UNIVERSITY OF MISKOLC
FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

A GRAPH-BASED INTELLIGENT TUTORING SYSTEM FOR ADAPTIVE KNOWLEDGE MODELING AND RETENTION IN DYNAMIC LEARNING CONTEXTS

PH.D. DISSERTATION

Prepared by:

LÁSZLÓ CSÉPÁNYI-FÜRJES
M.Sc. in Information Engineering

JÓZSEF HATVANY DOCTORAL SCHOOL
FOR COMPUTER SCIENCE AND ENGINEERING
APPLIED COMPUTER SCIENCE
INSTITUTE OF INFORMATION SCIENCE

Head of Doctoral School
Prof. Dr. László Kovács
full professor

Head of Topic Group
Prof. Dr. habil. Jenő Szigeti
full professor, DSc

Scientific Supervisor
Prof. Dr. László Kovács
full professor

Miskolc
2026

"Civilization is in a race between education and catastrophe. Let us learn the truth and spread it as far and wide as our circumstances allow." — H. G. Wells

Statement of Originality

The research presented in this dissertation is the original work of László Csépányi-Fürjes, except where explicitly indicated. In cases where derivations or results are based wholly or in part on the contributions of other sources, proper acknowledgement and full referencing to the original authors have been provided. This dissertation has not been submitted previously, in whole or in part, for the award of any degree, nor is it currently under consideration for such by any other institution.

Signed: László Csépányi-Fürjes

Declaration of Artificial Intelligence Assistance

The author declares that generative artificial intelligence tools were used in the preparation of this dissertation solely for language-related purposes. Specifically, such tools were used to check grammar, improve stylistics, and ensure an appropriate academic tone. No AI tools were used to generate or manipulate research data, perform analysis, or produce original academic content. The intellectual contribution, arguments, and conclusions presented in this dissertation remain the sole work of the author.

Signed: László Csépányi-Fürjes

Acknowledgements

Research is a wonderful and joyful journey, marked by both challenges and moments of fulfillment. There are times of uncertainty, when the path forward may seem unclear. However, one is never truly alone, as there are always inspiring and supportive people along the way who deserve recognition.

First of all, I would like to express my sincere gratitude to my supervisor, Prof. Dr. László Kovács, for his support, insightful discussions and guidance, which greatly contributed to the completion of this work.

I would like to thank my boss at Crosskey, Christa Martin-Sundman, and my coworkers in the Team Accounting for their patience and support throughout the research process.

I would like to thank my colleagues at the Institute of Information Science at the University of Miskolc for allowing me to be part of the team and for their supportive and encouraging comments throughout my PhD journey.

I will always remember with gratitude the advice and support I received from Péter Fuchs and Dénes Vadász, my late mentors and dear friends, who profoundly shaped my scientific mindset.

I would like to thank my sons, Bálint and Vince, for their youthful energy and inspiration that showed me how to fight my way through the jungle of problems.

And finally, and most importantly, I would like to thank my beautiful and loving wife, Mária, for all the support she has given me over the years. She is the one who listened when I was complaining about my failures and when I was happy about my successes. For all this, I dedicate this work to her. Mária, from the bottom of my heart, thank you!

Table of Contents

List of Figures

List of Tables

Abbreviations

AIEd:	Artificial Intelligence in Education
API:	Application Programming Interface
ARD:	Association Rules Discovery
AQG:	Automatic Question Generation
AV:	Answer Validator
BN:	Bayesian Network
BKTM:	Bayesian Knowledge Tracing Model
BSc:	Bachelor of Science
CBL:	Competency-Based Learning
CbKST:	Competency based Knowledge Space Theory
CDS:	Complex Dynamical Systems
CIL:	Continuous Implicit Learning
CN:	Concept Network
DAG:	Directed Acyclic Graph
DK:	Domain Knowledge
DM:	Domain Model
EKSG:	Evolving Knowledge Space Graph
EM:	Emotion Machine
ES:	Expert System
G4L:	Graph for Learn (Graph4Learn)
GenAI:	Generative Artificial Intelligence
GIFT:	Generalized Intelligent Framework for Tutoring
HLD:	High Level Design
HTN:	Hierarchical Task Network
IA:	Intelligence Augmentation
IFL:	Intuitionistic Fuzzy Logic
ITS:	Intelligent Tutoring System
JSON:	JavaScript Object Notation
KGCL:	Knowledge Graph Change Language
KSB:	Knowledge-Sharing-Bridge

