorgang 3“ zu sprechen, aber bei Datenbankabfragen ist es wünschenswert oder gar erforderlich, in der Aufgabenstellung die Tabellen, ihre Felder und die Fremdschlüsseleinschränkungen (Foreign-Key-Constraints) inhaltlich sinnvoll zu bezeichnen. Schließlich besteht die eigentliche Herausforderung bei Aufgaben zum Erstellen von Datenbankabfragen gerade darin, eine Fragestellung aus der realen Welt („Finde die Hersteller aller verderblichen Produkte, die aktuell auf Lager liegen!“) in eine abstrakte SQL-Abfrage umzuwandeln. Deshalb wurde entschieden, für den Aufgabentyp „SQL-Abfragen“ nicht Datenbanken mit bedeutungsleeren Bezeichnungen zu generieren, sondern mehrere frei zugängliche Datenbanken zu verwenden, die verschiedene Anwendungsbereiche abdecken, wie z. B. Flugbuchung, ERP-System und Arztpraxis. Beim Lösen einer entsprechenden Aufgabe wählen die Bearbeiter nun, für welche dieser Datenbanken sie eine Abfrage erstellen wollen. Die Komplexität der Abfrage kann anhand der Abfragebestandteile (WHERE-Bedingung, HAVING-Klausel, Aggregrationsfunktion etc.) und der Art und der Anzahl der Joins parametrisiert werden. ALADIN generiert und präsentiert die Aufgabe in natürlicher Sprache. Zur Generierung der Aufgabe verwendet ALADIN vordefinierte Textschablonen.

Patient

ID	Name	Vorname	Geburtsdatum	Geschlecht
0	Mustermann	Max	01.01.2000	m
1	Decker	Dirk	31.12.1999	m
2	Räubertochter	Ronja	03.02.1952	w
3	Lustig	Lea	04.05.1965	w

Patientenzustand

ID	PatientenID	Status	Erfassungsdatum
0	0	Genesen	14.04.2020
1	1	Geimpft	01.06.2021
2	2	Geimpft	21.08.2021
3	3	Infiziert	05.12.2020
4	1	Infiziert	01.01.2022

1 n

Abb. 3: Ausschnitt aus relationalen Datenbanktabellen

Für die Datenbank aus Abbildung 3 wäre z. B. eine sinnvolle Abfrage "*Welche Patienten wurden trotz Impfung infiziert?*". ALADIN ist zwar in der Lage die dieser Fragestellung zugrunde liegende SQL-Abfrage zu generieren, jedoch ist die Formulierung der Abfrage in natürlicher Sprache aufgrund der Anwendung der Textschablonen noch zu umfangreich und orientiert sich stark an der generierten SQL-Abfrage, wie Abbildung 4 verdeutlicht.

ALADIN reduziert den Aufwand für die Korrektur, indem es für bekannte Fehlerklassen [TSV18], gezielt Lösungshilfen leistet. Die Ausdrucksstärke von SQL, welche äquivalente

Bilde die Schnittmenge, welche die korrespondierenden Einträge der beiden Tabellen „Patient“ und „Patientenzustand“ enthält. Gib die Spalten „Name“ und „Vorname“ aus. Es sollen nur Daten ausgegeben werden für die „Zustand“ ‘Infiziert‘ ist und die Ausprägungen von „ID“ in einer Untermenge liegen, für die „Zustand“ ‘Geimpft‘ ist und „Erfassungsdatum“ kleiner ist als „Erfassungsdatum“ der Obermenge.

```
SELECT p.Name, p.Vorname FROM Patient AS p
JOIN Patientenzustand AS pz ON p.ID = pz.PatientenID
WHERE pz.Status = 'Infiziert'
AND pz.PatientenID IN
        (SELECT PatientenID FROM Patientenzustand
        WHERE Status = 'Geimpft'
        AND Erfassungsdatum < pz.Erfassungsdatum);
```

Abb. 4: Zuordnung von SQL-Abfragebestandteilen zu Textschablonen

Abfragen erlaubt, schränkt ALADIN nicht ein, da es die Korrektheit der Lösung anhand der Ergebnismenge der SQL-Abfrage verifiziert. Der Aufgabentyp „SQL-Abfragen“ ist ebenfalls ein Aufgabentyp der Klasse „Graphen modellieren“, wobei nicht das Datenbankschema, sondern der Abstrakte Syntaxbaum (Abstract Syntax Tree, AST) einer Abfrage modelliert wird.

