

Operationalisierung Graph-basierter Lernpfade für adaptive Serious Games

BACHELORARBEIT

KIT – KARLSRUHER INSTITUT FÜR TECHNOLOGIE
FRAUNHOFER IOSB – FRAUNHOFER-INSTITUT FÜR OPTRONIK,
SYSTEMTECHNIK UND BILDAUSWERTUNG

Alexis Delaunay

4. Oktober 2022

Verantwortliche Betreuer:	Prof. Dr.-Ing. Jürgen Beyerer Prof. Dr.-Ing. Thomas Längle
Betreuender Mitarbeiter:	Dipl.-Inf. Alexander Streicher

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe und die Satzung des Karlsruher Instituts für Technologie zur Sicherung guter wissenschaftlicher Praxis in der gültigen Fassung beachtet habe.

Karlsruhe, den 4. Oktober 2022

(Alexis Delaunay)

Inhaltsverzeichnis

Zusammenfassung	1
1 Einleitung	3
1.1 Problemstellung	4
1.2 Lösungsansatz	4
1.3 Zielsetzung	5
1.4 Projektumfeld	5
1.5 Gliederung	6
2 Stand der Forschung und Technik	7
2.1 Adaptive Serious Games (SGs)	7
2.2 Graph-basierte Wissensverfolgung	8
3 Grundlagen	10
3.1 Operationsforschung & Operationalisierung	10
3.1.1 Generisches Verfahren	11
3.1.2 Hypothesen	12
3.1.3 Messung und Skalierung	13
3.2 Serious Games und das Quiz-Spiel „Spot-X“	13
3.3 Adaption & Adaptivität in SGs	16
3.4 Beobachtungsdaten gemäß dem Experience API (xAPI)	18
3.5 User Models in Lernumgebungen	21
3.6 Graphentheorie	23
3.7 Graph-basierte Modellierung von Lernpfaden	26
3.7.1 Nutzungspfade, Lernpfade und ideale Pfade	26
3.7.2 Ideal Paths Models (IPMs)	27
3.8 Adaptive Lernpfade und Performance-Messung	28
3.9 Graph Performance Service (GPS)	31
3.10 Graph-based Knowledge Tracing (GKT)	32

4	Konzept für Graph-basierte Lernpfade in Serious Games (SGs)	36
4.1	Modellierung Graph-basierter Lernpfade für Spot-X	36
4.1.1	Modellierung von Interaktionsgraphen	37
4.1.2	Modellierung der Ideal Paths Models (IPMs)	39
4.2	Performance-Messung in Spot-X	41
4.3	Verallgemeinerung des Konzepts	42
5	Operationalisierung Graph-basierter Lernpfade am Beispiel von Spot-X	43
5.1	Festlegung der Hypothesen	44
5.2	Verwendung der erhobenen Beobachtungsdaten (xAPI)	45
5.3	Vorstellung des Beispielszenarios	48
5.4	Anwendung des Graph Performance Service (GPS)	52
5.5	Anwendung des Graph-based Knowledge Tracing (GKT)	57
6	Technische Verifikation & Diskussion	62
6.1	Anwendung in einem Nutzungsszenario	62
6.2	Bewertung der Hypothesen	65
6.3	Diskussion	66
7	Fazit und Ausblick	67
	Literaturverzeichnis	69
	Abbildungsverzeichnis	75
	Listings	77

Zusammenfassung

Diese Bachelorarbeit untersucht, wie sich Graph-basierte Lernpfade für adaptive digitale Lernspiele operationalisieren lassen. Adaptivität meint hierbei die automatische Anpassung an das Nutzungsverhalten und die jeweiligen Nutzerbedürfnisse, um ein charakterisierendes Ziel zu erreichen. Dies kann im E-Learning zum effektiven, effizienten Voranschreiten zu einem Lernziel verwendet werden. Dazu muss der Fortschritt auf Lernpfaden messbar gemacht werden. In dieser Arbeit wird eine benutzerspezifische Progress Performance (Fortschritt-Performance) auf Graph-basierten Lernpfaden gemessen, um darauf basierend eine adaptive Regelung, in Form von Lernempfehlungen, auszuführen. Hierzu werden unter anderem standardisierte Beobachtungsdaten und Graph-Algorithmen verwendet. Es werden Anwendungsszenarien mit typischen Benutzerprofilen (Personas) am Beispiel des Quiz-Spiel "Spot-X", einem digitalen Lernspiel für die Bildauswertung, angewendet und ausgewertet. Die wissenschaftliche Fragestellung untersucht, wie sich die verschiedenen Nutzungsszenarien in den Graphen widerspiegeln und wie die Performance sowie die (Lern-)Pfade miteinander zusammenhängen. Zusätzlich sollen durch die Analyse der Lernpfade, implizite Abhängigkeiten der Lerneinheiten entdeckt werden. Ein Vergleich mit etablierten Verfahren zum Graph-based Knowledge Tracing (GKT) zeigt den Mehrwert, Gemeinsamkeiten und Unterschiede. Idealerweise kann durch die Operationalisierung gezeigt werden, wie eine indirekte Regelung (Adaptivität) die Lerneffizienz steigern kann.

Abstract

This bachelor thesis examines how graph-based learning paths can be operationalized for adaptive Serious Games. Adaptivity describes the automatic adaption to usage patterns and user needs to reach a characterized goal. This can be used in E-learning for an effective and efficient improvement. Therefore the progress on these learning paths must be measurable. This bachelor thesis measures the user-specific Progress Performance of learning paths to execute an adaptive ruling by learning recommendations. This includes the use of standardized meta data and graph-algorithms. In this relation user scenarios and stereotypes are analyzed on the basis of "Spot-X", a digital Serious Game for Image Analysis. The scientific problem examines how these user scenarios can be reflected in graphs and how the performance is connected to learning paths. Furthermore, an analysis of learning paths should show implicit dependencies of learning sessions. Established Methods, such as Graph-based Knowledge Tracing (GKT), will be compared to show their added value, similarities and differences. In an ideal case the operationalization shows how the learning efficiency can be increased by adaptivity.

1 Einleitung

Um zielorientiertes Lernen auf digitalen Lernplattformen zu ermöglichen, benötigt es eine individuelle Anpassung an den Benutzer (Woolf 2009). Graphen in Kombination mit Graph Analytics können genutzt werden, um Adaptivität mittels sogenannten idealen Lernpfaden zu erreichen (Streicher u. a. 2021). Mithilfe dieser idealen Lernpfade soll die Effektivität und Effizienz des Lernens messbar gemacht werden, um geeignete Lerninhalte zu empfehlen (Baur 2022). Adaptive Serious Games (SGs) bieten hohes Potenzial für eine Vielzahl von Anwendungsbereichen, da sie Motivation, Akzeptanz, aber auch die Identifikation eines Benutzers verbessern können (Dörner u. a. 2016). Das Ziel adaptiver SGs ist es, sich möglichst gut an den Benutzer anzupassen (Dörner u. a. 2016). Für die Adaption existieren unterschiedliche Möglichkeiten, wie beispielweise eine Reduzierung des Zeitbudgets, um das Spiel schwieriger zu gestalten. Graphen bieten zudem die Möglichkeit, eine latente Wissensstruktur der Lernkonzepte herzuleiten, um eine nachvollziehbare Performance-Vorhersage treffen zu können (Nakagawa u. a. 2019). Diese Arbeit konzentriert sich auf die Adaption durch die Wahl geeigneter Lerninhalte. Dabei wird untersucht, wie sich verschiedene Einflussfaktoren aus den Benutzerdaten oder den verwendeten Methoden auf adaptive Lernpfade auswirken. Zur Operationalisierung und Umsetzung adaptiver Lernpfade, wird das Quiz-Spiel "Spot-X" verwendet. Dabei wird das übergeordnete Ziel der Auswahl angemessener Lernempfehlungen für einen Benutzer, wie in Abbildung 1.1 dargestellt, untersucht.

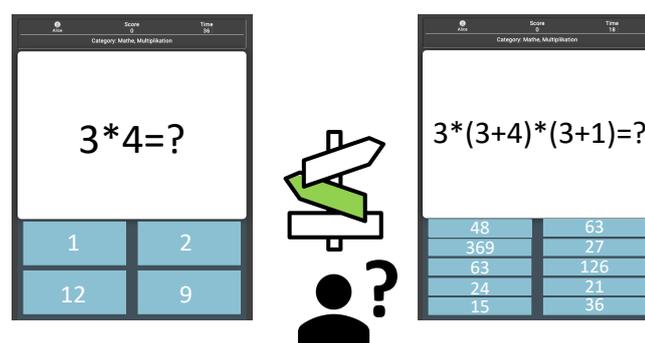


Abbildung 1.1: Unterstützung durch eine benutzerspezifische Adaption in Spot-X

1.1 Problemstellung

Die wissenschaftliche Fragestellung untersucht, wie sich verschiedene Nutzungsszenarien, beispielsweise in Form einer Reihe von Benutzerinteraktionen, in Graphen widerspiegeln. Hierbei soll ebenfalls gezeigt werden, wie die Messung eines Lern-Fortschritts mit adaptiven Lernpfaden zusammenhängt. Die Forschung bietet hierzu bereits mehrere Ansätze, um das Wissen des Nutzers zu verfolgen und Performance-Vorhersagen zu treffen (Nakagawa u. a. 2019; Gajwani und Chakraborty 2021; Alonso-Fernández u. a. 2019). Allerdings werden diese theoretischen Ansätze meist nur genutzt, um eine Verifikation mithilfe veralteter oder synthetischer Metadaten durchzuführen (Nakagawa u. a. 2019), ohne sie in einem konkreten digitalen Lernspiel zu integrieren. Da zahlreiche, unterschiedliche digitale Lernspiele existieren (Dörner u. a. 2016), stellt eine weitere Problematik die Verallgemeinerung auf andere SGs der hier angewandten Methoden dar. Eine Herausforderung spielt hierbei die Festlegung eines Lernziels (Woolf 2009), sowie das Definieren der idealen Lernpfade für unterschiedliche Nutzerklassen (Streicher und Heberle 2017).

1.2 Lösungsansatz

Unter Verwendung standardisierter Beobachtungsdaten (xAPI) sollen Graph-basierte Lösungen zur Wissensverfolgung und Performance-Vorhersage eines individuellen Benutzers angewendet werden, um Lernempfehlungen zu generieren. Hierzu werden vorhandenen theoretischen Konzepte genutzt, um eine Operationalisierung an dem konkreten SG Spot-X durchzuführen. Zur Durchführung der Operationalisierung werden die Graph-basierte Ansätze Graph-based Knowledge Tracing (GKT) von Nakagawa u. a. (2019) und der Graph Performance Service (GPS) (Baur 2022) an Spot-X angewendet. Zunächst wird ein Konzept zur Anwendung Graph-basierter Lernpfade an Spot-X vorbereitet. Dazu müssen Änderungen der theoretischen Konzepte und der Implementierung des GPS vorgenommen werden, um diese für Spot-X anzupassen. Anschließend wird die Operationalisierung an Spot-X unter Verwendung des GPS und GKT vorgenommen. Es werden standardisierte Beobachtungsdaten genutzt, die durch das SG Spot-X gesammelt werden, um darauf basierend die datengetriebene Ansätze zu operationalisieren. Durch Verwendung der Referenzmodelle der Ideal Paths Models (Streicher u. a. 2021) durch Anwendung des GPS (Baur 2022) wird versucht, die effizientesten Lernpfade in Spot-X zu bestimmen. Der Vergleich mit dem etablierten Verfahren des GKT von Nakagawa u. a. (2019) soll in dieser Arbeit ebenso verwendet werden, um eine Performance zu berechnen und Lerninhalte zu empfehlen. Der Vergleich mit dem GPS soll den Mehrwert, Gemeinsamkeiten und Unterschiede aufzeigen.

1.3 Zielsetzung

Diese Arbeit setzt sich das Ziel, den Graph-basierten Ansatz zur Bestimmung von Lernempfehlungen und Messung eines Lern-Fortschritts am Beispiel eines konkreten Serious Games (SG) zu operationalisieren. Dabei sollen zwei Graph-basierte Ansätze zur Operationalisierung angewendet werden und verglichen werden. Zudem soll überprüft werden, ob unter Verwendung der Graphen, die Wissensstrukturen aufweisen können, sinnvolle Empfehlungen für den individuellen Benutzer erzeugt werden können. Aufgrund der Diversität digitaler Lernspiele, sollen neben der Operationalisierung an Spot-X auch eine Verallgemeinerung des Konzepts erarbeitet und mögliche Einschränkungen diskutiert werden. Ebenso soll geprüft werden, wie sich verschiedene Nutzerklassen (z.B. Anfänger und Experte) hinsichtlich ihres idealen Lernpfades unterscheiden. Des Weiteren soll gezeigt werden, welche Arten von Adaptionen stattfinden müssen, um die Lerneffizienz zu steigern. Die erarbeiteten theoretischen Konzepte Graph-basierter Lernpfade sollen anschließend an einem konkreten SG an einem Nutzungsszenario angewendet werden. Ziel ist es durch die hier erarbeitete Operationalisierung, Graph-basierte Lernpfade in einem realen Nutzungskontext und nicht nur in Labortests effektiv anwendbar zu machen. Sobald beide Verfahren für Spot-X angepasst und implementiert sind, werden diese anhand eines Beispielszenarios technisch verifiziert.

1.4 Projektumfeld

Diese Arbeit wird im Rahmen der Abteilung Interoperabilität und Assistenzsysteme (IAS) am Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB) in Karlsruhe geschrieben. Am Standort Karlsruhe werden neue Verfahren der Bildverarbeitung und Produktionstechnik entwickelt und angewendet („Fraunhofer IOSB“ 2022). Die Abteilung IAS befasst sich mit der Forschung von Software-Architekturen für computergestützte Assistenzsysteme („Interoperabilität und Assistenzsysteme (IAS)“ 2022). Dabei werden SGs für den E-Learning Bereich entwickelt. Dazu gehört z. B. das SG “Lost Earth” („Lost Earth 2307“ 2022), mit dem Ziel, Bildauswertungen anhand von realem Bildmaterial zu trainieren. “Lost Earth 2307” ist ein Strategie-Spiel für die Fernerkundung zur Satellitenbildauswertung. Der Spieler lernt mithilfe des Spiels, optische Infrarot- und Radarbilder auszuwerten („Lost Earth 2307“ 2022). Daneben wurde im Rahmen eines studentischen Projekts in Kooperation mit dem Karlsruher Institut für Technologien das Quiz-Spiel “Spot-X” entwickelt. Spot-X wurde im Jahr 2020 erstmals implementiert.

1.5 Gliederung

In diesem Abschnitt wird der Aufbau der Arbeit kurz aufgegliedert. Kapitel 2 stellt den Stand der Forschung und Technik vor. Dabei werden Methoden zur Graph-basierten Wissensverfolgung und die aktuelle Forschung adaptiver SGs aufgezeigt. Kapitel 3 zeigt die theoretischen Grundlagen und Werkzeuge auf, die für die Untersuchung und Operationalisierung Graph-basierter Lernpfade benötigt werden. Unter anderem wird darauf eingegangen, wie mittels gesammelten Benutzerdaten und Anwendung der Graphentheorie eine Fortschritt-Performance gemessen werden kann, um Lernempfehlungen zu wählen. Kapitel 4 bereitet die Modellierung eines Konzepts für die Operationalisierung vor. Das Konzept zeigt, wie Nutzungs- und Lernpfade im SG Spot-X durch einen datengetriebenen Ansatz mittels xAPI-Daten erzeugt werden können. Im darauffolgenden Kapitel 5 wird die Operationalisierung Graph-basierter Lernpfade und die Ergebnisse dokumentiert. Hierbei wird der GPS mit dem GKT verglichen und diskutiert. Kapitel 7 schließt diese Arbeit mit einer Zusammenfassung der erarbeiteten Ergebnisse ab und weist auf mögliche Probleme der Verwendung dieser Methoden hin. Der Ausblick soll zudem mögliche Weiterentwicklungen anreißen.

2 Stand der Forschung und Technik

In diesem Kapitel wird der aktuelle Stand der Forschung und Technik aufgezeigt. Im Fokus liegen hierbei adaptive Serious Games (SGs) und Graph-basierte Methoden im Bereich von E-Learning Plattformen.

2.1 Adaptive Serious Games (SGs)

Laut Dörner u. a. (2016) ist eine wichtige Anforderung adaptiver Spiele, sich möglichst gut an Benutzereigenschaften anzupassen. Ein adaptives System besitzt die Fähigkeit, sich anzupassen oder angepasst zu werden. Ein adaptives System kann zudem die Fähigkeit besitzen, sich automatisch an veränderte Bedingungen anzugleichen. Dabei bezeichnet die Adaption die Handlung, sich einem neuen Zustand oder einer neuen Situation anzupassen. Die Adaption in Spielen bezeichnet eine dauerhafte Anpassung an die Interaktionen des Nutzers und dessen Performance. Dabei wird der aktuelle Zustand im Hinblick auf ein gewünschtes Ziel betrachtet. (Dörner u. a. 2016) SGs bieten durch ihre digitale Gestalt mehrere Adaptionmöglichkeiten (Dörner u. a. 2016). Zur Gewährleistung der Anpassungsfähigkeit existieren vielseitige Möglichkeiten, wie die Anpassung eines eigenen Avatars bis hin zur Auswahl eines geeigneten Spiellevels. Bei der Adaptivität in SGs geht es vor allem um die automatische Anpassung an die jeweilige Situation. Hierbei existieren ebenfalls zahlreiche Möglichkeiten, die Adaptivität zu garantieren, z. B. die automatische Auswahl leichter oder schwieriger Aufgaben oder das Bereitstellen von Hilfestellungen. (Dörner u. a. 2016; Streicher und Smeddinck 2016)

Abbildung 2.1 zeigt beispielhafte Adaptionmöglichkeiten für das SG "Lost Earth 2307" („Lost Earth 2307“ 2022). Das linke Bild zeigt eine Anpassung des Schwierigkeitsgrads durch die Manipulation des zu erratenden Bilds. Rechts wird ein virtueller Assistent abgebildet, der benutzerspezifische Hilfestellungen gibt (Streicher und Smeddinck 2016).

Für übliche Methoden zur Vorhersage einer Performance werden von Dörner u. a. (2016) die Klassifikation (zur Vorhersage einer Kategorie), Regression (zur Vorhersage kontinuierlicher Variablen) und Schätzung eines latenten Wissens (Beurteilung des Wissens oder der Fähigkeiten der Lernenden) genannt.

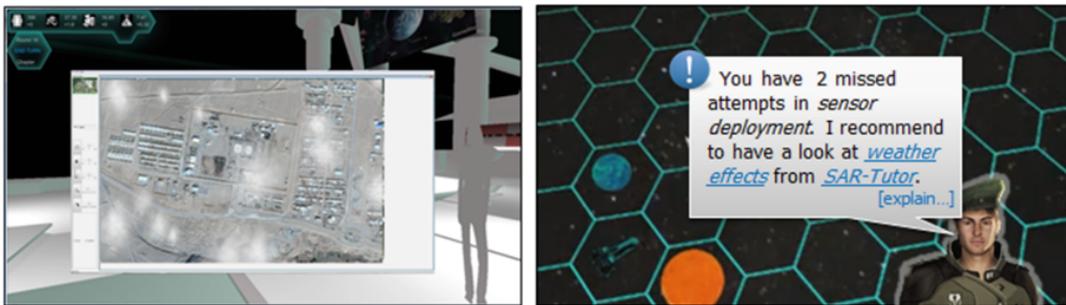


Abbildung 2.1: Beispiele von Adaptivität in dem SG "Lost Earth 2307" (Streicher und Smeddinck 2016)

Ifenthaler und Schumacher (2016) nennen zudem, dass es mithilfe der gesammelten Benutzerdaten möglich ist, "Lernwege zu vergleichen, Lernergebnisse zu analysieren und den Lernfortschritt im Bezug auf Lernziele zu verfolgen". Ebenso, sollen mithilfe von Prognosen, Lernwege optimiert werden, Empfehlungen generiert und schließlich die Erfolgsquote erhöht werden. (Ifenthaler und Schumacher 2016)

2.2 Graph-basierte Wissensverfolgung

Graph-basierte Ansätze können zur Modellierung von Wissen oder von Interaktionen verwendet werden (Huang u. a. 2019). In der Forschung existieren verschiedene Ansätze, die dazu dienen, ein besseres Verständnis über Benutzerdaten und Lernkonzepte zu bekommen (Minn u. a. 2018; Song u. a. 2022; Tong u. a. 2021). Mithilfe eines Graphen ist es beispielsweise möglich,

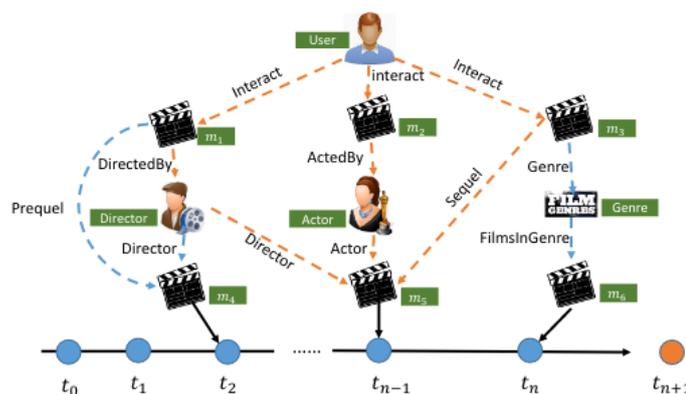


Abbildung 2.2: Beispiel eines Graphen mit Interaktionen zwischen Benutzern und Elementen aus der Filmbranche (Huang u. a. 2019)

Interaktionen und Beziehungen zwischen verschiedenen Aktivitäten darzustellen. Abbildung 2.2 zeigt ein Beispiel eines Graphen, in dem unterschiedliche Interaktionssequenzen abgebildet werden. In diesem Beispiel beschreibt der Graph, dass der Benutzer zum Zeitpunkt $t_n - 1$ den Film m_5 anschaut (Huang u. a. 2019).

Das in dieser Arbeit betrachtete Graph-based Knowledge Tracing (GKT) von Nakagawa u. a. (2019) ist ein etablierter Ansatz aus dem Fachgebiet des Knowledge Tracing (Wissensverfolgung). Das GKT kann genutzt werden, um vorherzusagen, ob ein Benutzer ein Lernkonzept richtig beantworten wird. Dabei soll möglichst genau geschätzt werden, mit welcher Wahrscheinlichkeit dieser Benutzer das Lernkonzept richtig beantworten wird. Es ist ebenfalls möglich, mithilfe des GKT eine Wissensstruktur herzuleiten, die die impliziten Abhängigkeiten von Lernkonzepten darstellt. Dabei werden Wissenskonzepte (engl. knowledge concepts) als Knoten unterteilt und die Abhängigkeiten dieser Wissenskonzepte mit Kanten dargestellt. (Nakagawa u. a. 2019) Als Beispiel solcher Abhängigkeiten kann man sich Lerneinheiten der Potenzrechnung vorstellen, die auf das Rechnen mit Multiplikation aufbauen.

3 Grundlagen

In diesem Kapitel werden die benötigten Grundlagen und fundamentalen Themen für diese Arbeit erläutert, die zur Operationalisierung Graph-basierter Lernpfade benötigt werden. Unter anderem werden hierbei die Themen Operationsforschung, Graphentheorie, Lernpfade, sowie die Adaptivität in Serious Games (SGs) behandelt. Ergänzend werden die verwendeten Werkzeuge und Methoden vorgestellt, darunter der Graph Performance Service (GPS) und das Graph-based Knowledge Tracing (GKT), die an dem SG Spot-X angewendet werden sollen, um Graph-basierte Lernpfade zu operationalisieren.

3.1 Operationsforschung & Operationalisierung

In dieser Arbeit wird die quantitative Methode der Operationalisierung zur Auswertung von Hypothesen im Kontext Graph-basierter Lernpfade angewendet. Im Allgemeinen bezeichnet eine Operationalisierung die Konkretisierung theoretischer Zusammenhänge von notwendigen Variablen in der Praxis (Albers u. a. 2007). Gemäß Heiss und Matthes (2020) werden hierbei konkrete theoretische Konzepte definiert und mithilfe von Indikatoren messbar gemacht. Im Umkehrschluss, kann man dadurch zunächst unmessbare Objekte, in messbare Objekte umwandeln, um so Hypothesen zu bestätigen (Bröder 2011). Die Messung der Zusammenhänge dient der besseren Auswertung der Beobachtungsdaten (Hager 2004).

Eine korrekte Operationalisierung bezeichnen Albers u. a. (2007) als die Übereinstimmung einer Hypothese mit dem tatsächlichen Sachverhalt, die mithilfe einer operativen Messung bestimmt wird. Dadurch können Hypothesen belegt werden und ermöglicht dadurch einen Erkenntniszuwachs. (Albers u. a. 2007)

3.1.1 Generisches Verfahren

Abbildung 3.1 zeigt die vier Schritte von Heiss und Matthes (2020), die zur Operationalisierung benötigt werden.