KST:	Knowledge Space Theory
KT:	Knowledge Tracing
KU:	Knowledge Unit
LLM:	Large Language Model
LMS:	Learning Management System
LO:	Learning Object
Neo4J:	Native graph database
NN:	Neural Network
ORQ:	Overarching Research Question
OWL:	W3C Web Ontology Language
REST:	Representational State Transfer
RNN:	Recurrent Neural Network
SPM:	Sequential Pattern Mining
SRL:	Self-Regulated Learning
TM:	Theoretical Modeling
ToC:	Theory of Change
UI:	User Interface
UUID:	Universally Unique Identifier
VR:	Virtual Reality
WDDI:	Weighted Distance Dependent Induction
XAI:	eXplainable Artificial Intelligence
YAML:	Yet Another Markup Language

Chapter 1

Introduction

The roots of education reach back to pre-literate societies, when knowledge was transmitted orally across generations. The more experienced members of the community taught the younger ones in order to preserve essential skills and collective wisdom. Hunting, fishing, agriculture, and craftsmanship became part of everyday life and were learned through continuous participation and observation. One of the key limitations of learning in this period was accessibility. Learning could occur only through prolonged interaction with experienced, often older, members of one's immediate environment. Technological know-how was deeply intertwined with moral and practical wisdom, making it an inseparable part of communal life [84].

The invention of writing made it possible to record and transmit knowledge both in time and in distance. However, the transfer of technological or practical expertise continued to rely primarily on experience and personal guidance. Young apprentices learned most effectively by helping older masters, observing their work, and gradually acquiring their skills through practice. Aspiring learners often had to leave their homes and travel long distances before a master was willing to accept them. Writing also enabled a significant expansion in the volume of knowledge that could be preserved, studied, and shared beyond the limitations of individual memory and oral tradition [31].

Ancient Egyptian priests and scribes possessed specialized knowledge that they carefully preserved and transmitted over generations. In ancient Greece, schools such as Plato's Academy and Aristotle's Lyceum played a pivotal role in shaping scientific and philosophical thought. During the Middle Ages, ecclesiastical institutions, mon-

asteries, and cathedral schools emerged as new centers of learning, laying the foundations for what would later become the first universities [34].

During the Renaissance, the invention of the printing press revolutionized access to books, granting education an unprecedented opportunity for expansion. Learners no longer needed to travel long distances in search of knowledge, as written materials became increasingly accessible. However, by the late nineteenth century it had become impossible for a single individual to master all branches of science. This marked the limit beyond which all human knowledge could no longer be comprehended by the mind of a single person [55].

The Industrial Revolution of the nineteenth century demanded new skills and forms of knowledge that increasingly had to be acquired through formal education. As a result, a gradual process toward broader access to schooling began, although educational opportunities remained limited and unevenly distributed across social classes and genders for a long time thereafter. Despite the dramatic expansion of available knowledge during this period, its representation remained largely unchanged, anchored in the written word, and preserved in books. As a result of the intellectual and scientific explosion of the seventeenth and eighteenth centuries, the overall volume of knowledge increased substantially; however, its rate of change remained within a range that was still humanly manageable. Retraining or transitioning between professions could be achieved relatively easily, provided that the individual had both the intention and the opportunity to do so. As knowledge and technology became increasingly complex, the retraining process correspondingly lengthened. By the twentieth century, adapting to major technological changes could require several months or even one to two years of dedicated learning and practice [87].