6 Technische Realisierung von ALADIN

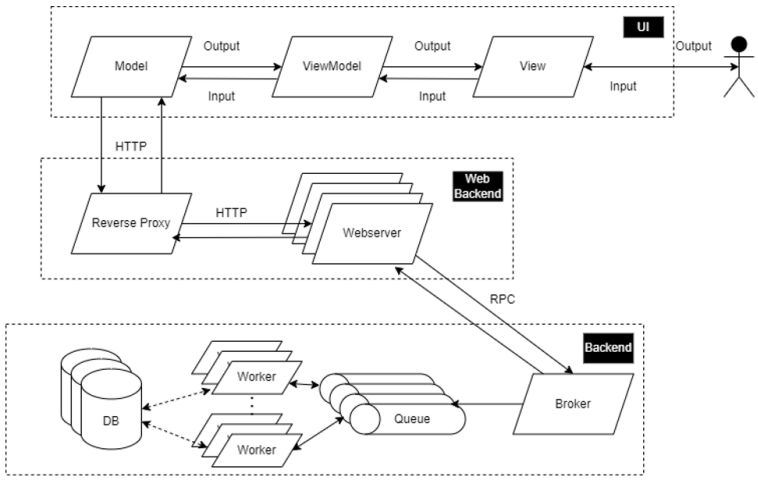


Abb. 5: Technische Realisierung von ALADIN

Bei ALADIN handelt es sich um eine quelloffene Software[4]. Ihre technische Realisierung und das Zusammenspiel ihrer Komponenten stellt Abbildung 5 dar. Das Frontend von ALADIN folgt dem MVVM-Entwurfsmuster (Model View Viewmodel), das die Umsetzung des Event-Sourcing vereinfacht, welches die Aufzeichnung und Wiedergabe von Nutzerinteraktionen, ihre nachträgliche Analyse und die Interpretation der deklarativen Beschreibung der Bedienelemente ermöglicht. Die Kommunikation zum Backend erfolgt über ein API-Gateway, welches die Anfragen an mehrere Webserver verteilt, die mittels RPC (Remote

[4] https://github.com/plc-dev/aladin

Procedure Calls) Events auslösen, die an einen Event-Broker im Backend kommuniziert werden. Das Backend selbst ist als ereignisbasierte Microservice-Architektur realisiert, wobei jeder Aufgabentyp eine eigene Queue darstellt. So kann zur Laufzeit die Anzahl der benötigten Worker erhöht werden, sofern mehr Events in den korrespondierenden Queues auflaufen. Die dargestellte Architektur erlaubt es, dank dynamischer Skalierbarkeit eine große Anzahl nebenläufiger Anfragen zu bearbeiten. Zudem ermöglichst das Publish-Subscribe-Muster das Anstoßen langlaufender asynchroner Prozesse.

7 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 befreit Lehrpersonal vom Stellen von Übungs- und Prüfungsaufgaben, von der Korrektur der Lösungen und der Betreuung der Studenten während der Lösung der Aufgaben. Das Aufzeichnen, Teilen und Abspielen von Lösungsversuchen ermöglicht die asynchrone Interaktion zwischen Studenten und Lehrpersonal. Für die Zukunft ist die Realisierung weiterer Aufgabentypen aus weiteren Fachgebieten geplant.

Eine interessante Herausforderung besteht darin, graphische Modelle und Aufgaben zu ihrer Modellierung zu generieren, bei denen die inhaltliche Bedeutung von Knoten- und Kantenbeschriftungen relevant ist. Bei Aufgaben zu UML-Strukturdiagrammen wird aktuell untersucht, wie Informationen etwa zu den Relationen „ist enthalten in“ oder „ist ein spezieller Fall von“ aus Ontologien oder der WordNet-Datenbank [Fe98] entnommen werden können. Als schwieriger erweist sich das Generieren inhaltlich sinnvoller Verhaltensdiagramme, wie z. B. BPMN-Diagramme. Hier ist vorgesehen, ähnlich wie bei den in Abschnitt 5.3 diskutierten SQL-Datenbanken öffentlich vorhandene Modelle (etwa von [Te18]) zu nutzen.

Literaturverzeichnis

[Fe98] Fellbaum, Christiane, Hrsg. WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press, 1998.

[Te18] Technische Hochschule Mittelhessen: , Referenzmodelle für GP, verfügbar auf http://wi-wiki.de/doku.php?id=funktionsbereiche, 2018.

[TSV18] Taipalus, Toni; Siponen, Mikko T.; Vartiainen, Tero: Errors and Complications in SQL Query Formulation. ACM Trans. Comput. Educ., 18(3):15:1–15:29, 2018.