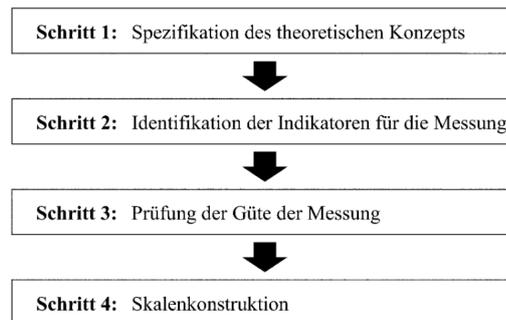


Abbildung 3.1: Allgemeiner Prozess der Operationalisierung (Heiss und Matthes 2020)

Zur Durchführung einer Operationalisierung gibt es in der Literatur zahlreiche Methoden und Anwendungsbeispiele (Kromrey u. a. 2016; Albers u. a. 2007).

Kromrey u. a. (2016) formulieren für die Operationalisierung einen Prozess bestehend aus drei Schritten:

- **Auswahl und Begründung von Indikatoren:** Ist ein Sachverhalt in der Realität nicht direkt messbar, müssen Indikatoren gewählt werden, über die eine Messung erfolgen kann, z. B. die erreichte Punktzahl in einem Quiz. Dabei sollten direkt Korrespondenzregeln formuliert werden, die den Zusammenhang mit dem zu messenden Sachverhalt beschreiben.
- **Angabe und Begründung der Beobachtungsoperationen:** Bezeichnet die Kategorisierung der Messergebnisse im Voraus sowie die Festlegung von Messvorschriften zur Durchführung und Protokollierung der Messung.
- **Formulierung und Begründung von Vorschriften:** Erfolgt zur Zusammenfassung der Ergebnisse einzelner Indikatoren zu einem eindeutigen Messergebnis.

Albers u. a. (2007) stellen zudem weitere Forderungen zur Durchführung einer Operationalisierung. Vor der Erhebung und Analyse der Daten muss zunächst festgelegt werden, welche Aussagen mithilfe dieser Daten gemessen und bestätigt werden sollen. Das Untersuchungsobjekt muss klar definiert sein und die dazugehörige Stichprobe muss dem Untersuchungsobjekt

entsprechen, damit die Generalisierbarkeit der Ergebnisse ermöglicht wird. Um eine korrekte Operationalisierung, also eine valide Messung zu zeigen, muss die operative Messung der Merkmalsausprägung mit den davor festgelegten Hypothesen korrelieren und diese bestätigen. Auch die korrekte Verarbeitung der Daten ist entscheidend. Um die methodische Angemessenheit sicher zu stellen, muss die Verarbeitung der Daten einem zugrundeliegenden Messniveau entsprechen. (Albers u. a. 2007)

3.1.2 Hypothesen

In diesem Kapitel wird die Unterscheidung der drei Arten von Hypothesen näher betrachtet, um in späteren Kapiteln geeignete Hypothesen für die Operationalisierung aufzustellen. Im Folgenden werden Forschungshypothesen, operationale Hypothesen und statistische Hypothesen unterschieden.

Forschungshypothesen können aus Voruntersuchungen, Beobachtungen, Überlegungen oder wissenschaftlichen Theorien abgeleitet werden und handeln von Vermutungen eines Untersuchungsgegenstands (Bortz und Döring 1995).

Statistische Hypothesen helfen dabei, sehr komplexe Hypothesen zu zeigen, welche nur schwer theoretisch belegbar sind (Albers u. a. 2007).

Operationale Hypothesen beschreiben Albers u. a. (2007) als Forschungshypothesen, die auf operationaler Ebene geprüft werden sollen. Dabei wird neben der Hypothese auch festgelegt, wie mit ihr statistisch verfahren werden soll und ab welchem Ergebnis die Hypothese als bestätigt gilt. (Albers u. a. 2007)

Zur Überprüfung der Hypothesen kann die Falsifizierbarkeit mittels einer statistischen Signifikanz geprüft werden. Zur Überprüfung wird die operationale Hypothese in statistische Hypothesen abgeleitet und mit einem Signifikanztest geprüft. Ein Signifikanztest beschreiben Albers u. a. (2007) als die Überprüfung sich gegenseitig ausschließenden Hypothesen. Hierbei unterscheidet man zwischen der Nullhypothese (H_0) und der Alternativhypothese (H_1), die sich gegenseitig ausschließen. Der Signifikanztest legt ein Signifikanzniveau fest, mit dem die Nullhypothese als nicht bestätigt geltend macht. H_0 wird gilt bereits als unbestätigt, sobald sie dem Signifikanzniveau widerspricht. Dadurch kann allerdings noch nicht gezeigt werden, dass H_0 gilt, sondern lediglich, dass die Beobachtungsdaten nicht im Widerspruch zu H_0 stehen (Döring und Bortz 2016; Bamberg und Baur 1998).

3.1.3 Messung und Skalierung

Die Operationalisierung als quantitativer Ansatz erfasst die Variabilität eines Merkmals durch eine spezifizierte Zuordnung von Zahlenwerten und eignet sich daher zum Testen von Hypothesen (Bamberg und Baur 1998). Bei der Messung unterscheidet man zwischen unabhängigen und abhängigen Variablen, die für die Operationalisierung beobachtet und gemessen werden (Hager 2004). Dabei spielen die Merkmale Validität und Reliabilität eine zentrale Rolle. Denn nur wenn Indikatoren diese Merkmale widerspiegeln, können mit ihnen Skalen und Indizes zuverlässig impliziert und komplexe Hypothesen empirisch erfasst werden (Heiss und Matthes 2020). Qualitative Merkmale werden über ihre Messung quantifiziert, also zahlenmäßig bestimmt. Durch die Messbarkeit können die Merkmale dann unterschieden werden. Dabei treten Skalen als eine Art Übersetzung der Merkmale auf (Albers u. a. 2007). Die Validität zeichnet sich durch die Genauigkeit des Messverfahrens aus. Das bedeutet, dass die gemessenen Werte der wahren Realität entsprechen. Dieses Merkmal soll verhindern, dass im Ergebnis zufällige sowie systematische Fehler enthalten sind (Churchill 1979; Balderjahn 2003). Die Reliabilität ist notwendig für die Validität der Messung (Churchill 1979). Die Reliabilität bezeichnet die Zuverlässigkeit der Messung und bemisst die Streuung der Merkmale, also den Grad der wahren Varianz der Testwerte (Moosbrugger und Kelava 2020). Als ein Beispielszenario beschreibt Moosbrugger und Kelava (2020) die Messung der Körpergröße in einer Schneiderei. Statt einem gewöhnlichen Maßband aus dünnem Stahl, verwendet der Schneider ein Gummiband, um die Körpergröße zu messen. Dabei ist es offensichtlich, dass die Verwendung des Gummibands aufgrund der unsystematischen Dehnbarkeit zu uneinheitlichen Maßen führen kann. Dadurch könnten z.B. zu lange Hosen entstehen. Diese fehlerhafte Messung durch die Verwendung des Gummibands kann also zu fehlerhaften Ergebnisse führen. (Moosbrugger und Kelava 2020)

3.2 Serious Games und das Quiz-Spiel „Spot-X“

Ritterfeld u. a. (2009) beschreiben Serious Games (SGs) als jede Form interaktiver und computergestützter Spielesoftware auf einer beliebigen Plattform. SGs gehören zur Untergruppe der digitalen Lernspiele, die neben der Unterhaltung, beim Erreichen eines charakterisierenden Ziels unterstützen sollen (Dörner u. a. 2016). Eines der übergeordneten Ziele von SGs ist die langfristige und extrinsische Motivation eines Spielers, sodass mehr Zeit, Ausdauer und Anstrengung in das Lernen investiert wird (Dörner u. a. 2016). SGs sind in der Lage das Erlernen schwieriger Inhalte zu einer unterhaltsamen, fesselnden Erfahrung zu machen und damit intellektuelle Schwerstarbeit in eine Spielerfahrung verwandeln (Ritterfeld u. a. 2009). Das Anwendungsgebiet von SGs ist breit gefächert und wird vor allem für akademische Zwecke

genutzt, um bspw. beim Auswendiglernen oder Wiedergeben von Fakten zu unterstützen (Ritterfeld u. a. 2009).

Das Quiz-Spiel Spot-X ist ein adaptives SG und wird in dieser Arbeit als Anwendungsbeispiel zur Operationalisierung Graph-basierter Lernpfade betrachtet. Spot-X profitiert durch die Einfachheit und ermöglicht es sich spielerisch Wissen anzueignen. Ein Spieldurchlauf besteht aus mehreren Aufgabendurchläufen. In einem Aufgabendurchlauf muss der Spieler die richtige Antwort innerhalb eines Zeitbudgets auswählen. Abbildung 3.2 zeigt eine beantwortete Multiple-Choice-Aufgabe. In Abbildung 3.2 soll die abgebildete Stadt bestimmt werden.

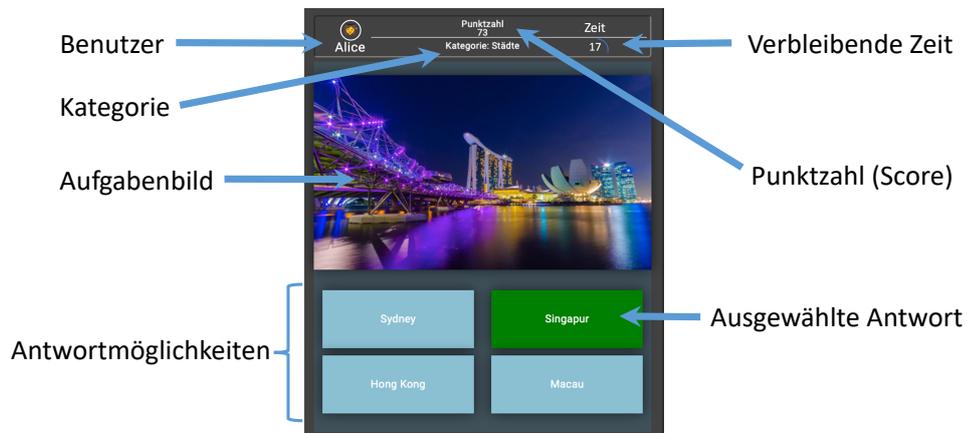


Abbildung 3.2: Eigenschaften und Attribute einer Multiple-Choice-Aufgabe in Spot-X

Tabelle 3.1 zeigt ergänzend zur Abbildung 3.2 die erhältliche Daten, die mit einem Aufgabendurchlauf verbunden sind, wobei A den individuellen Aufgabendurchlauf der Aufgabe "Singapur" bezeichnet:

Attribut	Wert
Benutzer	Alice
Kategorien	{“Städte”}
Aufgabenbild	Bild von Singapur
Antwortmöglichkeiten	{“Sydney”, “Singapur”, “Hong Kong”, “Macau”}
Ausgewählte Antwort	“Singapur”
Benötigte Zeit in Sekunden	$t(A) = \text{Zeitbudget} - 17 = 43$ mit $\text{Zeitbudget} = 60$
Erreichte Punktzahl	$S(A) = 73$ mit $S : A \rightarrow [0, 100]$
Aufgabename	“Singapur”
Schwierigkeitsgrad	$\text{diff}(A) = 0.5$ mit $\text{diff} : A \rightarrow [0,1]$
Schwierigkeitskategorie (SK)	$SK(\text{‘Singapur’}) = \text{mittel}$

Tabelle 3.1: Beispiel erhältlicher Daten eines Spieldurchlaufs in Spot-X

Spot-X besitzt zusätzliche Aufgabentypen wie Wimmelbilder- und Freitext-Aufgaben. In dieser Arbeit wird sich auf Multiple-Choice-Aufgaben beschränkt.

Abbildung 3.3 zeigt eine Aufgabe mit zwei unterschiedlichen Schwierigkeitsgraden. Der Schwierigkeitsgrad $diff \in [0,1]$ dient zur dynamischen Schwierigkeitsanpassung einer Aufgabe, wobei die Aufgabenstellung unverändert bleibt. Dies beeinflusst neben dem Zeitbudget, die Anzahl der Antwortmöglichkeiten. Abbildung 3.3 zeigt die Aufgabe (a) mit dem neutralen Schwierigkeitsgrad $diff = 0.5$, in der keine Adaption stattfindet. Aufgabe (b) besitzt hingegen den maximalen Schwierigkeitsgrad $diff = 1$, die durch die erhöhte Anzahl der Antwortmöglichkeiten zu erkennen ist.

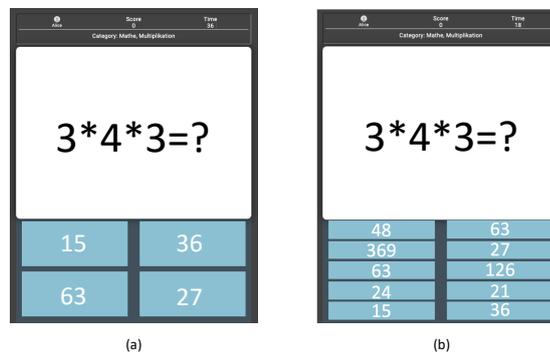


Abbildung 3.3: Multiple-Choice-Aufgabe mit unterschiedlichen Schwierigkeitsgraden

Neben der dynamischen Schwierigkeitsanpassung existiert die sogenannte Schwierigkeitskategorie (SK). Die SK wird klar vom $diff$ unterschieden und dient zur Einstufung der Schwierigkeit bezüglich der Aufgabenstellung. Jede Aufgabe besitzt entweder die SK "leicht", "mittel" oder "schwer". Abbildung 3.4 zeigt zwei Aufgaben, wobei $SK(a) = einfach$ und $SK(b) = schwer$.

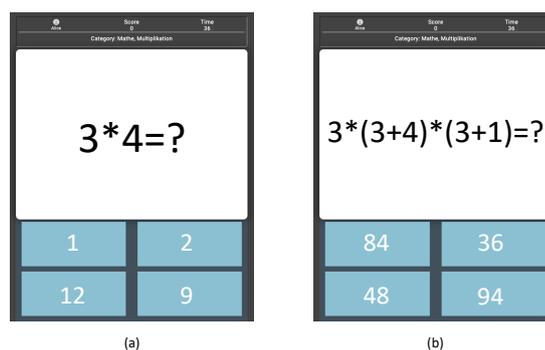


Abbildung 3.4: Zwei Aufgaben mit unterschiedlichen Schwierigkeitskategorien

3.3 Adaption & Adaptivität in SGs

Durch eine adaptive Anpassung soll die Attraktivität, die Motivation und damit auch die Effektivität des Lernens verbessert werden. Ein adaptives Spiel muss sich an Benutzereigenschaften anpassen und an die benötigten Anforderungen, um so ein charakterisierendes Ziel zu erreichen. (Dörner u. a. 2016)

Shute und Zapata-Rivera (2012) stellen einen Adaptivitätszyklus bestehend aus vier Phasen vor. Abbildung 3.5 zeigt die von Streicher und Smeddinck (2016) überarbeitete Darstellung des Adaptivitätszyklus. Dieser zielt darauf ab, die Benutzerbedürfnisse zu erkennen und zu befriedigen. Der Vier-Phasen Adaptivitätszyklus besteht aus der Erfassung (Capture), der Analyse der Beobachtungsdaten (Analyze), der Auswahl, Adaption (Select) und der Präsentation (Present) der Adaption. Nach der Datenanalyse wird das Ergebnis in einem User Model gespeichert. User Models modellieren einen Benutzer und dessen Wissen. Anschließend bestimmt das System, wann und wie die Adaption durchgeführt wird. Abschließend wird die ausgewählte Adaption durchgeführt und dem Benutzer präsentiert, was wiederum zu neuen Beobachtungsdaten führt. (Shute und Zapata-Rivera 2007)

Für adaptive SGs ist die Messung kurzfristiger Auswirkungen der Adaption mithilfe der Benutzerdaten möglich und kann wertvolle Informationen zur Optimierung der benutzerspezifischen Anpassungen liefern (Streicher und Smeddinck 2016).

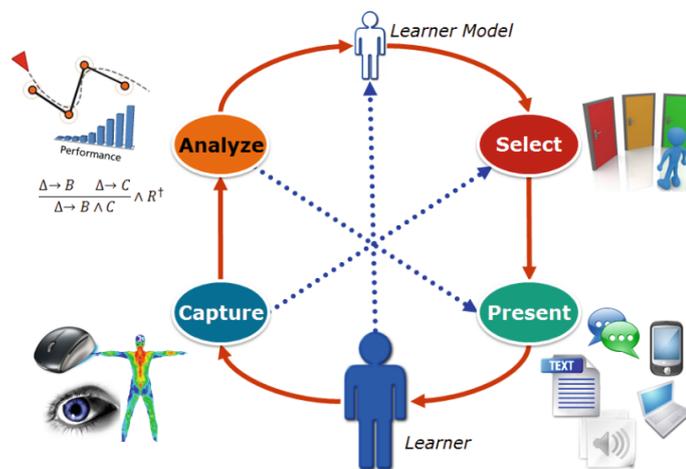


Abbildung 3.5: Vier-Phasen Adaptivitätszyklus (Streicher und Smeddinck 2016, basierend auf Shute und Zapata-Rivera 2012)

Einer der Hauptgründe, neben der Verbesserung der Lern-Effektivität, ist die Erreichung des sogenannten “Flow”-Zustands eines Benutzers (Csikszentmihalyi 2000). Dieser Flow versucht den Spieler zu motivieren und ein Gleichgewicht zwischen der Herausforderung und der Fähigkeiten des Benutzers zu finden (Csikszentmihalyi 2000). Der “Game Flow” (Sweetser und Wyeth 2005) beschreibt eine Spielerfahrung mit einer sehr hoher Konzentration. Der sogenannte “Dual Flow” (Sinclair 2011) zeigt eine Abwandlung des Flows (Csikszentmihalyi 2000). Abbildung 3.6 zeigt den Versuch, die Attraktivität und die Effektivität in Balance zu halten (Dörner u. a. 2016).

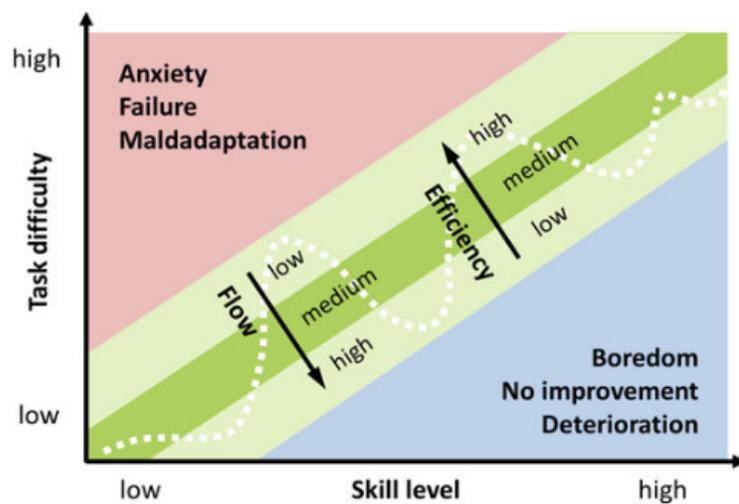


Abbildung 3.6: Das Konzept “Dual Flow” (Abbildung von Dörner u. a. 2016, basierend auf Sinclair 2011)

In vielen Fällen bieten vordefinierte Schwierigkeitsstufen, zur manuellen Anpassung von einfach bis schwer, nicht genügend Flexibilität (Streicher und Smeddinck 2016). Um diese Flexibilität zu erhöhen, wird die Adaption eingesetzt. Man unterscheidet dabei zwischen Mikro- und Makro-Adaption (Kickmeier-Rust u. a. 2011). Die Makro-Adaption betrachtet die Gesamtheit aller möglichen Pfade. Damit werden dem Benutzer geeignete Lernpfade empfohlen oder diese so modifiziert, dass diese optimal an den Benutzer angepasst sind (Streicher u. a. 2021). Die Mikro-Adaption hingegen betrachtet einzelne Elemente innerhalb der Pfade mithilfe der Knoten. So wirkt sich die Adaption unmittelbar auf die Empfehlung für den Benutzer aus (Streicher u. a. 2021). In dieser Arbeit werden adaptive Lernpfade unter Betrachtung mehrere möglichen Pfade betrachtet, um so während eines Spieldurchlaufs Mikro-Adaptionen durchzuführen.

3.4 Beobachtungsdaten gemäß dem Experience API (xAPI)

Zur Durchführung der Operationalisierung werden in dieser Arbeit Daten verwendet, die dem Standard Experience Application Programming Interface („xAPI“ 2022) entsprechen. Eine zentrale Herausforderung bei der Personalisierung und Adaption von Serious Games (SGs) ist die Interoperabilität der Daten (Streicher und Smeddinck 2016), um den Austausch von Lernerfahrungen zwischen mehreren Plattformen zu ermöglichen. Die digitale Form von SGs ermöglicht das Beobachten und Analysieren von Benutzerdaten (Streicher und Smeddinck 2016). Die Beobachtung und Analyse der benutzerspezifischen Lernprozesse dient als Grundlage für Entscheidungen, um beispielsweise die Gestaltung nächster Lerninhalte zu steuern, um die Lernumgebung zu verbessern (Dörner u. a. 2016). Die xAPI ist eine standardisierte Schnittstelle zum Senden und Empfangen von Daten („xAPI“ 2022). Diese Spezifikation dient zum effektiven und flexiblen Datenaustausch zwischen verschiedenen unabhängigen Systemen, z. B. zwischen Lernmanagementsystemen (LMS) und Spielen (Streicher und Smeddinck 2016). Die xAPI ermöglicht es, Erfahrungen (engl. experiences) eines Benutzers zu sammeln und wird unter anderem in E-Learning-Systemen verwendet (Streicher und Smeddinck 2016). Durch die Standardisierung eines gemeinsamen Formats zum Senden und Empfangen von Daten, wird xAPI zu einem idealen Werkzeug der Interoperabilität und ermöglicht den Austausch von Lernerfahrungen zwischen mehreren unabhängigen Systemen („xAPI“ 2022).

Nach Streicher u. a. (2021) werden typischerweise nicht alle kleinen Ereignisse erfasst, bspw. willkürliche Mausbewegungen. Hierbei sollte die logische Struktur des Systems berücksichtigt werden und Ereignisse, die aus Benutzerinteraktionen entstehen, erfasst werden. Alleine das Filtern der zahlreichen Beobachtungsdaten spiegelt bereits die Entscheidung wider, welche Art von Adaptivität verwendet wird. Würde also eine Adaptivität nur auf der Makroebene stattfinden, sind Ereignisse innerhalb einer Sitzung uninteressant und es wäre möglicherweise nur der Beginn und das Ende der Benutzersitzung erforderlich. (Streicher u. a. 2021)

Zur Beobachtung von Benutzerinteraktionen und Lernerfahrungen, werden in dieser Arbeit sogenannte xAPI-Statements („xAPI“ 2022) verwendet. Die xAPI-Statements geben das Format für bestimmte Lernerfahrungen bzw. Lernaktivitäten vor und folgen dem Triple-Prinzip, bestehend aus einem Akteur, Verb und Objekt. Diese xAPI-Statements dienen mit ihrer Grundstruktur zur Beschreibung von Erfahrungen zwischen einem Akteur und einem Objekt. Eine Erfahrung könnte zum Beispiel wie folgt lauten: „Akteur A hat das Level O abgeschlossen.“ Durch die Verwendung des Triple-Prinzips, wird versucht, möglichst viel Informationsgehalt in der festgelegten Struktur abzubilden und dennoch die Interoperabilität zwischen xAPI-konformen Tools bestmöglich zu nutzen. („xAPI“ 2022)

Diese xAPI-Statements werden in einem Learning Record Store (LRS) gespeichert, in dem zusätzlich die Kommunikationsmethoden zum Senden, Empfangen und Anfordern von Daten spezifiziert werden („xAPI“ 2022).

Listing 3.1 zeigt ein xAPI-Statement, das aus einem LRS mit Spot-X Daten stammen könnte. Aus dem xAPI-Statement ist zu entnehmen, dass ein Akteur bzw. Benutzer namens “Alice” eine Mathe-Aufgabe in Form einer Multiple-Choice-Aufgabe abgeschlossen hat.

```
1  {
2      "actor": {
3          "objectType": "Agent",
4          "mbox": "mailto:alice@user.de",
5          "name": "Alice"
6      },
7      "verb": {
8          "id": "http://adlnet.gov/expapi/verbs/completed",
9          "display": { "de": "Aufgabe abgeschlossen" }
10     },
11     "object": {
12         "id": "https://www.spotx.de/spotx/game/play/66",
13         "objectType": "Activity",
14         "definition": {
15             "type": "http://adlnet.gov/expapi/activities/question",
16             "name": { "de": "3*4*3=?" },
17             "description": { "de": "Spielte die Multiple Choice
18                             Aufgabe '3*4*3=?' und antwortete korrekt." }
19         }
20     }
21 }
```

Listing 3.1: Beispielhaftes xAPI-Statement

Akteure dienen zur Identifikation eines Benutzers. Ein Akteur bzw. ein Benutzer kann unabhängig von der Plattform indentifiziert werden und ist über mehrere Plattformen eindeutig („xAPI“ 2022).