The emergence of computers and computer networks in the twentieth century marked another major leap in both the volume and accessibility of knowledge. The traditional printed form of knowledge representation has been gradually supplemented and increasingly replaced by digital formats, enabling a transformation from linear to non-linear structures. This shift gave rise to new forms such as hypertext and introduced additional modalities, such as audio, visual, and video materials,

collectively known as multimedia. The evolution of these formats was driven by a fundamental need: to *represent the growing complexity of knowledge more effectively, to capture intricate interrelationships, and to facilitate deeper understanding for readers and learners who engage with the represented information* [80].

The widespread adoption of smart devices and the internet in the early twenty-first century marked yet another significant leap in the volume of accessible knowledge. This era, often referred to as the Information Age, is characterized by the central role of information and knowledge as key assets for individual and societal advancement. At the same time, the temporal validity of knowledge has decreased dramatically: what is known today may become obsolete tomorrow. Mastery of a computer program or an operating system, for instance, requires the acquisition of a substantial body of knowledge, yet, this expertise rapidly loses its relevance as new versions and updates are released, underscoring the accelerating pace of knowledge evolution in the digital era.

The simultaneous coexistence of multiple software and hardware versions often necessitates the parallel management of the corresponding variations or even contradictions, within domain-specific knowledge. In certain cases, a particular function may be executed using command A in one software version, while in a later version the same result requires command B. However, the release of a new version does not immediately render previous versions obsolete; they continue to exist and be used in practice. As a result, *parallel truths emerge: distinct, yet concurrently valid forms of knowledge that must be understood and managed simultaneously within evolving technological ecosystems.*

By the twenty-first century, technological knowledge, while increasingly accessible through digitalization, has also become overwhelmingly abundant and, in many cases, internally inconsistent due to the coexistence of parallel and sometimes contradictory truths. This unprecedented complexity poses challenges to professionals that do not have historical precedent. In the field of information technology, traditional means of knowledge acquisition, such as textbooks, whether printed or digital, are playing a diminishing role. For example, Gyula Obádovics' "Mathematics" (first

published in 1956) remains relevant today, reflecting the stability of foundational scientific knowledge. In contrast, Barnabás Bártfai's "Windows 8 and 8.1 for Everyone", published in 2013, has already become obsolete, as the release of Windows 10 introduced fundamental changes to both interface and internal architecture of the operating system. This example illustrates how technological knowledge can have an increasingly short lifespan, highlighting the growing need for adaptive and evolving frameworks of knowledge representation and learning.

1.1 Motivation and objectives

Having worked as a software developer for over 25 years, I have firsthand experience with the rapid pace of technological change, as well as the challenges posed by the coexistence of parallel and sometimes contradictory knowledge. As a university instructor, I am also faced with the difficulty of determining which content to teach students in a manner that remains relevant over time, particularly as the demand for highly qualified professionals continues to grow in certain sectors. At the same time, the presence of intelligent automation in the workplace is increasing. Although research has established that engaging in meaningful work is a key determinant of human mental well-being [115], the scope of such opportunities appears to be narrowing due to the rise of intelligent automation [112].

This paradoxical situation raises the critical question: how must human education evolve to effectively address these challenges? Education has long been one of the most studied domains. More than 30 years ago, a new research field: Artificial Intelligence in Education (AIEd) emerged in academia. AIEd adopts an interdisciplinary approach, integrating artificial intelligence with pedagogy, psychology, and neuroscience to support the development of learning tools that are personalized, effective, flexible, and engaging. Research in AIEd encompasses both traditional classroom settings and workplace learning, with the aim of enhancing formal education and lifelong learning opportunities.

In light of these ongoing socio-technological developments, I have concluded that amidst the unfolding Fourth Industrial Revolution, humanity increasingly needs the

support of AIEd. The advent of AI Large Language Models (LLMs), such as ChatGPT [1], capable of executing complex tasks and engaging in natural language interaction, has created an entirely new situation. These models enable the construction of computer systems that can communicate with ordinary human users in an everyday language. In practical terms, this implies that every learner could potentially have access to a virtual teaching assistant capable of guiding them through emerging technologies and providing explanations tailored to their understanding. Nevertheless, it is essential to preserve the central role of the human educator, because, as discussed in the historical overview, learning is not just the transfer of knowledge, but also a deeply social process, one that encompasses moral guidance, interpersonal interaction, and the shared experience of growth within a community.

Consequently, in my research, the following objectives and requirements were defined:

- Develop a model for dynamically representing curricula capable of accommodating rapid domain changes and parallel, potentially conflicting truths.
- Create personalized learning experiences and provide effective support for educators, without diminishing the central role of the educator.
- Establish a framework to help educators track student progress and learning outcomes.
- Design a framework to systematically oversee, manage, and develop curricula.
- Define a method for representing the learners' knowledge states in an accurate and actionable manner.
- Implement a prototype system based on the proposed model and frameworks.
- Propose an approach to generalize learning by enabling and promoting AI-driven knowledge sharing.

In addition to the specific research questions addressed in this dissertation, the research is guided by the following overarching research question (ORQ):

ORQ: How can an intelligent tutoring framework be designed to support adapt-

[1] OpenAI, available:https://openai.com/chatgpt

ive, personalized, and sustainable learning in fast-evolving knowledge domains while preserving the central role of educators?

1.2 Ethical considerations

Cognitive maturity: AIEd must be treated with a high degree of responsibility. This dissertation emphasizes that the use of artificial intelligence in education requires a high level of cognitive maturity. Adult learners possess the cognitive maturity and critical thinking skills necessary to interact with AI-driven recommendations without falling into uncritical dependency. Consequently, the system developed in the scope of this dissertation is recommended for Higher Education and for Adult learners.

Human-in-the-Loop: The proposed system in Higher Education is essentially a supplementary tool rather than a substitute for the educator, since it is designed to manage the technical scaffolding, while the teacher retains pedagogical authority over both the curriculum and the assessment. In an Adult learning environment, the distinction between teacher and learner is less clear, so learners need additional motivation to complete assessments and to persist with their studies.

Bias: To address potential bias, the proposed model is transparent and explainable. Unlike *black-box* AI systems, the graph-based approach allows educators to inspect prerequisite relations and prediction logic, ensuring that the system does not reinforce learning biases or unfair educational paths.

Empowerment through autonomy: By providing learners with the tools to understand their own knowledge state, the system grants them a high degree of pedagogical autonomy. The proposed framework is specifically designed to support self-regulated learning environments.

1.3 Dissertation outline

The dissertation is organized as follows. Chapter 1 provides a historical overview that contextualizes the central problem addressed in this research. It also presents

the motivation behind the study and defines its objectives and requirements. The literature review section offers an analysis of existing technologies and previous research relevant to the topic. Chapter 2 presents the implicit Knowledge Sharing Bridge architecture, which explores the concept of embedding an intrinsic teaching capability within AI systems. Chapter 3 introduces the proposed dynamic ITS data model, named the Evolving Knowledge Space Graph, including its formal specification and the algorithms developed to support it. Chapter 4 describes the overall framework that integrates the designed model into a functional system architecture. Finally, Chapter 5 presents details about the prototype implementation and reports on a user study conducted with BSc university students using the implemented system and analyzes the results to evaluate the feasibility and effectiveness of the proposed model and framework. Chapter 6 provides a summary of the dissertation and presents the four theses derived from the research.

Together, these chapters form a coherent body of work that bridges theoretical modeling, system implementation, and empirical validation, advancing the field of intelligent tutoring and human-centered AI. The proposed solutions aim to support learners and educators in rapidly evolving technological environments, providing adaptive and sustainable learning pathways that help individuals stay competent and confident in continuous change.