Verben werden mithilfe einer URI aus der Experience API Registry in Verbindung mit einer kurzen Beschreibung definiert. Verben dienen der Beschreibung, was zwischen einem Akteur und dem Objekt passiert ist, z.B. „erfahren“, „teilgenommen“, „versucht“, „abgeschlossen“ usw. („xAPI“ 2022)

Objekte fassen die Kernaussage eines xAPI-Statements zusammen, bei denen es sich meistens um Aktivitäten handelt. Aktivitäten werden durch eine eindeutige URI domänenspezifisch definiert und können optional einen beschreibenden Text enthalten. („xAPI“ 2022)

In Listing 3.1 wurde nicht ersichtlich, zu welchem Zeitpunkt Alice die Aufgabe gespielt hat oder welche Punktzahl erreicht wurde. Solche Informationen können dem xAPI-Statement mithilfe zusätzlicher Felder hinzugefügt werden. Eine Ergänzung des obigen Beispiels wird in Listing 3.2 aufgezeigt.

```
1  {
2    "timestamp": "2022-06-04T23:59:47.47Z",
3    "result": {
4      "success": true,
5      "duration": "PT33S",
6      "response": "36(correct)",
7      "score": {
8        "scaled": 0.75,
9        "raw": 75,
10       "min": 0,
11       "max": 100
12     },
13     "completion": true
14   },
15   "context": {
16     "platform": "http://127.0.0.1:3000/",
17     "extensions": {
18       "https://www.spotx.de/spotx/task/id": 66,
19       "https://www.spotx.de/spotx/task/name": "3*4*3=?",
20       "https://www.spotx.de/spotx/task/type": "MultipleChoice",
21       "https://www.spotx.de/spotx/task/categories": "Mathe",
22       "https://www.spotx.de/spotx/task/answers": "15(wrong);36(
23         correct);27(wrong);63(wrong)",
24       "https://www.spotx.de/spotx/task/difficulty": 0.5,
25       "https://www.spotx.de/spotx/task/difficultyCategory": "
26         schwer"
27     }
28   }
29 }
```

Listing 3.2: Ergänztes xAPI-Statement

Das Ergebnis-Feld (result) dient zur Angabe einer gemessenen Punktzahl. Im Listing 3.2 ist zu erkennen, dass der Akteur eine Aufgabe erfolgreich mit 75 von 100 Punkten abgeschlos-

sen hat und dafür 33 Sekunden benötigte. Zudem kann dabei auch angegeben werden, was der Benutzer geantwortet hat. Das Erweiterungs-Feld kann, wie die Bezeichnung andeutet, zusätzliche Informationen enthalten. Erweiterungen können nicht nur in die Hauptstruktur eines xAPI-Statements hinzugefügt werden, sondern auch als Teil des Aktivität-Objekts oder als Teil des Ergebnis-Felds ergänzt werden. Der Inhalt einer Erweiterung enthält meist anwendungsspezifische Informationen für eine konkrete Domäne. Um eine Kontextinformation hinzuzufügen, muss erneut eine eindeutige URI in Kombination mit dem Wert ergänzt werden. („xAPI“ 2022)

Das Kontext-Feld bietet Platz, um kontextbezogene Informationen hinzuzufügen („xAPI“ 2022). Im Rahmen von Spot-X wird das Kontext-Feld genutzt, um zusätzliche Aufgabeninformationen hinzuzufügen, wie beispielsweise, dass eine bestimmte Aufgabe der Schwierigkeitskategorie “schwer” angehört (siehe Listing 3.2).

3.5 User Models in Lernumgebungen

Das Lernen ist am effektivsten, wenn man dazu motiviert ist, daher ist es umso wichtiger den Benutzer engagiert und aktiv in eine authentische und herausfordernde Lernumgebung zu integrieren (Woolf 2009). Woolf (2009) nimmt hierzu die physischen Lernumgebungen zum Vergleich. In traditionellen Lernumgebungen, wie der Schule, versuchen Lehrer anhand sekundärer Lernmerkmale, wie dem Gesichtsausdruck, der Körpersprache und dem Tonfall des Schülers, das Verständnis gegenüber dem Schüler zu erweitern. Lehrer können dadurch ihre Strategien an den Schüler und dessen Lernbedürfnisse anpassen. Diese Interaktionen zwischen Lehrer und Schüler können auch wichtige Informationen über Ziele, Fähigkeiten, Motivation und Interessen des Schülers liefern. Eine Lernumgebung, die auf den bevorzugten Lernstil des Schülers zugeschnitten ist, erhöht das Interesse und fördert das Lernen. Dadurch können schwache Schüler unterstützt und die Stärken gefördert werden. (Woolf 2009)

Diese Herausforderungen aus der physikalischen Welt, sollen laut Woolf (2009) auch in digitalen Lernumgebungen mithilfe von User Models und Student Models umgesetzt werden. Diese Modelle präsentieren das Wissenskonzepte und können dazu verwendet werden, effiziente Operationen auf diese Konzepte durchzuführen. Die Wissensrepräsentation ist notwendig für die Entwicklung einer intelligenten Lernumgebung. Einer der Hauptgründe der Modellierung eines Schülers (Student Model) bezeichnet Woolf (2009) damit, sicherzustellen, dass die digitale Lernumgebung über grundlegende Kenntnisse über den individuellen Schüler verfügt. So soll effektiv reagiert werden und das Interesse des Schülers und das Lernen fördern. Das maßgeschneiderte Feedback ist entscheidend und fördert das Lernen. (Woolf 2009)

Adaptive Systeme benötigen diese Modelle, um die nächste bestmögliche Aktion bzw. einen

passenden Lerninhalt zu bestimmen (Streicher u. a. 2021). Nachdem die Datenanalyse aus dem Vier-Phasen Adaptivitätszyklus abgeschlossen ist, wird das Ergebnis in einem User Model abgebildet. Dieses Modell dient der Repräsentation des aktuellen Zustands eines lernenden Benutzers, im Raum aller Interaktionen die möglich sind. (Streicher und Smeddinck 2016)

Intelligent Tutoring Systems (ITS) sollen die gleichen Vorteile wie "One-on-One Tutoring" mitnehmen und effektiv die Lernzeit reduzieren (Woolf 2009). Je mehr ein Tutor über einen Schüler weiß, desto genauer ist das Schülermodell (Woolf 2009).

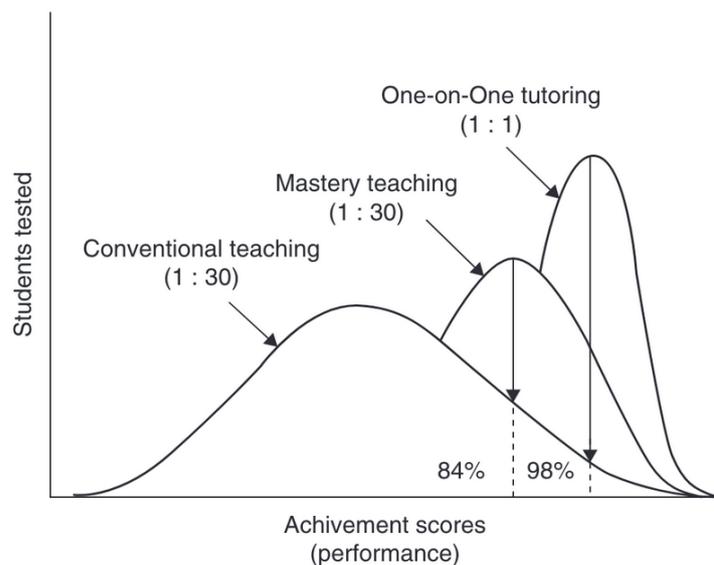


Abbildung 3.7: Statistisch belegte Vorteile von One-on-One Tutoring (Woolf 2009, überarbeitet aus Bloom 1984)

Woolf (2009) beschreibt Benutzermodelle bzw. User Models als Informationsträger, dafür wie, wann und womit ein Benutzer interagiert hat. Diese Modelle helfen dabei den Lernprozess zu individualisieren und gezielt auf die Stärken oder Schwächen des Benutzers einzugehen. Student Models ermöglichen zudem, das aktuelle Wissen und die Lernbedürfnisse darzustellen, zu begründen und darauf zu reagieren. Ein kognitives Benutzermodell enthält zudem Informationen über den kognitiven Zustand oder auch die Belastung (z. B. Stresslevel) eines Benutzers. Die Domänenmodelle können Interaktionspfade darstellen. Jeder lernende Nutzer besitzt ein individuelles Student Modell, das beispielsweise aus Daten des LRS gewonnen werden kann. Student Models enthalten Informationen des Benutzers und die damit enthaltenen Vernetzungen, was wiederum zu mehr Wissen über den Benutzer führt. (Woolf 2009)

Sobald sich ein neuer Benutzer in dem System anmeldet, können Fragebögen oder bereits frühe Beobachtungen der Performance benutzt werden, um das Wissens- oder „Skill“-Level der

Benutzer zu klassifizieren. Mithilfe dieser Klassifizierungen, kann der Nutzer einem vordefinierten Klassenstereotypen, wie „Anfänger“, „Fortgeschrittener“ oder „Experte“ zugeordnet werden (Streicher und Smeddinck 2016). Diese Art und Weise der Personalisierung wird meist auf statische Spiele mit vordefinierten Lernpfaden für die jeweiligen Stereotypen angewendet. Dabei werden keine Anpassung basierend auf den aktuellen Lernkontexten berücksichtigt (Streicher und Smeddinck 2016). Zudem scheinen Schüler effektiver zu lernen, wenn das Lernmaterial angepasst und individualisiert wird (Woolf 2009).

Das Domänenmodell beschreibt Woolf (2009) als qualitative Darstellung von Expertenwissen in einer bestimmten Domäne und umfasst einen Themenbereich (bspw. Einführung in Physik), der gelernt werden soll. Diese Domänenmodelle können hoch komplex und hochgradig vernetzt sein. Das Wissen eines Schülers kann als annotierte Version des Domänenwissens dargestellt werden. Solche Modelle zeigen den Unterschied zwischen Anfängern und Experten, indem sie angeben, ob Benutzer ein Thema beherrschen, Lerneinheiten mehr Aufwand erfordern oder ob Wissen gänzlich fehlt. (Woolf 2009)

Schülermodelle (Student Models) stellen typischerweise das Verhalten, Interaktionen, Antworten und Ergebnisse eines Schülers dar. Student Models können zudem Missverständnisse aufdecken, indem sie Interaktionen des Schülers mit speziellen Denkmustern vergleichen (Woolf 2009).

3.6 Graphentheorie

In diesem Kapitel werden grundlegende Begriffe und Definitionen aus der Graphentheorie definiert, die in dieser Arbeit verwendet werden, um Graph-basierte Lernpfade anzuwenden. Zunächst werden die verschiedenen Bestandteile und Darstellungsformen von Graphen aufgezeigt. Anschließend werden verschiedene Arten von Graphen unterschieden. Ebenfalls wird geklärt, wie sich Pfade in Graphen widerspiegeln. Zum Schluss werden zusätzlich Algorithmen vorgestellt, die auf Graphen angewendet werden können, um kürzeste Pfade zu berechnen.

Diestel (2017) beschreibt einen Graph G bestehend aus einer Menge von Knoten V und Kanten E . Ein Graph ist in der Regel ein Tupel $G = (V, E)$. Die Kantenmenge E ist eine disjunkte Menge, wobei $E \subseteq |V|^2$ gilt. Eine Kante ist somit eine zweielementige Teilmenge der Knoten V . Die visuelle Darstellung eines Graphen ist von seiner formalen Definition unabhängig. Somit ist die Art und Weise, wie man die Knoten durch die Kante verbindet, ob geschwungen oder gerade, nur eine Frage der Ästhetik und der Zweckmäßigkeit. (Diestel 2017)

In Abbildung 3.8 wird ein Graph mit der Knotenmenge $V = \{A, B, C, Z\}$ und der Kantenmenge $E = \{(A, B), (A, C), (B, Z), (C, Z)\}$ gezeigt.

Ein **gerichteter Graph** besteht aus einer Menge von Knoten V und einer Menge von Kanten E . Die Menge E setzt sich aus Paaren von Knoten aus V zusammen, wobei $E \subseteq V \times V$ gilt (Briskorn 2020). Ein Beispiel eines gerichteten Graphen ist in Abbildung 3.8 auf der rechten Seite zu sehen. Eine Kante aus der Menge E in Abbildung 3.8 ist zum Beispiel $(A, B) \in E$, wobei gilt $(B, A) \notin E$, da es sich hierbei um einen gerichteten Graphen handelt.

Auch wenn sich die zwei Graphen aus Abbildung 3.8 sehr ähneln, handelt es sich hierbei um unterschiedliche Graphen. Beispielsweise existiert die Kante von dem Knoten B nach A in Abbildung G_1 , jedoch nicht in G_2 , also $(B, A) \notin E_{G_2}$.

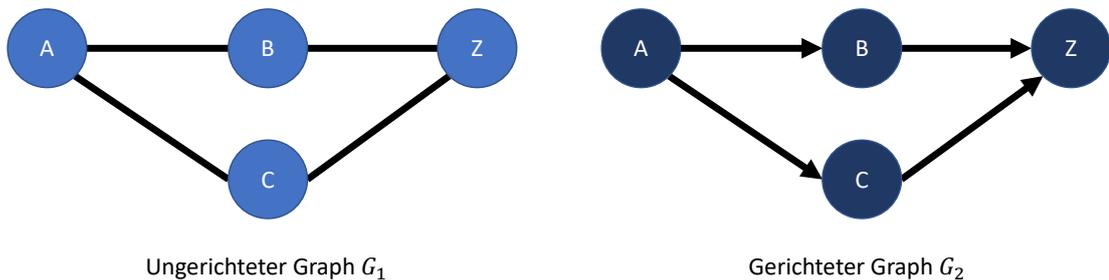


Abbildung 3.8: Beispiel eines ungerichteten und eines gerichteten Graphen

Ein **gewichteter Graph** besteht aus einem Tripel $G_3 = (V, E, w)$ mit einer zusätzlichen Funktion w , die jeder Kante aus E ein Gewicht in Form einer Zahl zuordnet (Briskorn 2020). Abbildung 3.9 zeigt einen gewichteten Graphen G_3 , wobei für das Gewicht zwischen den Knoten A und B folgendes gilt: $w(A, B) = 2$.

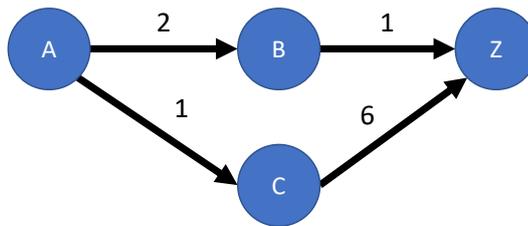


Abbildung 3.9: Beispiel eines gewichteten Graphen G_3

Die Adjazenzmatrix ist eine weitere Darstellungsform von Graphen und wird in dieser Arbeit verwendet, um diese in Datenstrukturen zu speichern. Turau (2009) definiert Adjazenzmatrizen als eine quadratische Matrix mit n Zeilen und n Spalten. Für eine Adjazenzmatrix $A \in \mathbb{R}^{n \times n}$ mit n Knoten, gilt für den Eintrag $a_{ij} \in A$, dass $a_{ij} \neq 0$, sobald eine Kante vom Knoten i nach j führt. Gilt $a_{ij} = 0$, existiert keine Kante, vorausgesetzt es gibt keine negativen Gewichte. (Turau 2009) Die Matrix 3.1 zeigt die Adjazenzmatrix A_3 des gewichteten Graphen G_3 aus Abbildung

3.9. Die erste Zeile der Matrix A_3 zeigt an, welche Kanten von Knoten A ausgehen, die zweite Zeile zeigt die Kanten von Knoten B aus an usw. Da $w(C, Z) = 6$ gilt, ist $a_{3,4} = 6$. Zudem ist zu erkennen, dass die letzte Zeile eine Nullzeile ist, da von Z keine Kanten ausgehen.

$$A_3 = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.1)$$

Ein Weg innerhalb eines Graphen kann durch die natürliche Folge der besuchten Knoten $P = x_0x_1\dots x_k$ beschrieben werden (Diestel 2017). Abbildung 3.8 zeigt den Weg $P = ABC$, mit Startknoten A und Endknoten C . Ein Weg von einem Knoten A nach Z ist eine Abfolge von Kanten, die in der gegebenen Reihenfolge abgelaufen wird, wobei am Knoten A begonnen wird und der Knoten Z erreicht werden soll (Briskorn 2020). In Abbildung 3.8 wird ein solcher Weg \mathcal{W} von A bis Z bestehend aus $\mathcal{W} = A, B, Z$ bzw. $\mathcal{W}' = A, C, Z$ dargestellt.

Ein Kreis ist ein Weg der von einem Startknoten zu demselben Endknoten führt (Briskorn 2020). In Abbildung 3.8 ist beispielsweise der Kreis $\mathcal{K} = A, B, Z, C, A$ zu erkennen. Ein Pfad ist eine spezielle Form eines Wegs, der keinen Kreis enthält bzw. einen Knoten höchstens einmal besucht (Briskorn 2020). Damit ist der Weg \mathcal{K} in Abbildung 3.8 kein Pfad.

Kürzeste Pfade sind spezielle Wege, die eine minimale Anzahl an Kanten zwischen einem Start- und Endknoten besitzen und keinen Kreis beinhalten (Briskorn 2020). Zur Berechnung kürzester Pfade existieren mehrere Algorithmen. Im Folgenden werden die von Needham und Hodler (2019) beschriebenen Algorithmen, die von der Graphen-Datenbank “Neo4j” („Neo4j“ 2022) angeboten werden, näher beschrieben. Der Algorithmus von Dijkstra (1959) ist einer der bekanntesten Algorithmen zur Berechnung eines kürzesten Pfades. Dieser bezieht die Kantengewichte mit ein und betrachtet zuerst die Kanten mit dem niedrigsten Gewicht. Pro Iteration wird jeweils der nächste Knoten mit dem niedrigsten Gewicht ausgewählt. Dadurch wird der Pfad vom Startknoten bis zum Endknoten mit der niedrigsten kumulativen Summe aus Gewichten bestimmt. (Needham und Hodler 2019)

Ein weiterer Algorithmus zur Bestimmung kürzester Pfade ist Yen’s k -Shortest-Paths Algorithmus (Yen 1971). Dieser ist in der Lage nicht nur einen, sondern bis zu $k - 1$ kürzeste Pfade zu bestimmen. In Abbildung 3.9 lässt sich für den Graphen G_3 von Knoten A nach C , der kürzeste Pfad $K = ABC$ nach nur drei Iterationen bestimmen, wobei die Gewichtssumme 3 beträgt.

Um Algorithmen auf Graphen anzuwenden, werden in dieser Arbeit, unter Anwendung des Graph Performance Service (GPS), die Graphen-Datenbank Neo4j verwendet.

3.7 Graph-basierte Modellierung von Lernpfaden

In dieser Arbeit werden Graph-basierte Ansätze zur Bestimmung der nächsten Lerneinheit operationalisiert. Hierbei werden Graph-basierte Lernpfade genutzt, um eine Progress Performance und eine Lernempfehlung zu bestimmen. In diesem Kapitel wird erklärt, wie sich solche Pfade in Graphen widerspiegeln und wie sich mithilfe von Referenzmodellen eine Performance berechnen lässt. Abschließend wird aufgezeigt, wie die berechnete Performance genutzt werden kann, um eine Adaptivität zu ermöglichen. Zunächst werden die verschiedenen Arten von Pfaden, darunter auch ideale Pfade bzw. ideal paths (IPs) definiert. Darauf basierend wird die Berechnung der Progress Performance mithilfe von IPs definiert. Anschließend wird geklärt, wie mithilfe der gemessenen Performance adaptive Lernpfade ermöglicht werden. Zum Schluss wird die aktuelle Version des GPS, die im Rahmen von Baur (2022) implementiert wurde, analysiert und beschrieben.

3.7.1 Nutzungspfade, Lernpfade und ideale Pfade

Streicher u. a. (2021) beschreiben Nutzungspfade als eine geordnete Sequenz von Benutzerinteraktionen, in Form von Knoten, mit einem System. Diese Nutzungspfade werden mithilfe von Beobachtungsdaten erstellt und in Form von gerichteten Graphen $G = (V, E)$ modelliert. Datengetriebene Nutzungspfade geben Hinweise darauf, wie Benutzer durch das System navigieren. Sie genügen allerdings nicht, um das benötigte student model (didaktisches Modell) im E-Learning Kontext zu offenbaren. Solche Beobachtungsdaten können beispielsweise den Beginn einer Session in Form eines Logins oder das Abschließen einer bestimmten Lerneinheit darstellen. Benutzerinteraktionen in Form von Knoten können zusätzliche Informationen enthalten, die mithilfe von Beobachtungsdaten gesammelt werden können. (Streicher u. a. 2021)

Laut Streicher u. a. (2021) kann die Sequenzierung von Interaktionen und die Modellierung von Nutzungspfaden dabei helfen, nächste Lerneinheiten zu empfehlen. Eine spezielle Form solcher Nutzungspfade sind sogenannte Lernpfade, die ein bedeutender Bestandteil adaptiver E-Learning Systemen sind. Lernpfade können dazu genutzt werden, um zu prüfen, ob sich der Benutzer auf dem richtigen Weg befindet. Lernpfade sind typischerweise vordefinierte Nutzungsmuster innerhalb eines Kontexts in E-Learning Plattformen. Diese Pfade können Informationen über die Struktur der Lerneinheiten übermitteln und einen benutzerspezifischen Lernfortschritt ermitteln. (Streicher u. a. 2021)

Dieser Lernfortschritt wird in dieser Arbeit als Progress Performance $Perf^P$ bezeichnet und soll durch Anwendung des GPS gemessen werden.

3.7.2 Ideal Paths Models (IPMs)

Die Referenzmodelle namens Ideal Paths Models (IPMs) können zur Kontrolle der Adaptivität genutzt werden (Streicher u. a. 2021). IPMs dienen zur Beschreibung der effizientesten Pfade, die auf direktem Weg über ideale Pfade bzw. Ideal Paths (IPs), zur nächsten Zielerreichung führen (Streicher u. a. 2018). Diese Pfade können aus benötigten (Lern-)Einheiten bestehen, die zur Zielerreichung führen (Streicher u. a. 2021). Der Graph Performance Service (GPS) verwendet zur Berechnung der Progress Performance die Referenzmodelle der IPMs. In dieser Arbeit wird der GPS neben dem GKT (Kapitel 3.10) zur Durchführung der Operationalisierung verwendet. Zudem zeigt diese Arbeit allerdings auch, dass es sich bei den IPs, um die effizientesten Pfade handelt. Baur (2022) umschreibt IPs als Expertenpfade. Im Kontext von Spot-X würde ein Experte beim Spielen der Aufgaben stets eine hohe Punktzahl erreichen und relativ schnell die Schwierigkeitskategorie "schwer" erreichen. Nachdem die Benutzerdaten in Form von xAPI-Statements gesammelt wurden, können laut Streicher u. a. (2021), alle Knoten und Kanten für mögliche Pfade in Graphen abgebildet werden. Das Analysieren der Graphen kann dabei helfen, individuelle oder auch gemeinsame Lernpfade zu identifizieren und somit auch IPMs herzuleiten. Der kürzeste Pfad kann beispielsweise mithilfe des Yen's k -Shortest-Paths Algorithmus (Needham und Hodler 2019) berechnet werden. (Streicher u. a. 2021)

Ein weiteres Beispiel wird in Abbildung 3.10 aufgezeigt, indem ein möglicher Nutzungspfad $P = \{A, B, C, D, E, F\}$ abgebildet wird, wobei sich dieser bei verschiedenen Benutzern unterscheidet. Mithilfe mehrerer Nutzungspfade kann ein gemeinsamer Pfad $P = \{A, C, F\}$ abgeleitet werden (Streicher u. a. 2021), bei dem nur drei, statt sechs Lerneinheiten benötigt werden. Aufgrund der Subjektivität des persönlichen Lernziels des Benutzers, dient der gemeinsame Pfad allerdings nur als Basis zur Bestimmung IPs (Streicher u. a. 2021).

Abbildung 3.11 zeigt ein Beispiel zweier IPs zur Erreichung des Lernziels "Simple Math Test Out" (Baur 2022).