1.4 Literature review and theoretical background

Rapid technological change, as outlined in the introduction, has transformed not only the structure of the labor market but also the way humans acquire, retain, and apply knowledge. The continuous evolution of software, hardware, and digital infrastructures has resulted in an ever-expanding and dynamic knowledge space, where previously learned information may become obsolete within months. This constant flux presents significant cognitive and educational challenges: people must repeatedly adapt their skills, update their understanding, and navigate multiple, sometimes contradictory, "truths" that coexist across different technological versions and platforms. Traditional curricula and teaching methods that are structured around stable, slowly

changing bodies of knowledge struggle to accommodate the speed and variability of modern technological environments.

This chapter reviews the theoretical foundations and existing research relevant to this challenge. Examines prior approaches to handling dynamic knowledge domains in education, with particular emphasis on Intelligent Tutoring Systems and their evolution toward adaptability and personalization. The discussion highlights the limitations of existing models in managing rapidly changing content and introduces the conceptual basis for the proposed dynamic ITS framework developed in this dissertation.

In fields such as computer science, engineering, and information technology, knowledge evolves so rapidly that even recently acquired competencies can become obsolete in a short period of time. This phenomenon is often referred to as the half-life of knowledge. The half-life of knowledge in computer science and technology is estimated to be between 12 and 18 months, making it one of the shortest among scientific disciplines [109]. This rapid obsolescence requires continuous learning and adaptation to new technologies and methodologies. Integration of emerging technologies such as AI, VR, and digital platforms into education and professional practice can improve knowledge retention and application [81]. E-learning emerged as a significant educational paradigm with the advent of digital technology and the Internet. The initial focus of the e-learning or e-tutoring systems was on content delivery and providing a platform for distance education [111]. However, the emergence of such systems soon enabled researchers and developers to move beyond static content transmission and explore mechanisms that dynamically assess learner knowledge. This shift could allow one to tailor educational content according to the actual understanding of the learner, guided by the underlying pedagogical or algorithmic logic [17].

To facilitate the administration, documentation, tracking, reporting and delivery of educational courses and training programs, a complex integrated platform is required, commonly referred to as a Learning Management System (LMS). LMS platforms serve as centralized environments where educators can design, organize, and distribute learning materials while also monitoring learner engagement and perform-

ance. These systems emerged as a response to the growing need for scalable and efficient educational management, particularly within institutions that provide instruction to large and diverse audiences [106].

Researchers quickly identified a number of fundamental challenges that needed to be addressed in order to develop systems capable of providing meaningful, personalized learning experiences. Key research questions included: How can learning content be stored and structured in a way that allows personalized delivery? How can a learner's current level of knowledge and cognitive abilities be accurately assessed? How can adaptive learning paths be generated for individual learners? And finally, how can a system continuously monitor learner progress, including emotional and cognitive states relevant to the learning process?

These questions called for a re-examination of the pedagogical and didactic principles to align them with computational models of learning [114]. Developing a high-quality LMS required the integration of insights from pedagogy, psychology, and computer science [93]. The ultimate goal of such advanced systems has been to achieve a level of effectiveness comparable to that of human tutors. This aspiration led to the emergence of a new class of educational technologies known as Intelligent Tutoring Systems [18].

Over the past decades, several canonical ITS architectures have been proposed to formalize the internal structure and interaction of these systems.

- Tutoring integrated Expert Systems (ES): these systems integrate expert knowledge and are often web-oriented, facilitating the automated formation of a unified ontological space of knowledge and skills [94].
- Agent-based architectures: these architectures utilize intelligent agents to perform various functions within the ITS, enhancing adaptability and personalization [46].
- Emotion Machine based architectures: adaptations of the Emotion Machine (EM) model, which incorporate multiple layers of "thinking" and dynamic agent activity, promoting learner-driven content creation [110].