Stehen die IPs fest, können diese genutzt werden, um Lernempfehlungen zu bestimmen. Knoten bzw. Lerneinheiten die sich auf diesen IPs befinden, jedoch nicht auf dem Nutzungspfad des betrachteten Benutzers, können potenzielle Lernempfehlungen darstellen (Streicher u. a. 2021). Um eine Adaption mit personalisierten, individuellen Pfaden in SGs zu ermöglichen, sollten laut Streicher und Smeddinck (2016), die eingesetzten Lerneinheiten modular und individuell nutzbar sein. Die Lerneinheiten sollten also nicht fest integriert sein, z. B. in einer festen Reihenfolge, sondern anpassbar und dynamisch einsetzbar sein (Streicher und Smeddinck 2016).

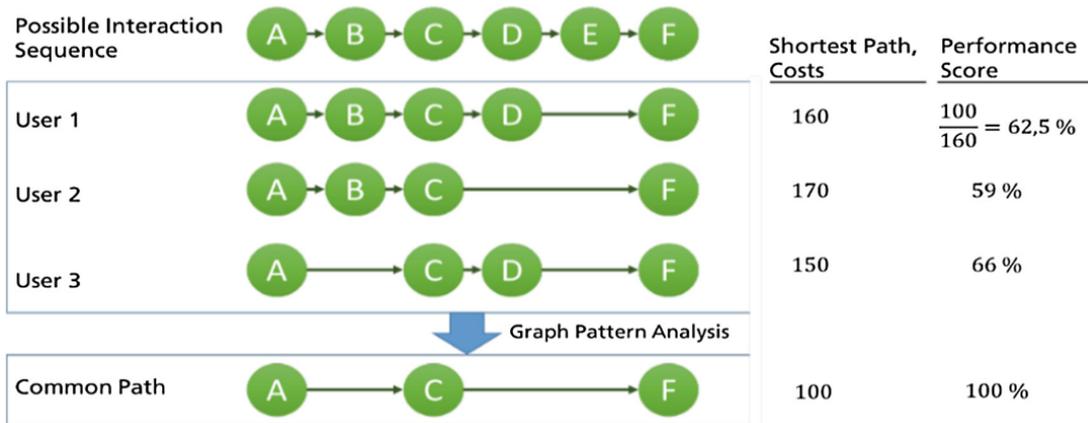


Abbildung 3.10: Beispiel eines gemeinsamen Pfads zur Berechnung einer Performance (Streicher u. a. 2021)

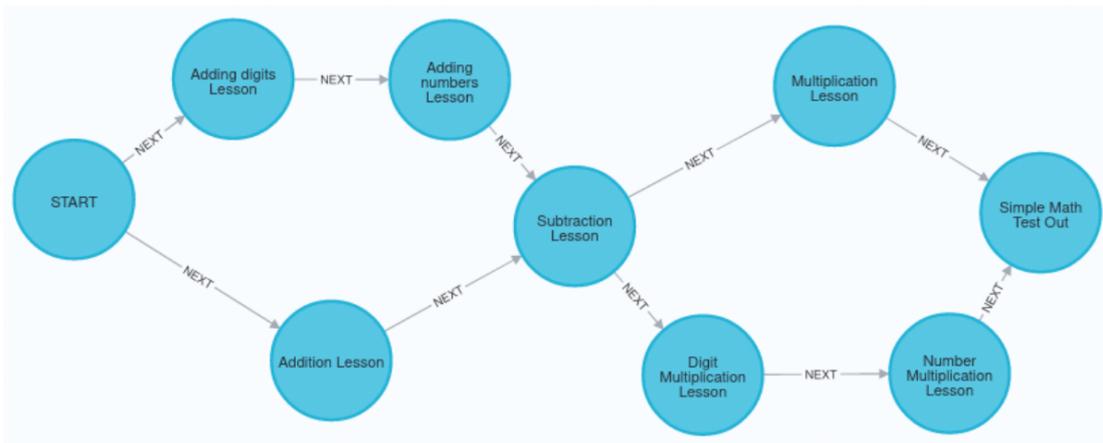


Abbildung 3.11: Beispiel von zwei idealen Pfaden (Baur 2022)

3.8 Adaptive Lernpfade und Performance-Messung

In diesem Kapitel wird der Begriff der Performance im Kontext von Lernpfaden in Serious Games (SGs) definiert. Um eine Adaption an den Benutzer zu ermöglichen, kann hierzu eine Form von Performance-Messung verwendet werden (Pavlik u. a. 2009). Unter dem Begriff Performance werden in der Literatur verschiedene Definitionen verwendet und meist nur allgemein behandelt, da die Performance vor allem von der betrachteten Domäne abhängt (Dörner u. a. 2016). Dörner u. a. (2016) erwähnen, dass mithilfe einer gemessenen Performance die Auswirkungen auf die Lernerfahrung abgeschätzt werden, um so anschließend darauf reagieren zu können. Die Berechnung einer Performance kann in SGs Prozesse, Ergebnisse oder

Benutzerinformationen mitberücksichtigen. Dabei kann es durchaus vorkommen, dass eine identische Performance durch unterschiedliche Ereignisse gemessen wird, z.B. die Note oder die Leistungspunktzahl. Die Performance-Messung ist domänenspezifisch und hängt damit von der Anwendung ab. Messwerte für die Performance können beispielsweise das Wissen, die Attitüde oder die Fähigkeiten eines Benutzers sein. (Dörner u. a. 2016)

Mithilfe der gemessenen Performance können Auswirkungen der aktuellen Parametereinstellungen (bspw. die Schwierigkeitsstufe) auf die Spielerperformance abgeschätzt werden (Streicher und Smeddinck 2016). Die gemessene Performance kann anschließend für Anpassungsmechanismen verwendet werden, um bspw. die Schwierigkeit anzupassen, sodass die Performance in einen gewünschten Bereich geregelt wird (Streicher und Smeddinck 2016). Durch die gewonnene Adaptivität soll z. B. die Motivation und die Effektivität des Lernens verbessert werden (Dörner u. a. 2016). Konkret wird die Performance-Vorhersage laut Dörner u. a. (2016) dazu verwendet, um zukünftiges Wissen oder die Performance eines Benutzers zu schätzen. Vorhersagemethoden zielen im Allgemeinen darauf ab, eine bestimmte Variable (z.B. die Punktzahl) vorherzusagen. Dies kann mithilfe einer Kombination anderer Indikatoren aus Daten geschehen. Die Vorhersage soll frühzeitig Indikatoren finden, die zum Lernerfolg, Versagen oder zu potenziellen Aussetzern führen können. Daraufhin können diese Indikatoren dazu verwendet werden, um den Spieler proaktiv zu unterstützen. (Dörner u. a. 2016)

In dieser Arbeit wird die Performance als Zusammensetzung zweier verschiedener Messwerte verstanden. Ein Bestandteil der Performance bezeichnet Dörner u. a. (2016), als das Ergebnis von Aktionen und wird im Weiteren als "Result Performance" bezeichnet. Die Result Performance $Perf^R$ kann sich beispielsweise mithilfe der Performance Factor Analysis (Pavlik u. a. 2009) bestimmen lassen oder mithilfe des hier angewendeten GKT vorhersagen lassen. Der zweite Bestandteil der hier definierten Performance wird als "Progress Performance" ($Perf^P$) bezeichnet (Baur 2022). Die gemessene $Perf^P$ soll nach Baur (2022) angeben, wie effektiv ein Benutzer lernt misst im Allgemeine eine Fortschrittsgeschwindigkeit messen.

Im Umfang dieser Arbeit, wird jedoch von der Definition von (Baur 2022) abgewichen. Da hierzu keine ähnliche Bezeichnung einer solchen Performance in der Literatur gefunden wird, bezeichnet in dieser Arbeit die gemessene $Perf^P$, die "Effizienz" der Zielerreichung. Auf diesen Messwert wird in dieser Arbeit besonderen Fokus gelegt, um die Adaptivität der Lernpfade zu regeln. Im Folgenden wird erklärt, wie sich mithilfe von IPMs eine Progress Performance ($Perf^P$) berechnen lässt. Also durch Bestimmung und Optimierung der Fortschrittsgeschwindigkeit des effizienten Lernens.

Allgemein lässt sich nach Baur (2022) die $Perf^P$ berechnen, indem der aktuelle Lernpfad und ein IP verglichen und gewichtet kombiniert wird. Die $Perf^P$ lässt sich berechnen, indem man die Übereinstimmungen des aktuellen Nutzungspfads des Benutzers, mit den idealen Pfaden (IPs)

bestimmt. Dies bedeutet, dass je höher die Übereinstimmung des aktuellen Nutzungspfad mit IPs, desto höher ist seine $Perf^P$ und somit auch seine Lerneffizienz. Im Umkehrschluss bedeutet dies, dass man dieses Referenzmodell und IPs dazu verwenden kann, um adaptive Lernpfade zu ermöglichen. Dabei können Lerninhalte dynamisch empfohlen werden, die möglichst nah oder womöglich auf IPs liegen, um eine höhere $Perf^P$ zu erreichen. Die gemessene $Perf^P$ kann demnach nicht nur genutzt werden, um unvollständige Pfade mit idealen Pfaden zu vergleichen, sondern ermöglicht zudem adaptive Lernpfade. (Baur 2022)

IPMs und Lernpfade können dazu genutzt werden, eine aktuelle Fortschrittsgeschwindigkeit auf dem Pfad zu ermitteln (Streicher u. a. 2021; Baur 2022). Der von Streicher u. a. (2021) vorgestellte Graph-basierte Ansatz verknüpft IPMs mit der Domäne und User Models. Benutzer, die sich am Startknoten eines IP befinden, können die gleiche Progress Performance besitzen, wie Benutzer, die das Lernziel über einen IP bereits erreicht haben (Baur 2022).

Um die Progress Performance auf einem aktuellen Nutzungspfad zu berechnen, definiert Baur (2022) folgende Formel:

$$Perf_1^P(P_U) = \frac{Progress}{Time} = a \frac{|V(P_U \cap V(P_I))|}{V(P_U)} + (1 - a) \frac{|E(P_U \cap E(P_I))|}{E(P_U)} \text{ mit } Perf^P \in [0,1]$$

Adaptive Lernpfade nutzen das Konzept der adaptiven Empfehlung zur benutzerspezifischen Anpassung nächster Lerneinheiten (Baur 2022). Bei der adaptiven Anpassung von Pfaden unterscheiden Streicher u. a. (2021) zwischen der Anpassung von Makropfaden und der Anpassung von Mikropfaden. Bei der Makroanpassung wird der gesamte Pfad betrachtet. Dabei wird dem Benutzer ein passender Lernpfad vorgeschlagen und ggf. angepasst (z. B. über die Reihenfolge). Die Mikroanpassung bezieht sich auf die unmittelbare Adaption einzelner Elemente bzw. Knoten innerhalb von Pfaden. (Streicher u. a. 2021)

Adaptive SGs benötigen nach Streicher u. a. (2021) zudem Referenzmodelle zur automatische Anpassung, um so das richtige Timing und den richtigen Pfad bestimmen zu können. Anhand von Modellwissen auf der Ebene des Lernpfads bzw. mithilfe einer Reihenfolge der Inhalte können individuelle Empfehlung der nächsten Lernobjekte vorgeschlagen werden. Ideal Path Models (IPMs) dienen als Referenzmodell zur adaptiven Kontrolle in adaptiven Systemen. (Streicher u. a. 2021)

Bei der Nutzung von IPMs, kann es durchaus vorkommen, dass mehrere IPs ermittelt werden. Dennoch soll derjenige IP ausgewählt werden, der dem aktuellen Nutzungspfad des Benutzers am ähnlichsten ist. Zur Auswahl des besten IP bzw. der nächsten Lerneinheit wird somit der IP gewählt, der zu der höchsten Progress Performance führt (Baur 2022):

$$Perf^P(P_U) = \max_{P_I \in IP_s} Perf_I^P(P_U, P_I)$$

3.9 Graph Performance Service (GPS)

Zur Berechnung einer Progress Performance innerhalb von Lernpfaden, wurde der Prototyp Graph Performance Service (GPS) mit Python implementiert (Baur 2022).

Metaphorisch gesehen, kann man sich den GPS als eine Art Navigationssystem vorstellen. Dabei navigiert der GPS den Lernenden mithilfe mehrerer Empfehlungen und weist ihn über den effektivsten und effizientesten Lernpfad zu einem festgelegten Lernziel. Um Lerneinheiten empfehlen zu können, misst der GPS, wie effektiv und effizient ein Benutzer auf seinem aktuellen Lernpfad lernt. Hierfür wird der aktuelle Lernpfad mit den berechneten idealen Pfaden (IPs) verglichen. (Baur 2022)

Der GPS kann mittels einer REST-API angefragt werden, um eine Progress Performance sowie die nächste Lernempfehlung zu erhalten. Eine Anfrage könnte beispielsweise folgendermaßen aussehen: “.../response?user=alice@example.com&goal=http://example.com/Lernziel” (Baur 2022). Die Antwort erfolgt in Form eines textbasierten JSON-Format (siehe Listing 3.3).

```
1  {
2      "success": true,
3      "performance": 0.5,
4      "recommendation": {
5          "next_activity": {
6              "id": "https://example.com/courseX/Lesson1",
7              "name": "https://example.com/definition/lesson"
8          }
9      }
10 }
```

Listing 3.3: Beispiel einer Ausgabe des GPS im JSON-Format (Baur 2022)

Baur (2022) beschreibt die drei Hauptbestandteile der Ausgabe wie folgt:

- “success”-Feld gibt an, ob die Anfrage erfolgreich bearbeitet wurde
- “performance”-Feld gibt die berechnete Progress Performance in Form einer Gleitkommazahl zurück
- “recommendation”-Feld gibt die nächste Lernempfehlung, bestehend aus einer ID und Name zurück

Der GPS verwendet zur Generierung und Speicherung der Graphen, die Open-Source-Graphdatenbank “Neo4j” (Baur 2022). Neo4j bietet neben der Graph-Modellierung, die Möglichkeit Graph-Algorithmen über sogenannte Cypher-Anfragen durchzuführen („Neo4j“ 2022).

Baur (2022) verwendet zur Modellierung der Nutzungspfade eine Sequenz von Lerneinheiten von einem Start- zu einem zuvor festgelegten Zielknoten. Zur Generierung der Graphen und um den datengetriebenen Ansatz zu füttern, werden xAPI Daten verwendet, die zur Laufzeit aus einem Learning Record Store (LRS) geladen werden. Die Akteure und Objekte aus den xAPI-Statements werden als Knoten abgebildet und die xAPI-Verben als Kanten hinzugefügt. (Baur 2022)

3.10 Graph-based Knowledge Tracing (GKT)

Das Graph-based Knowledge Tracing (GKT) von Nakagawa u. a. (2019), soll in dieser Arbeit verwendet werden, um geeignete Lerninhalte mittels einer Performance-Vorhersage zu empfehlen. Beobachtungsdaten und Lernerfahrungen in Graphen abzubilden helfen dabei, zu verstehen, warum bestimmte Lerneinheiten empfohlen werden (Battaglia u. a. 2018; Nakagawa u. a. 2019).

Das GKT von (Nakagawa u. a. 2019) wird in dieser Arbeit als Vergleich zum GPS anhand von Benutzerdaten aus Spot-X angewendet. Das GKT wurde nicht explizit zur Generierung idealer Pfade wie der GPS entwickelt, sondern zur Vorhersage einer Performance und Ermittlung von Abhängigkeiten zwischen Lernkonzepten. Das GKT verbessert die Interpretierbarkeit der Performance-Vorhersagen, indem es ein Kompetenzniveau für jedes Konzept und für jeden Benutzer eindeutig und angemessen modelliert. Im Kontext des GKT stellt die Performance die Wahrscheinlichkeit dar, ob ein Benutzer die Lerneinheit richtig beantworten wird. Das GKT ist in der Lage, auf Basis von Benutzerdaten, zeitliche Übergänge des Benutzerwissens für jedes Konzept mittels Graph-Algorithmen zu berechnen. Dieser Ansatz dient vor allem für Themengebiete, in denen eine implizite Abhängigkeitsstruktur der Lernkonzepte existiert. (Nakagawa u. a. 2019)

Für den Einsatz des GKT werden nach Nakagawa u. a. (2019) folgende Annahmen über den Kursaufbau und dessen Wissenskonzepte getroffen:

- Die Wissenskonzepte liegen einer latenten Graph-Struktur zugrunde und sind potenziell als Graph $G = (V, E)$ strukturiert und modellieren die zeitlichen Übergänge der studentischen Kenntnisse für jedes Konzept.
- Die Wissenskonzepte werden in N Knoten zerlegt: $V = \{v_1, \dots, v_N\}$.
- Die Konzepte teilen Abhängigkeitsbeziehungen in Form von Kanten: $E \subseteq V \times V$.

Abbildung 3.13 zeigt vier Methoden von Nakagawa u. a. (2019) zur Generierung der latenten Wissensstruktur. Diese Ansätze unterscheiden sich in den ausgegebenen Graphen. In der Durchführung der Operationalisierung wird der Variational Autoencoder (VAE) von Kingma und Welling (2014), der Dense-Graph und der Transition-Graph zur Performance-Vorhersage verwendet. Der VAE verwendet hierbei eine parametrisierten Adjazenzmatrix, der durch latente Variablen, die Kanten typisiert. Dadurch sollen gewisse Knotenmerkmale abgeleitet werden. (Nakagawa u. a. 2019)

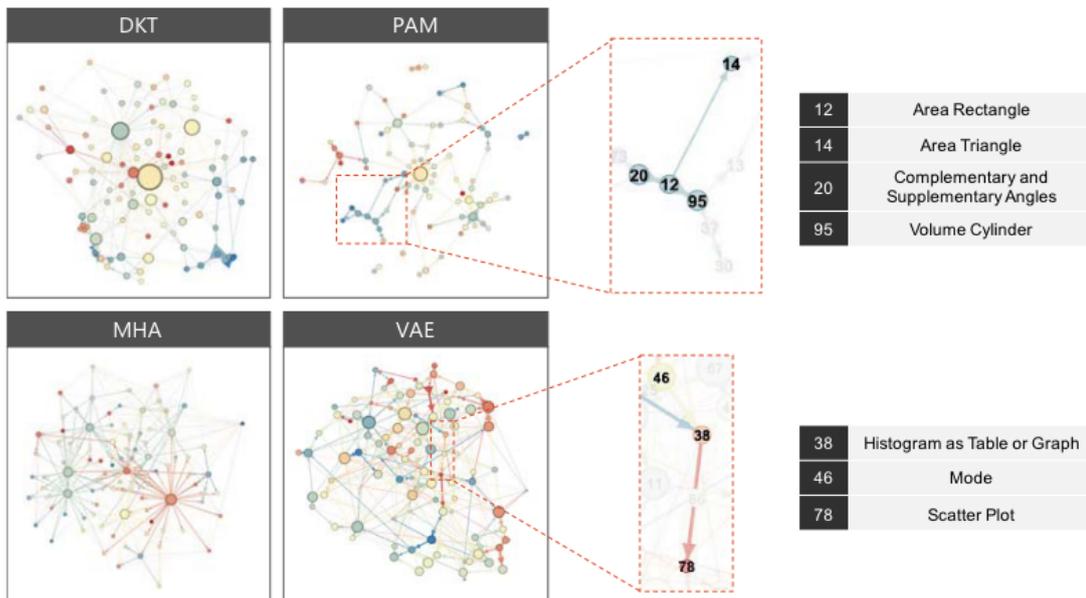


Abbildung 3.13: Generierung impliziter Wissensstrukturen durch das GKT (Nakagawa u. a. 2019)

Abbildung 3.14 zeigt die grundlegende Architektur und das allgemeine Vorgehen des GKT. Hierbei wird der Prozess zur Vorhersage der Performance in die drei Schritte Aggregate, Update und Predict unterteilt (Nakagawa u. a. 2019).

Aggregate: Im ersten Schritt werden die zum beantworteten Konzept dazugehörigen bzw. verwandten Konzepte aggregiert. Hierbei wird bereits unterschieden, ob ein Konzept richtig oder falsch beantwortet wurde. Die Eingabe für diesen Schritt umfasst die beantworteten Konzepte in Form von Vektoren: $x^t \in \{0, 1\}^{2N}$. Dabei wird für jedes Konzept angegeben, ob es zum Zeitschritt t richtig (1) oder falsch (0) beantwortet wurde. Matrix $E_x \in \mathbb{R}^{2N \times xe}$ beinhaltet die Indizes der Konzepte und die dazugehörigen Antworten mit den Vektoren x . Die Matrix $E_c \in \mathbb{R}^{N \times xe}$, die in Abbildung 3.14 genannt wird, bettet die Konzeptindizes ein, wobei $E_c(k)$ die k -te Reihe angibt. Die Matrix E_c gibt also für jedes Konzept in der jeweiligen Zeile an, ob

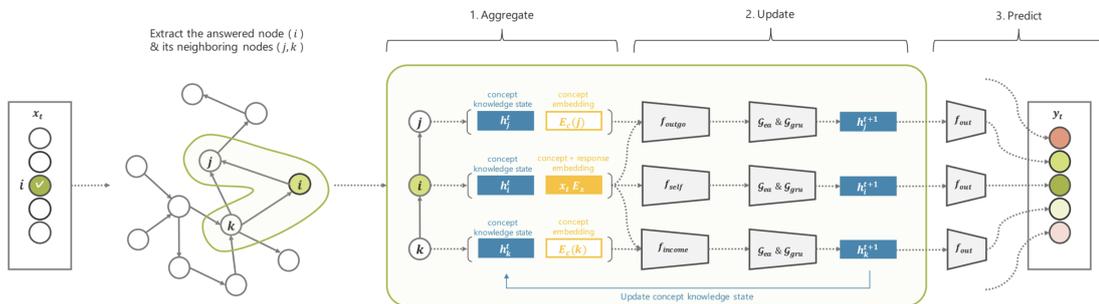


Abbildung 3.14: GKT-Architektur (Nakagawa u. a. 2019)

das Konzept zum Zeitschritt t richtig oder falsch beantwortet wurde. (Nakagawa u. a. 2019) Das Beispiel eines Vektors x für ein beantwortetes Konzept könnte folgendermaßen aussehen: $x = \{0, 0, 1\}$. x_1 gibt hierbei an, dass das Konzept in den ersten zwei Zeitschritten falsch beantwortet wurde, da $x^1 = 0$ und $x^2 = 0$.

Update: Im zweiten Schritt werden die relevanten bzw. benachbarten Konzepte auf Basis des Graphen aktualisiert. Dabei werden Konzepte angeglichen, die mit dem beantworteten Konzept in Verbindung stehen. Hierbei werden nur Wissensstände verwandter Konzepte aktualisiert und nicht die aller Konzepte. (Nakagawa u. a. 2019)

Predict: Im letzten Schritt wird schließlich eine Vorhersage für jedes eingegebene Konzept aus dem ersten Schritt zum nächsten Zeitschritt $t + 1$ ausgegeben. Mithilfe der Kanteninformationen werden für benachbarte Knoten, Wissensstände hergeleitet, um so eine Vorhersage für den nächsten Knoten bzw. Wissenskonzept herzuleiten. Dieser kann dazu genutzt werden, um vorherzusagen, wie wahrscheinlich ein Konzept richtig beantwortet wird. (Nakagawa u. a. 2019)

Zur Durchführung der Operationalisierung wird auf einer Implementierung des GKT¹ aufgebaut, um das GKT anzuwenden. Hierbei sollen für einzelne Spot-X Aufgaben eine Result Performance $Perf^R$ vorhergesagt werden, um damit Lernempfehlungen zu generieren.

1. jhljx. 2022. „GKT“. Original-date: 2020-09-18T14:47:51Z, 3. Juni 2022. Besucht am 3. Juni 2022. <https://github.com/jhljx/GKT>

4 Konzept für Graph-basierte Lernpfade in Serious Games (SGs)

In diesem Kapitel wird die Operationalisierung Graph-basierter Lernpfade für Serious Games (SGs) vorbereitet. Hierfür soll am konkreten Beispiel des SG Spot-X ein Konzept entwickelt werden. In dem Konzept wird zunächst eine Anforderungsanalyse für die Operationalisierung durchgeführt. Anschließend sollen die aus Spot-X vorhandenen Benutzerdaten, in Form der xAPI-Statements, in Lernpfade modelliert werden. Hierzu wird die Erzeugung der Knoten, Kanten und der dazugehörigen Pfade im Kontext von Spot-X betrachtet. Abschließend wird eine Verallgemeinerung des Konzepts erarbeitet. In Kapitel 5 wird das hier erarbeitete Konzept an Spot-X angewendet, operationalisiert und verifiziert.

Zur Anwendung des GPS und GKT müssen zunächst die domänenspezifischen Benutzerdaten in Graphen abgebildet werden. Hierzu muss geklärt werden, welche Arten von Interaktionen in der Domäne Spot-X existieren, beispielsweise das Einloggen in das System. Erst dann können diese Interaktionen als Pfade im Graph abgebildet werden. Welche Benutzerdaten und Interaktionen nun konkret in Form der xAPI-Statements verwendet oder gefiltert werden sollen, wird in der Durchführung der Operationalisierung in Kapitel 5 erläutert. Nachdem der domänenspezifische Interaktionsgraph und die Lernpfade modelliert wurden, müssen zusätzlich die Ideal Paths Models (IPMs) für Spot-X angewendet werden. Anschließend kann der GPS erweitert und angewendet werden.