- Competency-Based Learning (CBL) models: the focus shifts from time spent learning (e.g., "seat time" or course duration) to the demonstration of mastery over specific, pre-defined knowledge, skills, and dispositions (competencies) [23].
- Generalized Intelligent Framework for Tutoring (GIFT): an open, modular architecture supporting authoring, delivery, instruction, and evaluation of adaptive instruction [101].

In general, an ITS is made up of six major components:

1. Domain knowledge model: Stores the knowledge about the subject matter to be taught. It is essential to provide accurate and relevant instructional content [83].
2. Learner model: represents the system's understanding of the student's knowledge, preferences, and learning progress. This model is crucial for personalizing the learning experience [64].
3. Tutoring module: manages the instructional strategies and interactions with the learner. It adapts teaching methods according to the needs and progress of the student [95].
4. User Interface module: facilitates interaction between the student and the system. It ensures that the system is user-friendly and accessible [100].
5. Communication module: handles the communication between different components of the ITS, ensuring seamless integration and operation [71].
6. Pedagogical model: designs and implements teaching strategies, combining the interests of the student, the capabilities of the tutor, and the characteristics of the subject [71].

These components were originally designed for static or slowly evolving domains, where the structure and content of knowledge remain relatively stable over time. Consequently, the classical ITS architecture provides limited support for rapidly changing or dynamically evolving knowledge environments, such as those found in

computer science, software engineering, or other technologies-driven fields; however, it has nonetheless served as the foundation for numerous ITS implementations and research efforts that can be broadly categorized into three major areas:

1. learner-centered studies, focusing on modeling learner knowledge, abilities, and behavior;
2. content-centered research, addressing the representation and storage of learning materials for adaptive use; and
3. pedagogical-methodological studies, exploring instructional strategies and assessment techniques that guide both teaching and learner evaluation within ITS frameworks.

1.4.1 Learner-centered studies

The first major research direction within ITS is learner-centered modeling, which focuses on representing various aspects of learner characteristics, behaviors, and needs. This line of research seeks to enhance personalization and adaptivity by capturing the cognitive, emotional, and behavioral dimensions of learning. For example, gamification techniques have been explored to increase the motivation and engagement of learners [39]. Similarly, facial recognition and affective computing methods have been used to estimate the emotional states and attention levels of learners in real time [118]. Other works have investigated hybrid recommendation strategies that integrate the identification of learning styles with dynamically customized instructional content and activities to individual preferences [58].

In dynamically evolving domains, learner motivation plays a particularly critical role, as learners must often engage in continuous self-directed learning without external enforcement. Gamification techniques can support this process by maintaining engagement and fostering intrinsic motivation. However, continuous monitoring of a learner's emotional state through methods such as facial recognition raises legitimate privacy and ethical concerns, especially in educational settings. In contrast, identifying or allowing learners to self-select their learning style remains a valuable approach even in dynamic domains. This information can meaningfully inform adaptive in-

struction, helping systems adjust the presentation and sequencing of content while still respecting learner autonomy and data privacy.

1.4.2 Content-centered research

The second major research direction is content-centered, focusing on domain knowledge and its transformation into structured curricula. Domain knowledge models within ITS play a crucial role in representing, organizing, and managing the knowledge that is to be taught. These models define how educational content is structured, how concepts relate to each other, and how learning progression can be guided through these relationships. The main domain knowledge models used in ITS research include the following.

- Ontology-based models: These models provide a structured and formal representation of domain concepts and their relationships, enabling knowledge sharing, reuse, and automated reasoning [95]. Ontology-based approaches facilitate the updating and maintenance of domain knowledge [44] and support the personalization of learning experiences by adapting content according to the learner's profile and performance. Furthermore, ontologies promote interoperability between systems and enable the integration of heterogeneous learning resources [42].