4.1 Modellierung Graph-basierter Lernpfade für Spot-X

In diesem Abschnitt werden konkrete Graph-basierte Lernpfade für Spot-X modelliert. Um zu zeigen, wie Lernpfade in Spot-X aussehen, werden zunächst die Interaktionsgraphen modelliert. Hierzu wird durch Verwendung des Triple-Prinzips der xAPI-Statements auf die erarbeitete Modellierung von Baur (2022) aufgebaut. Zusätzlich soll geklärt werden, wie Start- und Endknoten im Kontext des Quiz-Spiels Spot-X festgelegt werden und wie daraus ideale Pfade entstehen können.

4.1.1 Modellierung von Interaktionsgraphen

Zunächst werden Interaktionsgraphen modelliert, die mittels Benutzerdaten aus Spot-X generiert werden sollen. Ein Interaktionsgraph soll hierbei erhobene Benutzerdaten und deren Benutzerinteraktionen chronologisch darstellen. Die Modellierung baut hierbei auf die Implementierung des GPS nach Baur (2022) auf. Dazu werden im Allgemeinen die xAPI-Objekte und Akteure jeweils in Knoten und die xAPI-Verben in Kanten abgebildet (Baur 2022). Zuerst werden die Knoten und deren Attribute definiert und anschließend die dazugehörigen Kanten modelliert. Nachdem die Knoten und Kanten im Kontext von Spot-X feststehen, können Lernpfade und ideale Pfade definiert werden.

Modellierung der Nutzungs- und Lernpfade

In Spot-X gibt es verschiedene Ereignisse und Benutzerinteraktionen in Form von xAPI-Objekten, die potenziell als Knoten modelliert werden können. Im Folgenden werden Benutzer (Akteure), Aufgaben und weitere verschiedene Benutzerinteraktionen, wie das Einloggen, als Kandidaten für Knoten betrachtet.

Der Akteur wird in dieser Arbeit als Knoten modelliert, in dem die dazugehörigen Benutzerinformationen enthalten sind (z. B. Name, E-Mail). Der Akteur kann durch seine eindeutige ID über mehrere Spot-X Plattformen identifiziert werden.

Die xAPI-Objekte abgeschlossener Aufgaben gehören zu den wichtigsten xAPI-Statements und stellen hier Lerneinheiten dar, die als Knoten auf einem Lernpfad modelliert werden. Das xAPI-Objekt einer Aufgabe, enthält wichtige Informationen, wie Aufgabenname, Kategorie, Schwierigkeitskategorie usw. Neben den Akteuren und Aufgaben, können auch zusätzliche Benutzerinteraktionen, wie das Einloggen in das SG, aus den xAPI-Objekten entnommen und als Knoten modelliert werden. Diese Knoten können im Weiteren dazu verwendet werden, Nutzungs- und Lernpfade, aber auch ideale Pfade zu modellieren.

Ein Problem bei der Modellierung der Knoten ist die Sicherstellung der plattformübergreifenden Eindeutigkeit der Aufgaben. Spot-X kann auf mehreren Plattformen bzw. für verschiedene Lernumgebungen gleichzeitig bereitgestellt werden. Dennoch werden die gesammelten Benutzerdaten aller Spot-X Plattformen im gleichen Learning Record Store (LRS) gespeichert. Dabei besteht die Möglichkeit, dass eine Aufgabe auf mehreren Spot-X Plattformen verfügbar ist und gespielt wird. Allerdings arbeiten die einzelnen Spot-X Plattformen isoliert auf ihrer eigenen Aufgaben-Datenbank, was wiederum zu Inkonsistenz führen kann. Im Folgenden werden verschiedene Ansätze zur Modellierung der Pfade erstellt, unter der Berücksichtigung, dass mehrere Spot-X Plattformen "SG A" und "SG B" existieren. Tabelle 4.1 zeigt die vier erarbeiteten Modellierungen der Pfade in Spot-X.

Modellierung	Beschreibung
	Erster Ansatz: Für jede Spot-X Instanz wird für inhaltlich identische Aufgaben ein jeweiliger Knoten generiert. Bei n Spot-X Instanzen existieren für eine Aufgabe somit n Aufgaben-Knoten.
	Zweiter Ansatz: Für eine identische Aufgabe auf verschiedenen Instanzen wird nur ein Knoten generiert. Instanzen können als Attribut im Aufgaben-Knoten aufgelistet werden.
	Dritter Ansatz: Für eine identische Aufgabe wird nur ein Knoten generiert. Dabei wird auch für jede Spot-X Instanz ein zusätzlicher Knoten generiert, der die Instanz darstellen soll. Hierdurch können sich Lernpfade durch die verwendete Spot-X Instanz unterscheiden.
	Vierter Ansatz: Für eine identische Aufgabe wird nur ein Knoten generiert. Die Information über die verwendete Spot-X Instanz findet sich auf der Kante als Attribut wieder.

Tabelle 4.1: Verschiedene Ansätze zur Modellierung der plattformübergreifenden Lernpfade

Hierbei ist zu beachten, dass sich die Wahl von einem der vier Ansätze auf die Modellierung der Lernpfade und Performance-Messung auswirken kann. Einerseits soll es ermöglicht werden, gleiche Aufgaben über verschiedene Spot-X Instanzen zu identifizieren und deren Daten gleichermaßen zu nutzen. Andererseits sollen eindeutige Lernpfade über diese Knoten modelliert und unterschieden werden. Zur Erzeugung der Graphen wird in dieser Arbeit eine Kombination aus dem zweiten und dritten Ansatz umgesetzt. Dadurch sollen eindeutige Pfade ermöglicht und zugleich ein besserer Überblick der vorhandenen Spot-X Instanzen ermöglicht werden.

Um die Kanten zwischen den Knoten zu erzeugen, wird das Verb aus dem xAPI-Statement verwendet. Das Verb wird zur Beschreibung von Interaktionen und zur Angabe dynamischer Informationen, wie der erreichten Punktzahl verwendet. Bei einem Verb kann es sich z. B. um das Abschließen einer Aufgabe handeln. Dabei enthalten die Kanten kontextbezogene Informationen. Ein xAPI-Statement einer abgeschlossenen Aufgabe wird in Spot-X durch das Verb "COMPLETED" beschrieben.

Eine “COMPLETED”-Kante, ausgehend von einem Knoten, beinhaltet daher zeitbezogene und benutzerspezifische Informationen über die abgeschlossene Aufgabe. Eine solche Kante soll mehrere Informationen über eine abgeschlossene Aufgabe beinhalten, wie die benötigte Zeit, die erreichte Punktzahl, die Antwort des Benutzers und die Plattform.

Durch diese Modellierung der Kanten besitzt man die Möglichkeit, diese zusätzlichen Kanteninformationen als Kantengewichte in Graphen-Algorithmen miteinzubeziehen. Somit können in der Operationalisierung in Kapitel 5 beispielweise verschiedene Algorithmen zur Bestimmung kürzester Lernpfade bestimmt werden. Das Beispiel aus Abbildung 4.1 verdeutlicht, dass die Berücksichtigung der Kantengewichte zur Generierung idealer Pfade sinnvoll ist. Ohne die Berücksichtigung der Punktzahl enthält der Graph aus Abbildung 4.1 zwei kürzeste Pfade.

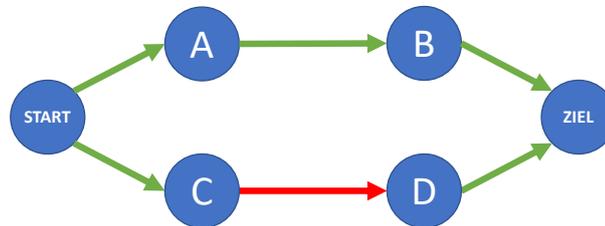


Abbildung 4.1: Problematik des Yen’s k -Shortest-Paths Algorithmus ohne Berücksichtigung der Kantengewichte

Nachdem die Knoten und Kanten in den vorherigen Abschnitten beschrieben wurden, werden diese zur Modellierung der Pfade in Spot-X kombiniert. Der Nutzungspfad eines individuellen Benutzers in dem Interaktionsgraphen erstreckt sich über eine chronologische Sequenz von Knoten. Diese Knoten können Benutzerinteraktionen oder Aufgaben darstellen. Abbildung 4.2 zeigt einen beispielhaften Pfad, der durch Benutzung des Triple-Prinzips erzeugt wurde. Hierbei wird konkret ein Pfad von “Alice” gezeigt, wobei die Aufgabe “Singapur” abgeschlossen wurde.

4.1.2 Modellierung der Ideal Paths Models (IPMs)

Wie in Kapitel 3.7.2 eingeführt, kann man sich einen idealen Pfad (IP) so vorstellen, als würde man den Lernpfad eines Experten entlanggehen. Die Gestalt von IPs wird im Folgenden abstrahiert und modelliert. Die Lerninhalte bzw. die gespielten Aufgaben in Spot-X werden zufällig aus einem Aufgabenpool genommen, der durch die Schwierigkeitskategorie (SK) eingeschränkt wird. In Spot-X sollte es nicht das Ziel sein, alle Aufgaben gespielt zu haben. Das Lernziel wird im Folgenden als die richtige Beantwortung einer Aufgabe der SK »schwer« festgelegt. Ein Vergleich des Nutzungspfades eines Benutzers mit einem IP am Beispiel von

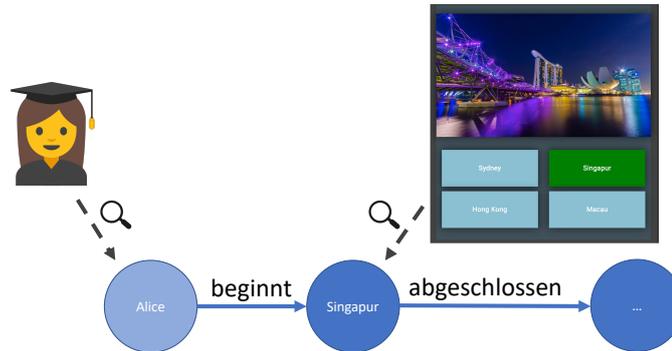


Abbildung 4.2: Darstellung des Triple-Prinzips als Graph am Beispiel Spot-X

Spot-X wird in Abbildung 4.3 dargestellt. Hierbei ist zu erkennen, dass der IP (grün) eine schwere Aufgabe deutlich effizienter erreicht als der aktuelle Pfad eines Benutzers (blau).

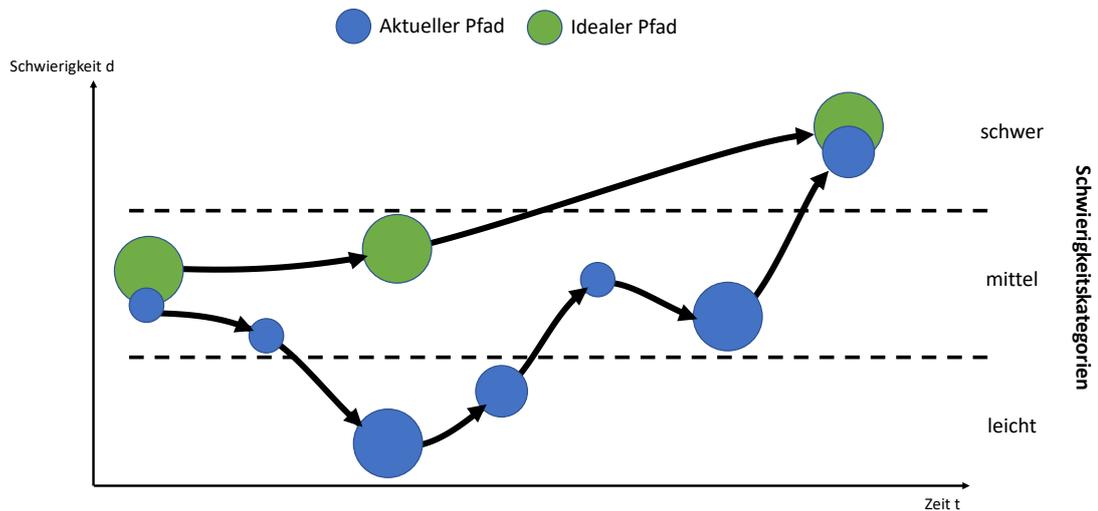


Abbildung 4.3: Vergleich der Lernpfade eines Benutzers und eines Experten (IP)

Die Problematik bei der Verwendung der Ideal Paths Models (IPMs) ist, dass hierbei ein konkretes Lernziel zur Bestimmung IPs benötigt wird (Baur 2022), z.B. in Form einer konkreten Aufgabe. Da in Spot-X die Aufgaben in keiner festgelegten Reihenfolge gespielt werden, ist eine Festlegung eines IPs oder eines konkreten Lernziels nicht intuitiv machbar. Demnach werden keine konkreten Aufgaben, sondern abstraktere Aufgabeneigenschaften genutzt, wie die SK in Kombination mit der dazugehörigen inhaltlichen Kategorie (z.B. Mathematik).

Ein definiertes Lernziel könnte auch als eine Art Checkpoint betrachtet werden. Somit werden hierzu keine konkreten Aufgaben-IDs verwendet, sondern Checkpoints in Form bereits erreichter SKs einer Kategorie. Ein Spieldurchlauf in Spot-X beginnt stets mit der SK »mittel« und verfolgt das Ziel, eine Aufgabe der SK »schwer« zu lösen. Somit wäre der Startknoten als »mittel« und der Endknoten als »schwer« definiert. Der Lernpfad eines Experten verläuft somit chronologisch von der SK »mittel« zu »schwer«.

4.2 Performance-Messung in Spot-X

Nachdem nun die Graphen im Kontext von Spot-X durch die erhobenen xAPI Daten generiert werden können, soll basierend auf diesen Graphen die Performance eines Benutzers berechnet werden. Zur Messung einer Performance werden im Folgenden die konkrete Anwendung des GPS und die des GKT für Spot-X für die Operationalisierung vorbereitet.

Der GPS kann nun den erzeugten Graphen in der Graphen-Datenbank Neo4j nutzen, um ideale Pfade zu berechnen. Darauf basierend wird eine Progress Performance berechnet, indem der aktuelle Lernpfad des Benutzers mit dem idealen Pfad, der die größte Ähnlichkeit aufweist, verglichen wird (siehe Abschnitt 3.9). Das GKT unterscheidet sich zum GPS insofern, dass es eine Performance-Vorhersage trifft. Diese Vorhersage wird in dieser Arbeit mit der Performance-Messung des GPS verglichen. Das GKT generiert aus den Spot-X Daten einen Graph der genutzt wird, um Abhängigkeiten zwischen Lernkonzepten zu erkennen. Dieser Erkenntniszuwachs über die Abhängigkeiten der Aufgaben wird genutzt, um eine Performance-Vorhersage für eine konkrete Spot-X Aufgabe zu treffen. Die Performance-Vorhersage gibt dabei die Wahrscheinlichkeit an, ob die Aufgabe richtig beantwortet werden wird. In dieser Arbeit wird vorhergesagte Performance verwendet, um die Aufgabe mit der höchsten Performance-Vorhersage zu wählen. Dadurch soll Schritt für jede Aufgabe eine Performance berechnet werden, um so implizit einen Pfad vorzuschlagen. Die gemessene Performance kann ebenso ein Indikator dafür sein, ob ein Benutzer mehr Unterstützung benötigt. Diese Messung kann bspw. dazu verwendet werden, um einen digitalen Assistenten zu aktivieren.

4.3 Verallgemeinerung des Konzepts

Durch Verwendung der Beobachtungsdaten, die dem xAPI-Standard folgen, wurde in der Erarbeitung dieses Konzepts deutlich, dass durch die generische Struktur des Triple-Prinzips, die Modellierung des Interaktionsgraphen für das SG Spot-X umsetzbar ist. Um das Konzept auf andere SGs anzuwenden, um dort ebenfalls Lernpfade zu generieren, ist dies durch das generische Triple-Prinzip umsetzbar. Allerdings sollte man darauf achten, welche Beobachtungsdaten wirklich in den Graphen abgebildet werden sollten und auf welche xAPI-Objekte ggf. mehr Wert gelegt werden sollte. Die Herausforderung bei der Modellierung der Lernpfade bzw. der idealen Pfade ist die Festlegung eines konkreten Lernziels. Allerdings konnte hier gezeigt werden, dass es durchaus Sinn macht, keine konkrete Lerneinheit als Ziel festzulegen. Stattdessen sollte ein charakterisierendes Ziel anhand seiner Eigenschaft festgelegt werden, bspw. die richtige Beantwortung einer "schweren" Lerneinheit. Dies wird durch die Möglichkeit zusätzlicher Informationen über das "Extension"-Feld eines xAPI-Statements ermöglicht, was zur Festlegung von sehr spezifischen oder sehr abstrakten Lernzielen verwendet werden kann. Beispielsweise könnte hierzu ein SG, das einen Gamification-Teil mit Achievements besitzt, betrachtet werden. Dabei wäre es beispielsweise möglich, ein Lernziel in Form der Erreichung eines beliebigen Achievements festzulegen. Somit könnte immer der effizienteste Lernpfad in Richtung des nächsten Achievements generiert werden, um so die extrinsische Motivation des Spielers zu steigern.

5 Operationalisierung Graph-basierter Lernpfade am Beispiel von Spot-X

In diesem Kapitel wird das erarbeitete Konzept aus Kapitel 4 umgesetzt, um eine Operationalisierung mit den vorgestellten Methoden am Beispiel von Spot-X durchzuführen. Mithilfe eines Beispielszenarios werden die Auswirkungen verschiedener Einflussfaktoren auf adaptive Lernpfade gezeigt. Hierzu soll die Justierung der Eingabe in Form der Benutzerdaten und Auswahl geeigneter Variablen, wie beispielsweise der erreichten Punktzahl, die Auswirkungen in Form einer Performance messen. Dazu soll ebenfalls verglichen werden, wie sich diese auf die adaptiven Lernpfade auswirken.

Zur Operationalisierung, werden zunächst einige Hypothesen, darunter auch die Nullhypothese festgelegt, die durch diese Operationalisierung belegt werden sollen. Anschließend werden die verfügbaren Benutzerdaten bzw. die xAPI-Statements inspiziert und potenzielle Einflussfaktoren zur Entstehung adaptiver Lernpfade ausgewählt. Sobald die Einflussfaktoren zur Justierung adaptiver Lernpfade feststehen, werden diese zur Generierung idealer Pfade und zur Performance-Messung operationalisiert. Hierfür wird ein Beispielszenario vorgestellt, das zur Operationalisierung verwendet wird.

Abbildung 5.1 zeigt den Vergleich der zwei Methoden die zur Operationalisierung angewendet werden. Zur Durchführung der Operationalisierung wird der Graph Performance Service (GPS) und das Graph-based Knowledge Tracing (GKT) angewendet. Dabei sollen ebenfalls Unterschiede und Gemeinsamkeiten der beiden Methoden aufgezeigt werden, indem die Eingabe, die Modellierung und die Ausgabe verglichen werden.

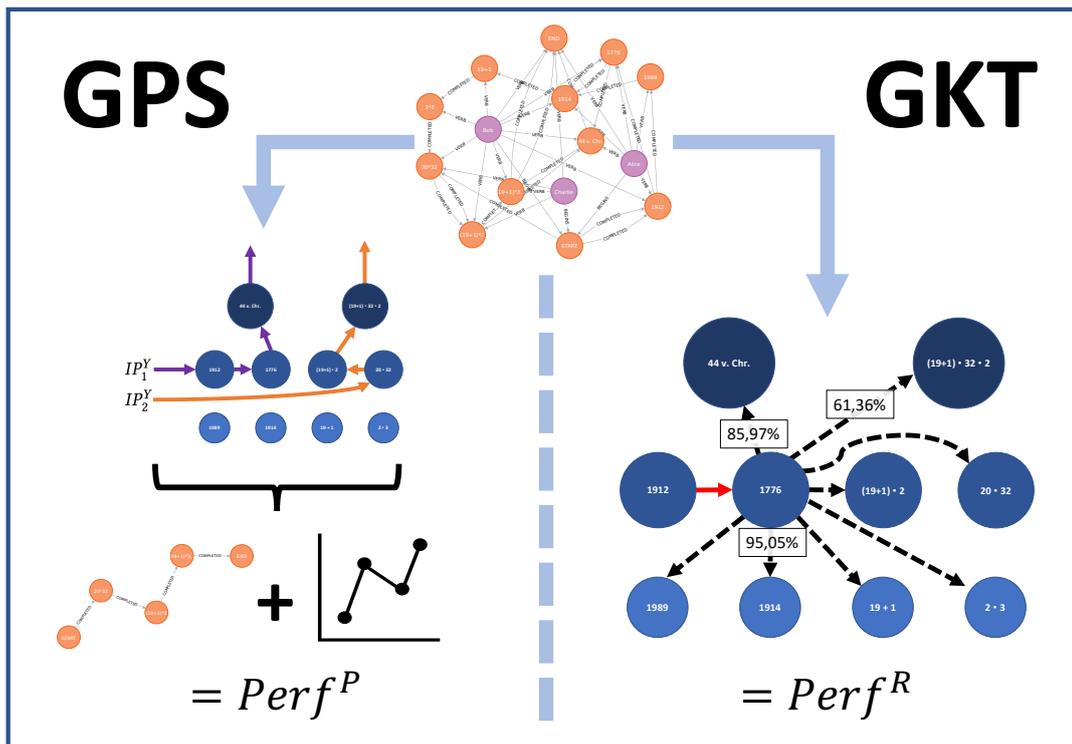


Abbildung 5.1: Vergleich des GPS und GKT

5.1 Festlegung der Hypothesen

Zunächst müssen Vermutungen und Annahmen über die Beobachtungsdaten in Form von Hypothesen getroffen werden. Dazu werden in diesem Abschnitt die Hypothesen bzw. Nullhypothesen festgelegt, die durch eine operative Messung der Merkmalsausprägung Graph-basierter Lernpfade in Beziehung gesetzt und geprüft werden sollen.

Die Nullhypothese H_0 lautet:

Die Lern-Effizienz (Progress Performance) steigt durch die Verwendung Graph-basierter Lernpfade und die Nutzung der dadurch generierten Lernempfehlung.

Die Alternativhypothese H_1 , die es zu widerlegen gilt, ist die Verschlechterung der Effizienz durch die Nutzung Graph-basierter Lernpfade.

Es wurden weitere Hypothesen bzw. operationale Hypothesen aufgestellt, die im Rahmen einer Operationalisierung belegt oder widerlegt werden sollen:

1. Durch Verwendung Graph-basierter Lernpfade werden Lerneinheiten mit einer impliziten Abhängigkeit zum Lernziel mit höherer Wahrscheinlichkeit empfohlen.
2. Die Berücksichtigung wichtiger Einflussfaktoren, wie die erreichte Punktzahl durch Kantengewichte, führt zu sinnvolleren Lernpfaden.
3. Die gemessene Progress Performance spiegelt sich in Stereotypen des Benutzers wider.
4. Je mehr Beobachtungsdaten vorhanden sind, desto effizientere Lernpfade können durch Adaptivität entstehen.

5.2 Verwendung der erhobenen Beobachtungsdaten (xAPI)

Die gesammelten Beobachtungsdaten und Lernerfahrungen aus Spot-X folgen dem xAPI-Standard und somit auch dem Triple-Prinzip. Diese Beobachtungsdaten sind in einem Learning Record Store (LRS) zugänglich. In dieser Arbeit wird die Operationalisierung an fiktiven Benutzern angewendet. In diesem Abschnitt wird festgelegt, welche Beobachtungsdaten aus dem LRS genutzt und welche potenziellen Einflussfaktoren operationalisiert werden sollen. Diese Einflussfaktoren, bestehend aus benutzerspezifischen und messbaren Beobachtungsdaten, sollen zur Anwendung des datengetriebenen GPS und GKT genutzt werden.

Die drei Hauptbestandteile eines xAPI-Statements (Akteur, Verb und Objekt) sollen zur Erzeugung der Lernpfade genutzt werden. In dieser Arbeit wird sich auf die wichtigsten xAPI-Objekte der gesammelten Spot-X Daten beschränkt. Dadurch soll es ermöglicht werden, die Auswirkungen an den Lernpfaden besser erkennbar zu machen. Ein Beispiel eines xAPI-Statements, nach Vollendung einer Multiple-Choice-Aufgabe, wird in Listing 5.1 aufgelistet. Dieses xAPI-Statement kann folgendermaßen interpretiert werden: „Ein Benutzer namens Alice hat am 01.06.2022 die Multiple-Choice-Aufgabe »1912« aus der Kategorie »Geschichte« mit 0 Punkten abgeschlossen.“

```
1 {
2   "timestamp": "2022-06-01T12:00:00.000Z",
3   "actor": {
4     "objectType": "Agent",
5     "mbox": "mailto:alice@user.de",
6     "name": "Alice"
7   },
8   "verb": {
9     "id": "http://adlnet.gov/expapi/verbs/completed",
10    "display": { "de": "Aufgabe abgeschlossen" }
11  },
12  "object": {
13    "id": "https://.../game/play/19",
14    "objectType": "Activity",
15    "definition": {
16      "type": "http://activitystrea.ms/schema/1.0/game",
17      "name": { "de": "Geschichte" },
18      "description": { "de": "Beantwortete die Multiple-Choice
19        -Aufgabe '1912' falsch." }
20    }
21  },
22  "context": {
23    "platform": "http://spotxapp.herokuapp.com/",
24    "extensions": {
25      "https://.../task/difficultyCategory": "mittel",
26      ...
27    }
28  },
29  "result": {
30    "success": false,
31    "duration": "PT4S",
32    "response": "1889",
33    "score": {
34      "scaled": 0,
35      "raw": 0,
36      "min": 0,
37      "max": 100
38    }
39  }
40 }
```

Listing 5.1: Gekürztes xAPI-Statement einer gespielten Aufgabe in Spot-X

Neben den xAPI-Statements abgeschlossener Aufgaben, sendet Spot-X auch Informationen gesamter Spieldurchläufe an das LRS. Auch feingranulare Interaktionen werden festgehalten, wie beispielsweise das Einloggen in das System oder das Anschauen des Ergebnisses eines Spieldurchlaufs. Diese Beobachtungsdaten können ebenfalls Hinweise auf wichtige Interaktionen sein, um passende Aktivitäten oder Lerneinheiten zu empfehlen. Tabelle 5.1 listet einige xAPI-Objekte auf, die im Rahmen der Nutzung von Spot-X gesendet werden.

Objekt-URI	Beschreibung
http://activitystrea.ms/schema/1.0/game	Beschreibt den Beginn oder die Vervollständigung eines Spieldurchlaufs, bestehend aus mehreren Aufgaben.
http://adlnet.gov/expapi/activities/question	Beschreibt eine einzelne Aufgabe, in Form einer Multiple-Choice, Freitext oder Wimmelbild-Aufgabe.
http://adlnet.gov/expapi/activities/course	Dient zur Beschreibung einer Benutzerinteraktion, z. B. eine Anmeldung auf einer Spot-X Plattform.
http://activitystrea.ms/schema/1.0/page	Dient zur Beschreibung feingranularer Benutzerinteraktionen, z. B. dass die Ergebnisübersicht angeschaut wurde.

Tabelle 5.1: Objekte der xAPI-Statements aus Spot-X

Zur Darstellung der Nutzungspfade sollen alle erhältlichen Beobachtungsdaten in den Graphen wiederspiegelt werden. Zur Darstellung und Eingrenzung der Lernpfade in Spot-X, werden die Aktivitäten eines Spielstarts, einer Aufgabenvollendung und das Ende eines Spieldurchlaufs verwendet. Um irrelevante Benutzerdaten herauszufiltern, wird der GPS erweitert, sodass beispielsweise das xAPI-Statement eines "Ausloggen" herausgefiltert werden kann. Die Lernpfade sind im Kontext dieser Operationalisierung Sequenzen von Lerneinheiten, in Form von Spot-X Aufgaben. Diese Filterung kann über Umgebungsvariablen für spätere Zwecke konfiguriert werden.

Das xAPI-Objekt eines Spielstarts bzw. Spielende indiziert im Kontext dieser Arbeit den Start- bzw. Endknoten eines Lernpfads (siehe erste Zeile in Tabelle 5.1).

Im Folgenden werden verschiedene Attribute aufgelistet, die aus einem xAPI-Statement einer abgeschlossenen Aufgabe entnommen werden können (vgl. Listing 5.1). Hierzu werden die Attribute und deren dazugehörigen Wertebereiche in Tabelle 5.2 aufgezeigt. Diese Attribute können zur Kantengewichtung der Lernpfade verwendet werden und stellen somit auch wichtige Einflussfaktoren der Operationalisierung dar.

Attribut	Wertebereich
Punktzahl	$p \in [0,100]$
Result Performance (scaled score)	$Perf^R \in [0,1]$
Schwierigkeitsgrad	$diff \in [0,1]$
Schwierigkeitskategorie	$SK \in \{einfach, mittel, schwer\}$
benötigte Zeit (time)	$time \in [0,60]$
Kategorien	Z.B. {"Mathe", "Geschichte"}

Tabelle 5.2: Wichtige Attribute und deren Wertebereiche im xAPI-Statement einer abgeschlossenen Spot-X Aufgabe

In dieser Arbeit wird zur Operationalisierung auf den Einfluss der erreichten Punktzahl fokussiert und wie sich dieser in adaptiven Lernpfaden auswirken kann.

5.3 Vorstellung des Beispielszenarios

Der verwendete GPS und das GKT treffen Entscheidungen auf Basis beobachteter Benutzerdaten. Zur Durchführung wird daher ein synthetisches Datenset verwendet, um ideale Pfade und Lernempfehlungen zu erzeugen. Für das Beispielszenario werden Spot-X Aufgaben entworfen und zusätzlich synthetische Benutzerdaten durch die Anwendung von Spot-X erzeugt. Das Datenset soll dazu genutzt werden, eine Operationalisierung durchzuführen. Hierbei werden Einflussfaktoren durch Verwendung verschiedener Methoden, beispielsweise zur Bestimmung idealer Pfade (IPs), operationalisiert und verglichen.

Das Aufgabenset besteht aus zehn Aufgaben der beiden Kategorien „Geschichte“ und „Mathe“. Dabei stellen die Geschichte-Aufgaben Jahreszahlen historischer Ereignisse und die Mathe-Aufgaben Rechenaufgaben dar. Abbildung 5.2 zeigt das Aufgabenset bestehend aus vier leichten, vier mittel-schweren und zwei schweren Aufgaben. Das Lernziel ist die richtige Beantwortung einer schweren Aufgabe, also eine Aufgabe der Schwierigkeitskategorie (SK) „schwer“. In dem betrachteten Beispiel gibt es demnach zwei potenzielle Lernziele: Aufgabe „44 v. Chr.“ und Aufgabe $(19+1)*32*2$ (siehe erste Zeile in Abbildung 5.2). Abbildung 5.3 zeigt beispielhaft die Aufgabe „1989“ aus dem Aufgabenset mit $SK(1989) = einfach$.

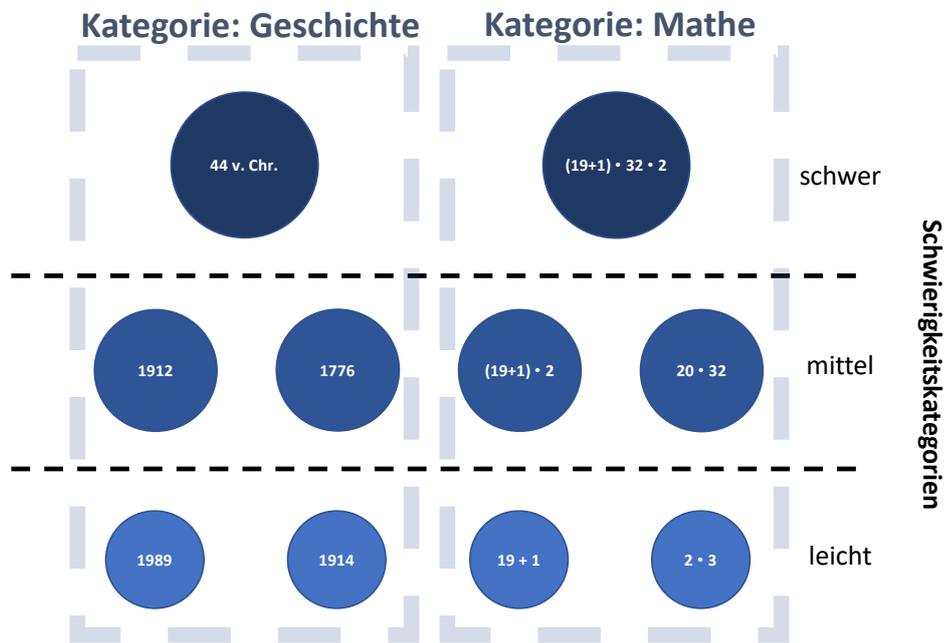


Abbildung 5.2: Aufgabenset des Beispielszenarios zur Erzeugung synthetischer Lernpfade



Abbildung 5.3: Aufgabe "1989" aus dem Aufgabenset des Beispielszenarios (Bildquelle Berliner Mauerfall „Bundesbildstelle - Bilddatenbank“ 2022)

Nachdem nun das Lernziel für Spot-X feststeht, werden darauf basierend synthetische Lernpfade erstellt. Basierend auf dem Aufgabenset werden drei fiktive Benutzer in Form von xAPI-Statements erzeugt. Diese Beobachtungsdaten stellen demonstrativ gesammelte Daten dar, in denen keine Adaption durch Lernempfehlungen stattfand. Bei der Erstellung der synthetischen Daten wird darauf geachtet, dass die Nutzungspfade Abhängigkeiten zwischen den Aufgaben aufzeigen. Diese Abhängigkeiten sollen im Laufe der Operationalisierung erkenntlich gemacht werden.

Tabelle 5.3 zeigt drei beispielhafte Stereotypen von Benutzern und deren Lernpfade. Ein Lernpfad stellt im Rahmen dieser Operationalisierung einen Spieldurchlauf in Spot-X dar. In der ersten Zeile der Tabelle 5.3 wird der Lernpfad von „Alice“ gezeigt. Darin wird die erste Aufgabe „1912“ der Kategorie Geschichte falsch beantwortet, was durch die rote Kante indiziert wird. Hierbei ist zu erwähnen, dass diese rote Kante aus dem xAPI-Statement aus Listing 5.1 erzeugt werden könnte.

Aus den Lernpfaden der Tabelle 5.3 ist zu entnehmen, dass eine schwere Aufgabe richtig beantwortet wird, sobald die darunter liegenden Aufgaben ebenfalls richtig beantwortet werden. Dies lässt sich besonders gut am Lernpfad von „Bob“ erkennen. Der Lernpfad lässt vermuten, dass „Bob“ Schwierigkeiten bei den Aufgaben der Kategorie Geschichte hat, wohingegen er alle Aufgaben der Kategorie Mathe richtig beantwortet.

Abbildung 5.4 zeigt die Vereinigung aller möglichen Pfade aus Tabelle 5.3. Dabei lassen sich bereits mögliche Kandidaten idealer Pfade erkennen, die zu einer Aufgabe der SK „schwer“ führen.

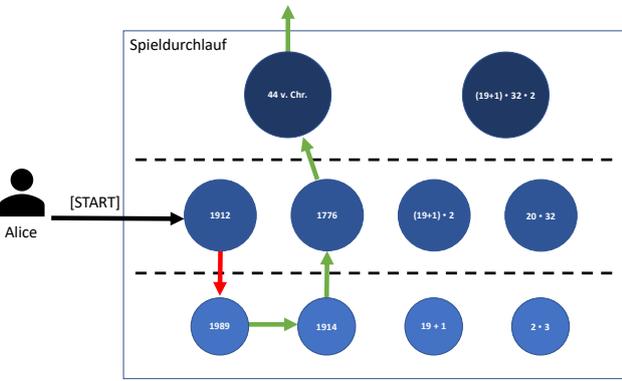
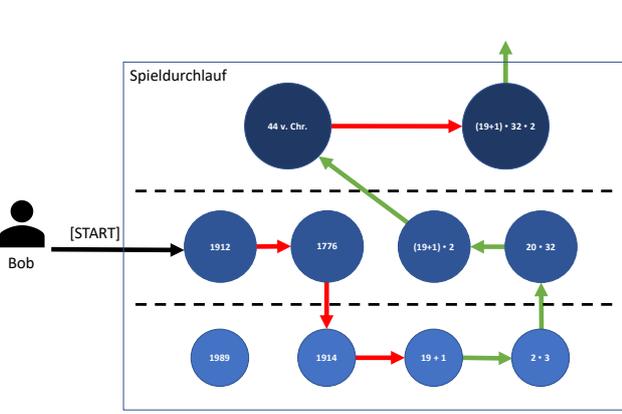
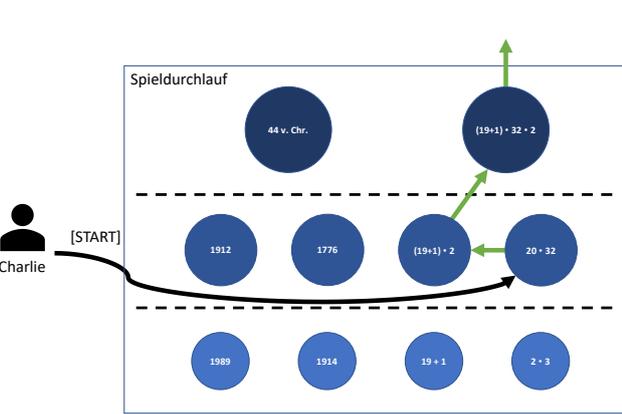
Lernpfad	Beschreibung
 <p>The diagram shows a grid of nodes representing tasks. Alice starts at node 1912. Her path is: 1912 → 1776 → 1914 → 44 v. Chr. → (19+1) * 32 * 2. Other nodes include 1989, 1912, 1776, (19+1) * 2, 20 * 32, 1989, 1914, 19 + 1, and 2 * 3.</p>	<p>Alice stellt eine durchschnittliche Benutzerin dar. Sie erreicht das Lernziel über einen kleinen Umweg, nachdem fünf Aufgaben gespielt wurden.</p>
 <p>The diagram shows the same grid of nodes. Bob starts at node 1912. His path is: 1912 → 1776 → 1914 → 19 + 1 → 2 * 3 → (19+1) * 2 → 44 v. Chr. → (19+1) * 32 * 2.</p>	<p>Bob stellt einen Anfänger dar. Er besitzt den längsten, ineffizientesten Lernpfad. Dennoch erreicht er nach neun gespielten Aufgaben das Lernziel.</p>
 <p>The diagram shows the same grid of nodes. Charlie starts at node 1912. His path is: 1912 → 20 * 32 → (19+1) * 2 → (19+1) * 32 * 2.</p>	<p>Charlie stellt einen Experten dar, wobei das Lernziel bereits nach drei Aufgaben erreicht wird.</p>

Tabelle 5.3: Stereotypen des Beispielszenarios und deren Lernpfade

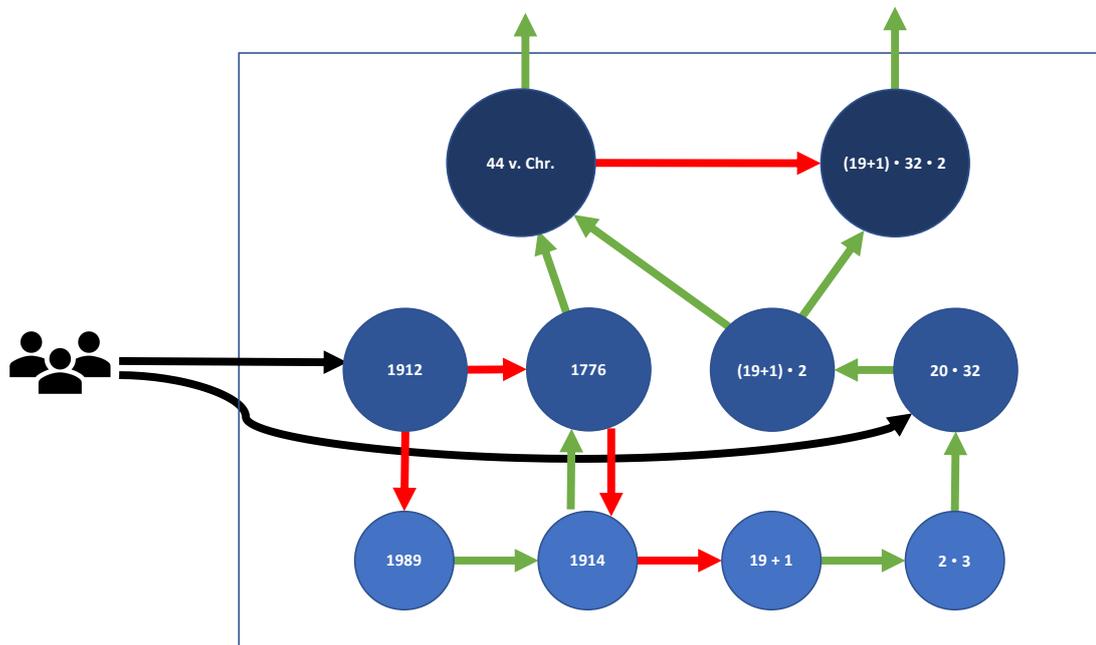


Abbildung 5.4: Vereinigung aller möglichen Pfade des Beispielszenarios aus Tabelle 5.3

5.4 Anwendung des Graph Performance Service (GPS)

In diesem Abschnitt wird die Operationalisierung Graph-basierter Lernpfade durch die Anwendung des GPS an Spot-X durchgeführt. Im Zwecke dieser Arbeit, wird die Implementierung zur Berechnung idealer Pfade und Performance-Messung angepasst und erweitert. Dabei werden verschiedene Einflussfaktoren operationalisiert, die in den Benutzerdaten aus Spot-X vorhanden sind. Die Auswirkungen werden anhand der draus resultierenden idealen Pfade und der gemessenen Performance verglichen. Zur Umsetzung werden verschiedene Algorithmen zur Berechnung kürzester Pfade verwendet, um beispielweise die Kantengewichte zu berücksichtigen.

Abbildung 5.5 soll das allgemeine Vorgehen des hier angewendeten GPS verdeutlichen. Hierzu werden verschiedene Indikatoren aus den Benutzerdaten zur Bestimmung idealer Pfade mitberücksichtigt. Deren Auswirkungen werden anhand der entstehenden Lernpfade und der $Perf^P$ gemessen, die vom GPS ausgegeben wird.

Zur Anwendung des GPS an Spot-X wird die Implementierung erweitert. Der Algorithmus zur Erzeugung des Interaktionsgraphen in Neo4j wurde angepasst, sodass alle wichtigen Informationen aus einem xAPI-Statement in den Pfaden enthalten sind. Dadurch soll es ermöglicht werden, die erreichte Punktzahl in die Berechnung kürzester Pfade einzubeziehen.

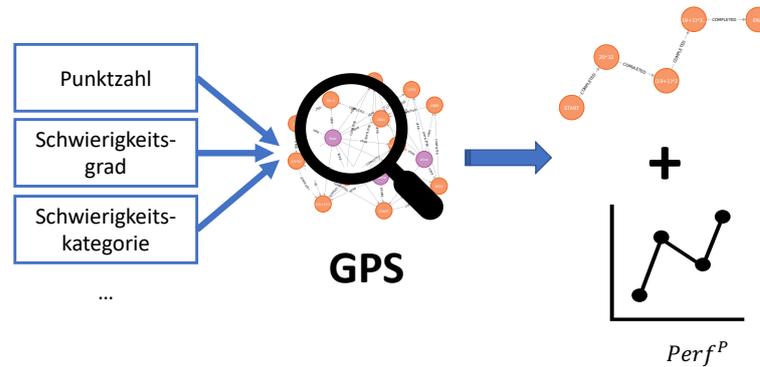


Abbildung 5.5: Anschauung der Operationalisierung Graph-basierter Lernpfade durch Anwendung des GPS

Listing 5.2 zeigt den überarbeiteten Algorithmus zur Erzeugung eines Nutzungspfades. Hierbei wird für den Benutzer mit der E-Mail “alice@user.de” ein chronologischer Nutzungspfad, ausgehend vom Start-Knoten, generiert. In Zeile 11 werden die Informationen eines xAPI-Verbs in die dazugehörige Kante vollständig übertragen.

```

1 MATCH (start:Activity {name: 'START'})
2 MATCH (user:Agent {userid: 'alice@user.de'})
3 WITH user
4 MATCH (activity:Activity) -[verb]- (user)
5 WITH user, verb, activity ORDER BY verb.time ASC
6 WITH COLLECT(activity) AS activities, COLLECT(verb) AS verbs
7 FOREACH (n in RANGE(0, size(activities) - 2) |
8     FOREACH (prev IN [activities[n]] |
9         FOREACH (next IN [activities[n + 1]] |
10             Merge (prev) -[c:COMPLETED]-> (next)
11             SET c = verbs[n]
12         )))
13 RETURN activities

```

Listing 5.2: Cypher Algorithmus zur Generierung eines benutzerspezifischen Nutzungspfades in Neo4j (bauend auf Implementierung von Baur (2022))

Da in Spot-X kein konkretes Lernziel in Form einer konkreten Aufgaben festgelegt wird, wurde die Schnittstelle zur Anfrage der nächsten Lernempfehlung überarbeitet. Hierzu wird die Pflichtangabe eines Lernziels entfernt und auf die Angabe einer Benutzer-ID reduziert. Mithilfe des “Context”-Parameters wird es ermöglicht, zusätzliche Benutzerinformationen oder Konfigurationen hinzuzufügen. Beispielweise kann hierdurch ein charakterisierendes Ziel durch ein Knoten-Attribut festgelegt werden:

“.../response?user=alice@example.com&context={'attr_name': 'category' 'attr_value': 'Mathe'}”

Zur Einführung eines charakterisierenden Lernziels, werden Knoten-Attribute verwendet, die aus dem Erweiterungs-Feld eines xAPI-Statements stammen können. Dafür wird die Implementierung erweitert, sodass die Angabe eines Lernziels in Form einer konkreten Aufgabe hinfällig ist. Das Lernziel soll stattdessen anhand der charakterisierenden Eigenschaften festgelegt werden. Dadurch kann bspw. das Ziel der Beantwortung einer schweren Aufgabe gesetzt werden. Demnach werden alle Aufgaben-Knoten A mit $SK(A) = schwer$ als potenzielle Lernziele für ideale Lernpfade betrachtet. Ein alternatives Lernziel könnte in einem SG z. B. das Erreichen eines Achievements sein. Steht ein Lernziel fest, werden pro Lernziel IPs bestimmt werden. Anschließend wird der IP mit der höchsten Ähnlichkeit zum aktuellen Lernpfad des Benutzers empfohlen. Im Folgenden werden die verschiedenen Berechnungsmöglichkeiten von IPs verglichen. Hierfür werden die Algorithmen von Yen und Dijkstra angewendet und verglichen. Abbildung 5.8 zeigt die IPs vom Start eines Spieldurchlaufs, unter Anwendung des Yen's k -Shortest-Paths Algorithmus, bestimmt wurden. Hierbei ist zu erkennen, dass IP_1^Y mit dem Lernpfad des Experten “Charlie” übereinstimmt. Der IP_2^Y zeigt einen gemeinsamen Pfad der Lernpfade von “Alice” und “Bob”.

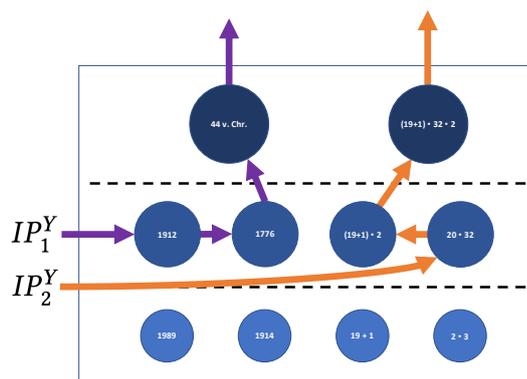


Abbildung 5.8: Berechnete IPs durch Anwendung des Yen's k -Shortest-Paths Algorithmus

Das Problem der Verwendung des Algorithmus von Yen ist, dass hier ausschließlich auf die Kantenzahl und nicht auf Kantengewichte geachtet wird. Hierdurch wird eine falsch beantwortete Aufgabe mit einer richtig beantworteten Aufgabe gleichgesetzt. Im Folgenden wird unter Berücksichtigung der erreichten Punkte, in Form von Kantengewichten, IPs bestimmt. Dabei soll der kürzeste Pfad mit der höchsten Summe der Punktzahlen bestimmt werden. Die Anwendung des Dijkstra's Shortest-Path Algorithmus mittels der normierten Punktzahlen als Kantengewichte, würde zwar den kürzesten Pfad bzgl. der Kantenzahl berechnen, jedoch den Pfad mit der geringsten kumulierten Summe Punkte bestimmen. Der Algorithmus der dieses Problem löst wird in Listing 5.3 dargestellt. Dieser negiert zunächst die Kantengewichte (Punktzahl) und bestimmt anschließend den kürzesten Pfad anhand der negierten Punktzahl.

```

1 MATCH ()-[k:COMPLETED]->()
2 SET k._score = -k.score
3 WITH k
4 MATCH (startNode {name: '$start'}), (endNode {name: '$goal'})
5 CALL apoc.algo.dijkstra(startNode, endNode, 'COMPLETED', '_score')
6 YIELD path as nodeIds, weight AS distance
7 RETURN nodeIds, distance

```

Listing 5.3: Cypher zur Berechnung idealer Pfade mit Dijkstra's Shortest-Path Algorithmus

Der resultierende IP_1^D unter Anwendung des Algorithmus aus Listing 5.3 wird in Abbildung 5.9 dargestellt. Hierbei ist zu erkennen, dass nur der Pfad des Experten "Charlie" zurückgegeben wird. Es wird demnach kein zweiter IP berechnet, da der orangene IP aus Abbildung 5.8 eine rote Kante besitzt (vergleiche Abbildung 5.4).

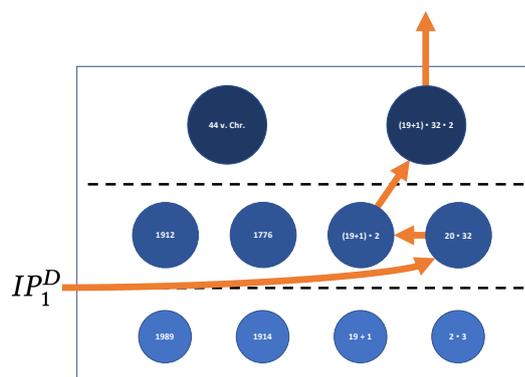


Abbildung 5.9: Berechneter IP_1^D durch Anwendung des Dijkstra's Kürzester-Pfad Algorithmus mit negierten Gewichten

5.5 Anwendung des Graph-based Knowledge Tracing (GKT)

In diesem Abschnitt wird das GKT auf das Beispielszenario bzw. die synthetischen xAPI-Daten angewendet. Ziel ist es, mit dem GKT Lernempfehlungen mittels einer Performance-Vorhersage zu treffen. Wie in Abschnitt 3.10 erwähnt, meint hier die Performance nicht die Lerneffizienz, sondern die Wahrscheinlichkeit, dass eine Aufgabe richtig beantwortet wird. Die Performance-Vorhersagen des GKT sollen in dieser Arbeit implizit zu einem Lernpfad zusammengefasst werden, um diesen mit dem Ergebnis des GPS zu vergleichen. Dazu wird die Lernempfehlung mit der höchsten Performance-Vorhersage als Lernempfehlung gewählt. Zusätzlich ist das GKT in der Lage, Wissensstrukturen und Abhängigkeiten der Konzepte zu generieren. Ein Wissenskonzept soll im Kontext von Spot-X mit einer Aufgabe gleichgesetzt werden.

Da zur Performance-Vorhersage des GKT eine Liste der Aufgaben in Form eindeutiger Skill-IDs übergeben wird, wird in Abbildung 5.10 das ursprüngliche Beispielszenario mit den dazugehörigen Skill-IDs zur besseren Übersicht dargestellt. Dabei wurde das Beispielszenario nicht verändert, es wurden hier lediglich die Aufgabennamen durch die vom GKT generierten Skill-IDs ersetzt.

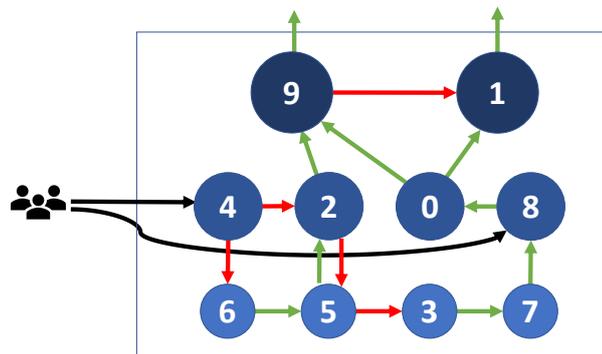


Abbildung 5.10: Graph des Beispielszenarios mit zugeordneten Skill-IDs für das GKT

Bei der Operationalisierung unter Anwendung des GKT wird gezeigt, dass nicht alle Beobachtungsdaten für die Berechnung der Graphen miteinbezogen werden sollten. Zunächst wurden zur Durchführung der Operationalisierung, die unveränderten Spot-X Beobachtungsdaten in das GKT als Eingabe in Form einer CSV-Datei eingegeben (siehe Abbildung 5.4). Dabei ist zu erkennen, dass beispielweise auch der Zeitstempel als Eingabe berücksichtigt wird.

Die Ergebnisse der Performance-Vorhersagen der vollständigen Beobachtungsdaten führen zu irreführenden Performance-Schätzungen. Dies lässt vermuten, dass dadurch Faktoren, wie der Zeitstempel berücksichtigt wurde, um den Graphen zu generieren. Das Ergebnis einer beispielhaften Eingabe in Form einer Sequenz von Skill-IDs $P_U = \{4,6,5,2,1\}$, soll einen

timestamp	stater Image	https://www.	user_id	correct	statements/result/duration	statements/result/response	
2022-09-29T20:37:05.422Z	69	http://127.0.0.1:30	normal	Alice	False	PT4S	12(wrong)
2022-09-29T20:37:08.429Z	61	http://127.0.0.1:30	easy	Alice	True	PT1S	7(correct)
2022-09-29T20:37:11.429Z	62	http://127.0.0.1:30	easy	Alice	True	PT1S	12(correct)
2022-09-29T20:37:14.505Z	68	http://127.0.0.1:30	normal	Alice	True	PT1S	84(correct)
2022-09-29T20:37:18.062Z	63	http://127.0.0.1:30	hard	Alice	True	PT1S	84(correct)
2022-09-29T20:40:28.493Z	69	http://127.0.0.1:30	normal	Bob	False	PT4S	19(wrong)

Tabelle 5.4: Ausschnitt der ungefilterten Beobachtungsdaten zur Eingabe des GKT

Geschichtslehrer darstellen. Dabei wird festgelegt, dass er die Geschichtsaufgaben $\{4,6,5,2\}$ richtig beantwortet. Bei der letzten Skill-ID 1, handelt es sich um eine schwere Mathe-Aufgabe. An ihr soll geschätzt werden, wie wahrscheinlich der Geschichtslehrer die Aufgabe richtig beantworten wird. Das GKT berechnet auf Grundlage der Eingabe-Daten aus Abbildung 5.4 und in Kombination der latenten Wissensstruktur (Dense), folgende Performance-Schätzungen: $\{0.6608, 0.8637, 0.9210, 0.9432\}$. Die Ausgabe ist so zu interpretieren, dass die Wahrscheinlichkeit der richtigen Beantwortung kontinuierlich steigt. Die Wahrscheinlichkeit 66,08% deutet darauf hin, dass die Aufgabe 6 nach der richtigen Beantwortung der Aufgabe 4 ebenfalls richtig beantwortet wird. Die letzte Performance-Vorhersage gibt wider, mit welcher Wahrscheinlichkeit die schwere Mathe-Aufgabe richtig beantwortet wird.

Für die weitere Anwendung des GKT wurden daher die Eingabedaten nach den wichtigsten Einflussfaktoren gefiltert. Abbildung 5.5 zeigt die gefilterte Version der Eingabe für das GKT.

Zur Generierung einer latenten Wissensstruktur in Form eines Graphen, der zur Anwendung des GKT benötigt wird, wird im Folgenden der Dense-, der Transition und der VAE-Graph verwendet. Nachdem die xAPI-Daten in das CSV Dateiformat zur Eingabe in das GKT übergeben werden, werden mithilfe der jeweiligen Methode die Abhängigkeiten zwischen den Aufgaben hergeleitet. Dabei wird jeder Aufgabe eine individuelle Skill-ID zugewiesen. Diese Skill-ID kann für eine Performance-Vorhersage im nächsten Schritt als Eingabe in Form einer Matrix $E_c \in \mathbb{R}^{N \times e}$ dargestellt werden. Die Matrix umfasst alle bisher beantworteten Skill-IDs und die dazugehörigen Antworten zu einem bestimmten Zeitschritt. Zur Berechnung der Performance auf einem Pfad wird daher eine einzeilige Matrix übergeben. Diese Matrix weist allen Skill-IDs eine dazugehörige Antwort, in Form einer binären 1 (für richtig) oder 0 (für falsch) zu. In der Implementierung findet das in Form von zwei Arrays statt. Das Array *question_list* beinhaltet die chronologische Sequenz der Skill-IDs, für die eine Performance-Vorhersage auf Basis von G_{VAE} berechnet wird. Der Wert an Index i in dem zusätzlichen Array *feature_list* gibt an, ob die Skill-ID an *question_list*[i] richtig (1) oder falsch (0) beantwortet wurde. Tabelle 5.6 zeigt ergänzte Benutzer, zur sinnhaften Beschreibung der Ergebnisse des GKT beitragen sollen. Die Tabelle zeigt die Bezeichnungen der Benutzer in Kombination einer kurzen Beschreibung.

skill_id	Name	categories	SK	user_id	correct	feature
4	1912	Geschichte	normal	alice@user.de	False	0
6	1989	Geschichte	easy	alice@user.de	True	1
5	1914	Geschichte	easy	alice@user.de	True	1
2	1776	Geschichte	normal	alice@user.de	True	1
9	44 v. Chr.	Geschichte	hard	alice@user.de	True	1
4	1912	Geschichte	normal	bob@user.de	False	0
2	1776	Geschichte	normal	bob@user.de	False	0
5	1914	Geschichte	easy	bob@user.de	False	0
3	19+1	Mathe	easy	bob@user.de	True	1
7	2*3	Mathe	easy	bob@user.de	True	1
8	20*32	Mathe	normal	bob@user.de	True	1
0	(19+1)*2	Mathe	normal	bob@user.de	True	1
9	44 v. Chr.	Geschichte	hard	bob@user.de	False	0
1	(19+1)*32*2	Mathe	hard	bob@user.de	True	1
8	20*32	Mathe	normal	charlie@user.de	True	1
0	(19+1)*2	Mathe	normal	charlie@user.de	True	1
1	(19+1)*32*2	Mathe	hard	charlie@user.de	True	1

Tabelle 5.5: Gefilterte Eingabe der xAPI-Daten für die Anwendung des GKT

Diese werden im Weiteren als Eingabe für das GKT verwendet, um Performance-Vorhersagen zu treffen. Die Ergebnisse zeigen, dass die Verwendung des Dense-Graphen zeigt die besten Ergebnisse und ist neben dem Transition-Graph und VAE-Graph in Tabelle 5.7 aufgelistet.

Benutzer	Beschreibung
A	Identischer Lernpfad von Bob aus dem Beispielszenario.
B	Geschichtsexperte, der nur Geschichts-Aufgaben spielt.
C	Geschichtsexperte, der alle Geschichtsaufgaben richtig löst und am Ende eine schwere Mathe-Aufgabe richtig beantworten soll.
D	Mathe-Experte, der am Ende eine schwere Geschichts-Aufgabe richtig beantworten soll.
E	Benutzer, der Geschichts-Aufgaben falsch beantwortet und am Ende eine schwere Geschichts-Aufgabe beantworten soll.

Tabelle 5.6: Ergänzte Stereotypen zur Anwendung des GKT

Dense			
Benutzer	question_list	feature_list	Performance-Vorhersagen
A	4,2,5,3,7,8,0	0,0,0,1,1,1,1	0.7975, 0.9505, 0.6256, 0.5042, 0.5441, 0.7001
B	4,6,5,2,9	1,1,1,1,1	0.7504,0.9557,0.9634,0.9660
C	4,6,5,2,1	1,1,1,1,1	0.7504,0.9557,0.9634,0.4808
D	8,0,1,9	1,1,1,1	0.5481,0.5546,0.9592
E	4,2,5	0,0,1	0.7975,0.9505

Transition			
Benutzer	question_list	feature_list	Performance-Vorhersagen
A	4,2,5,3,7,8,0	0,0,0,1,1,1,1	0.5721,0.7612,0.6348,0.6837,0.6609,0.6984
B	4,6,5,2,9	1,1,1,1,1	0.9488,0.9154,0.9292,0.8984
C	4,6,5,2,1	1,1,1,1,1	0.9488,0.9154,0.9292,0.6825
D	8,0,1,9	1,1,1,1	0.5861,0.6309,0.6576
E	4,2,5	0,0,1	0.5721,0.7612

VAE			
Benutzer	question_list	feature_list	Performance-Vorhersagen
A	4,2,5,3,7,8,0	0,0,0,1,1,1,1	0.5576,0.5429,0.5342,0.5384,0.5658,0.5675
B	4,6,5,2,9	1,1,1,1,1	0.5568,0.5556,0.5699,0.5939
C	4,6,5,2,1	1,1,1,1,1	0.5568,0.5556,0.5699,0.5563
D	8,0,1,9	1,1,1,1	0.5315,0.5436,0.6059
E	4,2,5	0,0,1	0.5576,0.5429

Tabelle 5.7: Vergleich der berechneten Performance-Vorhersagen, unter Verwendung verschiedener Graph-Typen des GKT

Abbildung 5.11 ergänzt das Ergebnis für den Benutzer C, der einen Geschichtsexperten darstellt, mit einem Diagramm seiner Performance-Vorhersagen $Perf^R$ für jeden Zeitschritt t . Aus dem Ergebnis ist zu erkennen, dass er die letzte Aufgabe mit einer relativ niedrigen Wahrscheinlichkeit von 48,08% beantworten wird. Wobei es sich bei der letzte Aufgabe um eine Mathe-Aufgabe der SK schwer handelt.

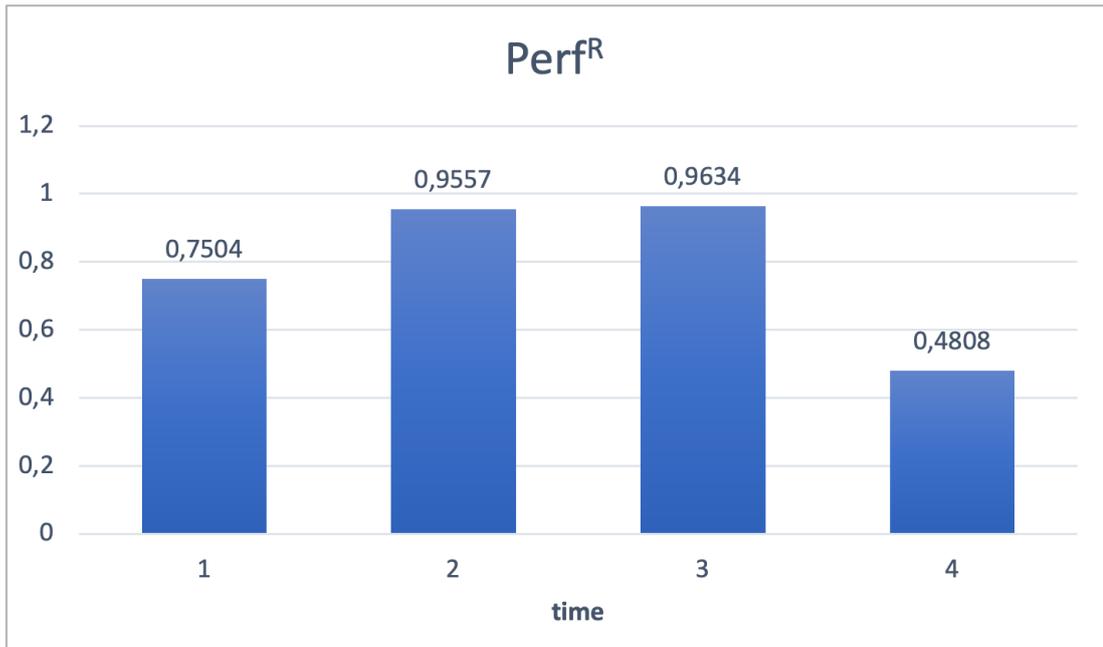


Abbildung 5.11: Performance-Vorhersage $Perf^R$ von Benutzer C zum Zeitschritt t

6 Technische Verifikation & Diskussion

In diesem Kapitel wird die Durchführung der Operationalisierung abgeschlossen. Anhand eines neuen fiktiven Benutzers in Spot-X werden adaptive Lernpfade angewendet, um Lernempfehlungen auszuwählen. Das Beispielszenario aus Abschnitt 5.3 wird hierfür durch einen neuen fiktiven Benutzer "Zoe" mit einem unvollständigen Pfad ergänzt. Für Zoe soll hiermit eine Steigerung ihrer Lern-Effizienz gemessen werden. Hierfür wird eine Progress Performance $Perf^P \in [0,1]$ berechnet und eine Lernempfehlung auf Basis des Beispiels bestimmt.

6.1 Anwendung in einem Nutzungsszenario

Abbildung 6.1 zeigt den aktuellen Pfad von Zoe, wobei die Adaptivität erst ab der zweiten Aufgabe verwendet wird. Hierbei soll die Adaptivität während eines Spieldurchlaufs in Spot-X durch eine geeignete Aufgabenempfehlungen gelenkt werden.

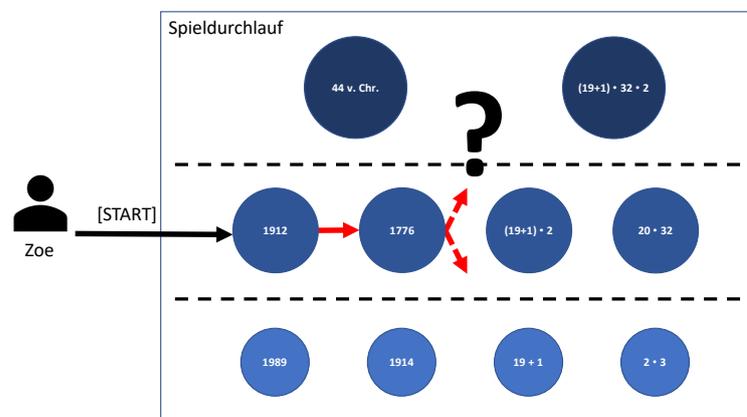


Abbildung 6.1: Unvollständiger Lernpfad eines neuen Benutzers

Im Folgenden wird untersucht, welche Aufgaben-Empfehlung der GPS für das betrachtete Szenario auswirft.

Die Ausgabe des GPS unter Verwendung des Dijkstra's Shortest-Path, der die negierten Punktzahlen erhält, wird für Zoe folgende Ausgabe gegeben:

```

1  {
2  "success": true,
3  "performance": 0.42,
4  "recommendation": {
5    "next_activity": {
6      "name": "44 v. Chr.",
7      "id": "https://www.spotx.de/spotx/game/play/63"
8    }
9  }
10 }
```

Listing 6.1: GPS-Ausgabe für Zoe unter Berücksichtigung der Punktzahl

In Listing 6.1 ist zu erkennen, dass die Aufgabe “44 v. Chr.” empfohlen wird und eine $Perf^P = 0.42$ gemessen wird. Dies wird dadurch begründet, dass sie sich unter Anwendung des Dijkstra's Shortest-Path nicht auf einem IP befindet (vgl. Abbildung 5.9).

Im Folgenden wird das zu erreichende Lernziel konfiguriert und durch eine alternative charakterisierende Eigenschaft ersetzt. Dabei soll das Ziel die richtige Beantwortung einer mittel-schweren Mathe-Aufgabe sein. Die dafür benötigte Anfrage lautet:

```
“.../response?user=zoe@example.com&context={'attr_name': 'category' 'attr_value': 'Mathe',
'SK': 'mittel'}”.
```

Listing 6.2 zeigt die daraus resultierende Ausgabe des GPS, wobei die empfohlene Aufgabe in Richtung IP_1^D führt.

```

1  {
2  "success": true,
3  "performance": 0.19,
4  "recommendation": {
5    "next_activity": {
6      "name": "1989",
7      "id": "https://www.spotx.de/spotx/game/play/61"
8    }
9  }
10 }
```

Listing 6.2: GPS-Ausgabe für Zoe unter Berücksichtigung der Punktzahl zur Erreichung einer Mathe-Aufgabe

Abschließend werden in Tabelle 6.1 die unterschiedlichen Lernziele zusammengefasst, die durch ein Knoten-Attribut festgelegt werden können. Hierbei wird die Lernempfehlung und die gemessene Performance auf den jeweils berechneten IPs bezogen und angegeben.

Berücksichtigte Einflussfaktoren	$Perf^P$	Lernempfehlung
Kantenlänge	0.42	“44 v. Chr.”
Erreichte Punktzahl	0.08	“44 v. Chr.”
Erreiche Kategorie “Mathe” mit SK mittel	0.19	“1989”

Tabelle 6.1: Einfluss der charakterisierenden Ziele auf die Performance und die Lernempfehlung

Zum Vergleich wird nun das GKT zur Generierung einer Aufgabenempfehlung verwendet. Hierzu wird aufgrund der erarbeiteten Ergebnisse (siehe Abschnitt 5.5) der Dense-Graph verwendet. Auf Basis dieses generierten Graphs wird eine Performance-Vorhersage für die bisher ungespielten Aufgaben von Zoe getroffen. Daraufhin soll die Aufgabe mit der höchsten Performance-Vorhersage empfohlen werden. Hierfür wird der aktuelle Lernpfad von Zoe als Liste der gespielten Aufgaben-IDs und die dazugehörige Liste übergeben, die angibt, ob eine Aufgabe richtig beantwortet wurde. Anschließend wird für jede nicht gespielte Aufgaben-ID eine Performance-Vorhersage auf Basis der bereits gespielten Aufgaben von Zoe berechnet. Um zu prüfen, ob die Aufgabe “1914” empfohlen werden sollte, wird die Liste der IDs für die Aufgaben {“1912”, “1776”, “1914”} mit der Feature-Liste {0,0,1} übergeben. Ein Teil der resultierenden Performance-Vorhersagen ist in Abbildung 6.2 dargestellt. Demnach wird die Aufgabe “1914”, mit einer Performance-Vorhersage von 95,05%, empfohlen.

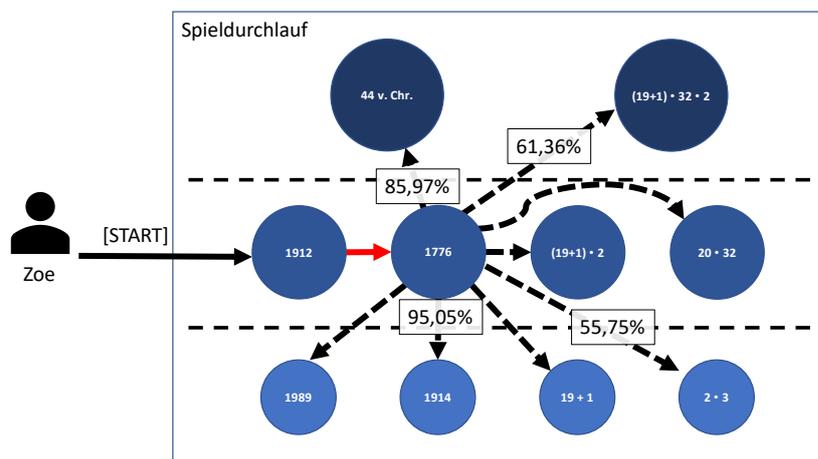


Abbildung 6.2: Performance-Vorhersagen des GKT

Zum Abschluss wird am Beispiel von Zoe gezeigt, dass sich durch die Anwendung adaptiver Lernpfade die Lern-Effizienz verbessern lässt. Betrachtet man den aktuellen Pfad von Zoe in Abbildung 6.1, lässt sich erkennen, dass es sich bei diesem Pfad um eine Schnittmenge des Lernpfads von Bob aus dem Beispielszenario handelt (vgl. Abbildung 5.2). Da angenommen wurde, dass der Lernpfad von Bob ohne eine Adaption erzeugt wurde, wird nun die Steigerung der Effizienz durch Anwendung des GPS mit dem Ergebnis aus Listing 6.1 verglichen. Durch Anwendung der Adaption erhält Zoe den Lernpfad mit der Aufgabensequenz {"1912", "1776", "44 v. Chr."} und erreicht bereits nach drei Aufgaben das Lernziel. Bob hingegen erreicht das Lernziel erst nach Vollendung von neun Aufgaben. Hierdurch ist zu erkennen, dass durch die Adaption des Lernpfads von Zoe eine Steigerung der Effizienz um über 66% erreicht wird.

6.2 Bewertung der Hypothesen

Am Beispiel fiktiver Benutzer, in Form von Stereotypen, kann gezeigt werden, dass durch die Adaption der Lernpfade die Erreichung eines davor spezifizierten Lernziels effizienter erreicht werden kann. Anhand des Beispiels von Zoe aus Abschnitt 6.1 wird gezeigt, dass die Anwendung einer Lernempfehlung die Effizienz potenziell steigern kann. Allerdings ist hier zu beachten, dass die Wahl der Einflussfaktoren bzw. der verwendete Benutzerdaten einen hohen Einfluss auf die erzeugten Lernempfehlungen haben. Anhand der niedrigen $Perf^P$ von Zoe ist erkennbar, dass sich ihr Stereotyp in der gemessenen $Perf^P$ widerspiegelt. Dies lässt sich damit begründen, dass ihr Lernpfad dem vom ineffizientesten Benutzer "Bob" aus dem Beispielszenario ähnelt. Die Anwendung des GKT zeigt zusätzlich, dass die Graph-basierten Methoden Abhängigkeiten zwischen den Aufgaben erkennen, um so relevantere Aufgaben mit höheren Abhängigkeiten vorschlagen zu können. Allerdings wird bei der Anwendung des GKT besonders deutlich, dass das Einbeziehen aller verfügbaren Beobachtungsdaten nicht sinnvoll ist. Daher müssen vor der Anwendung eines datengetriebenen Ansatzes zunächst irrelevante Daten aus den Beobachtungsdaten herausgefiltert werden. Des Weiteren wird gezeigt, dass die Anzahl der existierenden Lernpfade zu effizienteren Lernpfaden im Kontext von Spot-X führen.

6.3 Diskussion

Für das Quiz-Spiel Spot-X wurden verschiedene Ansätze umgesetzt, um adaptive Lernpfade zu generieren. Dabei können verschiedene Einflussfaktoren zur Berechnung von IPs mit einbezogen werden. Durch Anwendungen des GPS und GKT wird gezeigt, dass sich diese anhand ihrer Eingabe und Ausgabe vergleichen lassen, jedoch zu unterschiedlichen Ergebnissen führen können. Dabei ist zu berücksichtigen, dass die Verfahren jeweils unterschiedliche Modellierungsansätze der Graphen aufweisen. Der GPS generiert Lernpfade auf Grundlage der eingegebenen Beobachtungspfade, worauf anschließend Graphen Operationen durchgeführt werden. Das GKT hingegen berechnet basierend auf den Beobachtungsdaten ein Wissensstruktur in Form eines Graphen, die die Abhängigkeiten zwischen den Aufgaben abbildet. Das GKT dient in erster Linie zur Vorhersage einer Result Performance und nicht wie der GPS, um die Lern-Effizienz zu messen. Auf Basis dieses Graphen, bestehend aus den Aufgaben-Knoten und deren Abhängigkeiten in Form der Kanten, werden Performance-Vorhersagen getroffen. Dabei werden beantwortete Aufgaben mittels einer Matrix eingegeben. Das Problem beider Verfahren ist die Voraussetzung vieler Benutzerdaten, um dadurch bessere Schlussfolgerungen ziehen zu können. Dennoch kann bereits mit dem hier betrachtete Beispielszenario gezeigt werden, dass sich die Lernempfehlungen durch nur wenige Justierungen beeinflussen lassen.

7 Fazit und Ausblick

In dieser Arbeit wird gezeigt, wie sich Graph-basierte Lernpfade effektiv auf das Beispiel des SG "Spot-X" anwenden lässt. Hierbei wird eine kontrollierte Adaption mittels Graph-basierter Lernpfade durchgeführt. Durch die Operationalisierung werden Unterschiede und Gemeinsamkeiten zwischen dem Graph Performance Service (GPS) und dem etablierten Ansatz "Graph-based Knowledge Tracing" (GKT) gezeigt. Beiden Verfahren zeigen Ähnlichkeiten in der Generierung von Lernempfehlungen. Dabei ist jedoch zu beachten, welche Abwandlung der jeweiligen Methode verwendet wird. Für den GPS und das GKT werden verschiedene Konfigurationen der Verfahren gezeigt und verglichen. Auf Grundlage eines synthetischen Datensets, bestehend aus fiktiven Spot-X Benutzern, wird eine Operationalisierung durchgeführt. Bei der Konzeption dieser Arbeit wird gezeigt, dass sich anhand von charakterisierenden Eigenschaften von Lerneinheiten, ein Lernziel definieren lässt. Somit wird keine Festlegung eines Ziels in Form einer konkreten Lerneinheit benötigt, sondern kann z. B. durch ein abstrakteres Ziel, wie die Beantwortung eines bestimmten Themengebiets, festgelegt werden. Die Operationalisierung zeigt, dass es bei beiden Verfahren unterschiedliche Einflussfaktoren in den Eingabedaten in Form der Benutzerdaten gibt. Diese führen bereits bei einer kleinen Justierung zu Änderungen der resultierenden adaptiven Lernpfade. Bei der Anwendung des GPS wird zudem gezeigt, dass die verwendete Methode zur Berechnung der idealen Pfade maßgeblich zur Bestimmung von Lernempfehlungen ist. Dabei lässt sich verdeutlichen, dass die Kantenlänge eines Lernpfads nicht ausreicht, um sinnvolle ideale Pfade zu bestimmen. Die Betrachtung der erreichten Punktzahl als Kantengewicht verdeutlicht die Vorteile und den Einfluss auf die IPs. Durch Anwendung des GKT wird klar, dass auch die richtige Wahl der Eingabedaten in Form gefilterter Benutzerdaten maßgeblich ist, um sinnvolle Ergebnisse durch den datengetriebenen Ansatz zu erzielen.

In dem Vergleich der beiden Methoden, wird aufgezeigt, wie sich die Modellierungen und die Ausgaben der Verfahren unterscheiden. Das GPS generiert Graphen, die Lernpfade widerspiegeln und berechnet basierend darauf eine Progress Performance $Perf^P$. Das GKT hingegen erzeugt keine konkreten Lernpfade, sondern generiert zunächst einen Graph, der eine Wissensstruktur in Form von Abhängigkeitsbeziehungen der verschiedenen Lerneinheiten darstellt. Basierend auf dem Graph, sagt das GKT eine Result Performance $Perf^R$ vorher, die ange-

ben soll, wie wahrscheinlich eine bestimmte Lerneinheit richtig beantwortet wird. Trotz der Unterschiede der beiden Modellierungsansätze wird gezeigt, dass sich die resultierenden Lernempfehlungen ähneln. Im Rahmen dieser Arbeit wird nicht nur gezeigt, wie die Lernpfade durch verschiedene Indikatoren beeinflusst werden, zusätzlich werden die Auswirkungen einer generierten Wissensstruktur anhand des GKT verdeutlicht. Darüber hinaus wird bei der Anwendung des GPS an einem Beispielszenario verdeutlicht, dass sich die Stereotypen in der gemessenen $Perf^P$ widerspiegeln.

In weiterführenden Arbeiten kann die gemessene $Perf^P$ als Indikator eines schwachen Benutzers verwendet werden, um ihn mithilfe eines digitalen Assistenten zu unterstützen. Somit können ineffizienten Benutzern unterstützt werden, indem beispielsweise eine Adaption der Schwierigkeit empfohlen wird. In Zukunft kann ebenso untersucht werden, wie sich die Motivation durch die Empfehlung effizientester Lernpfade beeinflussen lässt. Hierbei kann das abstrakte Lernziel der Erreichung eines beliebigen Achievements festgelegt werden. Somit kann der Spieler regelmäßig an kleineren Erfolgserlebnissen in Form von Achievements profitieren. Hierzu kann ebenfalls untersucht werden, ob dadurch die Motivation des Spielers gesteigert werden kann.

Im Umfang dieser Arbeit wird nicht gezeigt, ob sich durch die Anwendung adaptiver Lernpfade, die Lern-Effektivität steigern lässt. Hauptfokus dieser Arbeit ist es, die Effizienz der Erreichung eines Lernziels zu erhöhen. Daher sollte im Anschluss dieser erarbeiteten Ergebnisse eine Nutzerstudie durchgeführt werden, um eine Korrelation hinsichtlich der Effektivität messen zu können. Zudem können dadurch zusätzlichen Beobachtungsdaten die Vorteile der Erkenntnisgewinne durch Anwendung der datengetriebenen Ansätze verstärkt werden. Alternativ können qualitative Einflussfaktoren eines Benutzers zur Operationalisierung miteinbezogen werden. Dies kann mittels einer qualitativen Befragung, z. B. in Form eines Dialogfensters nach einem Spieldurchlauf stattfinden, um so ggf. eine Lern-Effektivität zu operationalisieren.

Literaturverzeichnis

- Albers, Sönke, Daniel Klapper, Udo Konradt, Achim Walter und Joachim Wolf, Hrsg. 2007. *Methodik der empirischen Forschung*. 2., überarbeitete und erweiterte Auflage. Wiesbaden: Gabler.
- Alonso-Fernández, Cristina, Antonio Calvo-Morata, Manuel Freire, Iván Martínez-Ortiz und Baltasar Fernández-Manjón. 2019. „Applications of data science to game learning analytics data: A systematic literature review“. *Computers & Education* 141 (November): 103612. Besucht am 15. September 2022. <https://linkinghub.elsevier.com/retrieve/pii/S0360131519301654>.
- Balderjahn, Ingo. 2003. „Validität : Konzept und Methoden“. *Wirtschaftswissenschaftliches Studium : WiSt ; Zeitschrift für Studium und Forschung*, Wirtschaftswissenschaftliches Studium : WiSt ; Zeitschrift für Studium und Forschung. - München : Beck, ISSN 0340-1650, ZDB-ID 120285-6. - Vol. 32.2003, 3, p. 130-135, 32 (3).
- Bamberg, Günter, und Franz Baur. 1998. *Statistik*. Oldenbourgs Lehr- und Handbücher der Wirtschafts- und Sozialwissenschaften. München Wien: Oldenbourg.
- Battaglia, Peter W., Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti u. a. 2018. *Relational inductive biases, deep learning, and graph networks*. Issue: arXiv:1806.01261, 17. Oktober 2018. arXiv: 1806.01261[cs,stat].
- Baur, Rafael. 2022. *Graph Analytics für Performance Messungen und adaptive Lernpfade*.
- Bortz, Jürgen, und Nicola Döring. 1995. „Hypothesengewinnung und Theoriebildung“. In *Forschungsmethoden und Evaluation*, herausgegeben von Jürgen Bortz und Nicola Döring, 327–367. Springer-Lehrbuch. Berlin, Heidelberg: Springer.

- Briskorn, Dirk. 2020. „Graphentheorie“. In *Operations Research: Eine (möglichst) natürlich-sprachige und detaillierte Einführung in Modelle und Verfahren*, herausgegeben von Dirk Briskorn, 291–401. Berlin, Heidelberg: Springer. Besucht am 18. Juni 2022. https://doi.org/10.1007/978-3-662-60783-1_5.
- Bröder, Arndt. 2011. *Versuchsplanung und experimentelles Praktikum*. Hogrefe Verlag.
- „Bundesbildstelle - Bilddatenbank“. 2022. Besucht am 25. September 2022. <https://www.bundesbildstelle.de/bpa/de/search/?yearfrom=&yearto=&query=00012815>.
- Churchill, Gilbert A. 1979. „A Paradigm for Developing Better Measures of Marketing Constructs“. *Journal of Marketing Research* 16 (1): 64–73. <https://www.jstor.org/stable/3150876>.
- Csikszentmihalyi, Mihaly. 2000. „FLOW: The Psychology of Optimal Experience“, 6.
- Diestel, Reinhard. 2017. *Graphentheorie*. 5. Auflage. Lehrbuch. Berlin; [Heidelberg]: Springer Spektrum.
- Dijkstra, E. W. 1959. „A Note on Two Problems in Connexion with Graphs“, 45:287–290.
- Döring, Nicola, und Jürgen Bortz. 2016. *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. 5. vollständig überarbeitete, aktualisierte und erweiterte Auflage. Unter Mitarbeit von Sandra Pöschl-Günther. Springer-Lehrbuch. Berlin Heidelberg: Springer.
- Dörner, Ralf, Stefan Göbel, Wolfgang Effelsberg und Josef Wiemeyer, Hrsg. 2016. *Serious Games*. Cham: Springer International Publishing. Besucht am 15. Juni 2022. <http://link.springer.com/10.1007/978-3-319-40612-1>.
- „Fraunhofer IOSB“. Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB. 2022. Besucht am 24. September 2022. <https://www.iosb.fraunhofer.de/>.
- Gajwani, Juhi, und Pinaki Chakraborty. 2021. „Students’ Performance Prediction Using Feature Selection and Supervised Machine Learning Algorithms“. In *International Conference on Innovative Computing and Communications*, herausgegeben von Deepak Gupta, Ashish Khanna, Siddhartha Bhattacharyya, Aboul Ella Hassanien, Sameer Anand und Ajay Jaiswal, 347–354. Advances in Intelligent Systems and Computing. Event-place: Singapore. Springer.

- Hager, Willi. 2004. *Testplanung zur statistischen Prüfung psychologischer Hypothesen: die Ableitung von Vorhersagen und die Kontrolle der Determinanten des statistischen Tests*. Göttingen ; Seattle: Hogrefe.
- Heiss, Raffael, und Jörg Matthes. 2020. „Operationalisierung und Messung“. In *Handbuch Methoden der Politikwissenschaft*, herausgegeben von Claudius Wagemann, Achim Goerres und Markus B. Siewert, 63–78. Wiesbaden: Springer Fachmedien. Besucht am 29. August 2022. https://doi.org/10.1007/978-3-658-16936-7_3.
- Huang, Xiaowen, Quan Fang, Shengsheng Qian, Jitao Sang, Yan Li und Changsheng Xu. 2019. „Explainable Interaction-driven User Modeling over Knowledge Graph for Sequential Recommendation“. In *Proceedings of the 27th ACM International Conference on Multimedia*, 548–556. MM '19: The 27th ACM International Conference on Multimedia. Nice France: ACM, 15. Oktober 2019. Besucht am 3. Oktober 2022. <https://dl.acm.org/doi/10.1145/3343031.3350893>.
- Ifenthaler, Dirk, und Clara Schumacher. 2016. „Learning Analytics im Hochschulkontext“. *WiSt - Wirtschaftswissenschaftliches Studium* 45 (4): 176–181. Besucht am 6. Juli 2022. <http://elibrary.vahlen.de/index.php?doi=10.15358/0340-1650-2016-4-176>.
- „Interoperabilität und Assistenzsysteme (IAS)“. Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB. 2022. Besucht am 24. September 2022. <https://www.iosb.fraunhofer.de/de/kompetenzen/systemtechnik/interoperabilitaet-assistenzsysteme.html>.
- jhljx. 2022. „GKT“. Original-date: 2020-09-18T14:47:51Z, 3. Juni 2022. Besucht am 3. Juni 2022. <https://github.com/jhljx/GKT>.
- Kickmeier-Rust, Michael D., Elke Mattheiss, Christina Steiner und Dietrich Albert. 2011. „A Psycho-Pedagogical Framework for Multi-Adaptive Educational Games“. *International Journal of Game-Based Learning (IJGBL)* 1 (1): 45–58. Besucht am 7. Juli 2022. <https://www.igi-global.com/article/psycho-pedagogical-framework-multi-adaptive/50556>.
- Kingma, Diederik P., und Max Welling. 2014. *Auto-Encoding Variational Bayes*, arXiv:1312.6114, 1. Mai 2014. Besucht am 2. Oktober 2022. arXiv: 1312.6114[cs,stat]. <http://arxiv.org/abs/1312.6114>.
- Kromrey, Helmut, Jochen Roose und Jörg Strübing. 2016. *Empirische Sozialforschung: Modelle und Methoden der standardisierten Datenerhebung und Datenauswertung*. UTB, 10. Oktober 2016.

- „Lost Earth 2307“. Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB. 2022. Besucht am 10. September 2022. <https://www.iosb.fraunhofer.de/de/projekte-produkte/lost-earth-2307.html>.
- Minn, Sein, Yi Yu, Michel C. Desmarais, Feida Zhu und Jill-Jënn Vie. 2018. „Deep Knowledge Tracing and Dynamic Student Classification for Knowledge Tracing“. In *2018 IEEE International Conference on Data Mining (ICDM)*, 1182–1187. 2018 IEEE International Conference on Data Mining (ICDM). November.
- Moosbrugger, Helfried, und Augustin Kelava, Hrsg. 2020. *Testtheorie und Fragebogenkonstruktion*. Berlin, Heidelberg: Springer Berlin Heidelberg. Besucht am 26. September 2022. <https://link.springer.com/10.1007/978-3-662-61532-4>.
- Nakagawa, Hiromi, Yusuke Iwasawa und Yutaka Matsuo. 2019. „Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network“. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 156–163. 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI). Oktober.
- Needham, Mark, und Amy E. Hodler. 2019. *Graph algorithms: practical examples in Apache Spark and Neo4j*. O’Reilly Media.
- „Neo4j“. Neo4j Graph Data Platform. 2022. Besucht am 4. Oktober 2022. <https://neo4j.com/>.
- Pavlik, Philip I., Hao Cen und Kenneth R. Koedinger. 2009. *Performance Factors Analysis – A New Alternative to Knowledge Tracing*. Besucht am 2. Juni 2022. <https://eric.ed.gov/?id=ED506305>.
- Piech, Chris, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas und Jascha Sohl-Dickstein. 2015. „Deep Knowledge Tracing“. In *Advances in Neural Information Processing Systems*, Bd. 28. Curran Associates, Inc. Besucht am 23. September 2022. <https://proceedings.neurips.cc/paper/2015/hash/bac9162b47c56fc8a4d2a519803d51b3-Abstract.html>.
- Ritterfeld, Ute, Michael Cody und Peter Vorderer. 2009. *Serious Games: Mechanisms and Effects*. Routledge, 10. September 2009.
- Shute, Valerie, und Diego Zapata-Rivera. 2012. „Adaptive Educational Systems“. *Adaptive Technologies for Training and Education* (1. Januar 2012): 7–27.

- Shute, Valerie J., und Diego Zapata-Rivera. 2007. „Adaptive Technologies“. *ETS Research Report Series* 2007 (1): i–34. Besucht am 23. September 2022. <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2333-8504.2007.tb02047.x>.
- Sinclair, Jeff. 2011. „Feedback control for exergames“, 313.
- Song, Xiangyu, Jianxin Li, Qi Lei, Wei Zhao, Yunliang Chen und Ajmal Mian. 2022. *Bi-CLKT: Bi-Graph Contrastive Learning based Knowledge Tracing*. Issue: arXiv:2201.09020, 22. Januar 2022. Besucht am 23. September 2022. arXiv: 2201.09020[cs]. <http://arxiv.org/abs/2201.09020>.
- Streicher, Alexander, und Florian Heberle. 2017. „Learning Progress and Learning Pathways“, 19.
- Streicher, Alexander, Sebastian Leidig und Wolfgang Roller. 2018. „Eye-Tracking for User Attention Evaluation in Adaptive Serious Games“. In *Lifelong Technology-Enhanced Learning*, herausgegeben von Viktoria Pammer-Schindler, Mar Pérez-Sanagustín, Hendrik Drachler, Raymond Elferink und Maren Scheffel, 583–586. Lecture Notes in Computer Science. Event-place: Cham. Springer International Publishing.
- Streicher, Alexander, Rainer Schönbein und Stefan Pickl. 2021. „Graph-Based Modeling for Adaptive Control in Assistance Systems“. In *Advances in Artificial Intelligence, Software and Systems Engineering*, herausgegeben von Tareq Z. Ahram, Waldemar Karwowski und Jay Kalra, 39–46. Lecture Notes in Networks and Systems. Event-place: Cham. Springer International Publishing.
- Streicher, Alexander, und Jan D. Smeddinck. 2016. „Personalized and Adaptive Serious Games“. In *Entertainment Computing and Serious Games: International GI-Dagstuhl Seminar 15283, Dagstuhl Castle, Germany, July 5-10, 2015, Revised Selected Papers*, herausgegeben von Ralf Dörner, Stefan Göbel, Michael Kickmeier-Rust, Maic Masuch und Katharina Zweig, 332–377. Lecture Notes in Computer Science. Cham: Springer International Publishing. Besucht am 20. Juni 2022. https://doi.org/10.1007/978-3-319-46152-6_14.
- Sweetser, Penelope, und Peta Wyeth. 2005. „GameFlow: a model for evaluating player enjoyment in games“. *Computers in Entertainment* 3, Nr. 3 (Juli): 3–3. Besucht am 30. September 2022. <https://dl.acm.org/doi/10.1145/1077246.1077253>.
- Tong, Hanshuang, Zhen Wang, Qi Liu, Yun Zhou und Wenyuan Han. 2021. *HGKT: Introducing Hierarchical Exercise Graph for Knowledge Tracing*. arXiv:2006.16915. arXiv, 21. April 2021. Besucht am 21. Mai 2022. arXiv: 2006.16915[cs]. <http://arxiv.org/abs/2006.16915>.

- Turau, Volker. 2009. *Algorithmische Graphentheorie*. 3., überarb. Aufl. München: Oldenbourg.
- Woolf, Beverly Park. 2009. *Building intelligent interactive tutors: student-centered strategies for revolutionizing e-learning*. Amsterdam ; Boston: Morgan Kaufmann Publishers/Elsevier.
- „xAPI“. XAPI.com. 2022. Besucht am 4. Oktober 2022. <https://xapi.com/>.
- Yen, Jin Y. 1971. „Finding the K Shortest Loopless Paths in a Network“. Publisher: INFORMS, *Management Science* (1. Juli 1971). Besucht am 2. Oktober 2022. <https://pubsonline.informs.org/doi/abs/10.1287/mnsc.17.11.712>.

Abbildungsverzeichnis

1.1	Unterstützung durch eine benutzerspezifische Adaption in Spot-X	3
2.1	Beispiele von Adaptivität in dem SG “Lost Earth 2307” (Streicher und Smeddinck 2016)	8
2.2	Beispiel eines Graphen mit Interaktionen zwischen Benutzern und Elementen aus der Filmbranche (Huang u. a. 2019)	8
3.1	Allgemeiner Prozess der Operationalisierung (Heiss und Matthes 2020)	11
3.2	Eigenschaften und Attribute einer Multiple-Choice-Aufgabe in Spot-X	14
3.3	Multiple-Choice-Aufgabe mit unterschiedlichen Schwierigkeitsgraden	15
3.4	Zwei Aufgaben mit unterschiedlichen Schwierigkeitskategorien	15
3.5	Vier-Phasen Adaptivitätszyklus (Streicher und Smeddinck 2016, basierend auf Shute und Zapata-Rivera 2012)	16
3.6	Das Konzept “Dual Flow” (Abbildung von Dörner u. a. 2016, basierend auf Sinclair 2011)	17
3.7	Statistisch belegte Vorteile von One-on-One Tutoring (Woolf 2009, überarbeitet aus Bloom 1984)	22
3.8	Beispiel eines ungerichteten und eines gerichteten Graphen	24
3.9	Beispiel eines gewichteten Graphen G_3	24
3.10	Beispiel eines gemeinsamen Pfads zur Berechnung einer Performance (Streicher u. a. 2021)	28
3.11	Beispiel von zwei idealen Pfaden (Baur 2022)	28
3.12	Beispiel eines Graphen mit abhängigen Konzepten (Ausschnitt aus Piech u. a. 2015)	33
3.13	Generierung impliziter Wissensstrukturen durch das GKT (Nakagawa u. a. 2019)	34
3.14	GKT-Architektur (Nakagawa u. a. 2019)	35
4.1	Problematik des Yen’s k -Shortest-Paths Algorithmus ohne Berücksichtigung der Kantengewichte	39
4.2	Darstellung des Triple-Prinzips als Graph am Beispiel Spot-X	40
4.3	Vergleich der Lernpfade eines Benutzers und eines Experten (IP)	40

5.1	Vergleich des GPS und GKT	44
5.2	Aufgabenset des Beispielszenarios zur Erzeugung synthetischer Lernpfade . . .	49
5.3	Aufgabe “1989” aus dem Aufgabenset des Beispielszenarios (Bildquelle Berliner Mauerfall „Bundesbildstelle - Bilddatenbank“ 2022)	49
5.4	Vereinigung aller möglichen Pfade des Beispielszenarios aus Tabelle 5.3	52
5.5	Anschauung der Operationalisierung Graph-basierter Lernpfade durch Anwendung des GPS	53
5.6	Nutzungspfad des Benutzers “Alice” in Neo4j	54
5.7	Erzeugter Graph in Neo4j	54
5.8	Berechnete IPs durch Anwendung des Yen’s k -Shortest-Paths Algorithmus . . .	55
5.9	Berechneter IP_1^D durch Anwendung des Dijkstra’s Kürzester-Pfad Algorithmus mit negierten Gewichten	56
5.10	Graph des Beispielszenarios mit zugeordneten Skill-IDs für das GKT	57
5.11	Performance-Vorhersage $Perf^R$ von Benutzer C zum Zeitschritt t	61
6.1	Unvollständiger Lernpfad eines neuen Benutzers	62
6.2	Performance-Vorhersagen des GKT	64

Listings

3.1	Beispielhaftes xAPI-Statement	19
3.2	Ergänzttes xAPI-Statement	20
3.3	Beispiel einer Ausgabe des GPS im JSON-Format (Baur 2022)	31
5.1	Gekürztes xAPI-Statement einer gespielten Aufgabe in Spot-X	46
5.2	Cypher Algorithmus zur Generierung eines benutzerspezifischen Nutzungspfad in Neo4j (bauend auf Implementierung von Baur (2022))	53
5.3	Cypher zur Berechnung idealer Pfade mit Dijkstra's Shortest-Path Algorithmus	56
6.1	GPS-Ausgabe für Zoe unter Berücksichtigung der Punktzahl	63
6.2	GPS-Ausgabe für Zoe unter Berücksichtigung der Punktzahl zur Erreichung einer Mathe-Aufgabe	63