- Concept Networks (CN) and Learning Objects (LO): Concept networks represent the key concepts within a domain and their interrelationships, providing a graphical and cognitive map of the knowledge space. They help learners visualize and comprehend the structural organization of the domain, thus supporting meaningful learning and knowledge integration [123]. Learning Objects are modular and self-contained units of educational content that can be associated with the nodes of the concept network. LOs promote reusability and flexibility, as they can be repurposed in different courses or learning scenarios [60]. They are typically described using standardized metadata that facilitate their organization, retrieval, and integration into learning management systems.

- Data Mining and Knowledge Discovery: In domains where knowledge struc-

tures are not well-defined or explicitly modeled, data mining and knowledge discovery techniques play a critical role in identifying implicit patterns of expertise and learning behavior [43]. These methods rely on analyzing learner interactions, system logs, and performance data to infer relationships and dependencies between knowledge components. Within this approach, Sequential Pattern Mining (SPM) is used to detect recurring sequences of learner actions or problem-solving steps, thus uncovering implicit domain structures and learning trajectories. Similarly, Association Rule Discovery (ARD) identifies relationships between different knowledge units, supporting the construction of richer and more comprehensive domain models [75].

- Expert Models: Expert models represent the idealized knowledge structures and problem-solving strategies of a domain expert. They serve as reference frameworks against which learners' actions and solutions can be compared, allowing the system to identify discrepancies and knowledge gaps [51].

Despite the extensive body of research on intelligent tutoring systems and knowledge representation, relatively few studies have addressed the challenges posed by the rapid obsolescence of knowledge and the coexistence of parallel or contradictory truths in dynamic technological domains. This gap in the literature highlights the critical need for educational models capable of adapting to the pace of technological change while maintaining conceptual consistency and pedagogical coherence.

Education systems in general are increasingly conceptualized as Complex Dynamical Systems (CDS), emphasizing the interdependence, adaptability, and emergent behavior of their components [59]. Learning environments, like other complex systems, exhibit continuous evolution over time, where stability and change coexist in a delicate balance. Consequently, incorporating temporal information into educational data models becomes essential to accurately represent the evolution of the system and capture the dynamics of knowledge development.

In terms of evolving-knowledge representation, in general, the Knowledge Graph Change Language (KGCL) has emerged as a standardized framework for describing modifications in knowledge graphs and ontologies. KGCL provides a structured

means to document, request and communicate graph changes at a high level of abstraction, enabling transparency, collaboration, and traceability among multiple stakeholders [50]. However, the primary objective of KGCL is to support traceable and transparent modifications to existing models rather than to represent the coexistence of multiple knowledge states. Consequently, KGCL lacks the ability to handle parallel truths or to dynamically switch between alternative or evolving versions of knowledge.

Other attempts to address the problem of change in knowledge representation can be found primarily in the field of biomedical ontologies. For example, the DynDiff tool provides a mechanism for comparing different versions of biomedical ontologies by identifying and highlighting changes [37]. Nevertheless, its focus lies on detecting changes rather than representing or managing them within a continuously evolving knowledge structure, as required in educational domains.

In addition, frameworks such as the Theory of Change (ToC) can help in planning and implementing curricular revisions by making implicit assumptions explicit and guiding the development of new pedagogical interventions [119]. However, while ToC contributes to the conceptual and organizational understanding of change, it does not explicitly target the representation of evolving knowledge within a formal model.

After reviewing existing models of representation of educational knowledge, it became evident that representing change and parallel truths requires a hybrid approach that combines the strengths of multiple paradigms.

Ontologies have proven to be highly effective in modeling domain concepts. Typically, the knowledge domain represented by an ontology is separated from the instructional content itself. Ontological structures are grounded in natural object hierarchies, where the relations between elements are based on the generalization and specialization of conceptual entities [86]. In this sense, ontologies capture semantic and taxonomic relations rather than prerequisite or learning dependency relations, which are central to epistemological modeling in education.

A more suitable foundation for representing dynamic knowledge relationships can be found in the Knowledge Space Theory (KST). In KST, a knowledge domain is

represented as a network of nodes, where nodes correspond to problems or questions, and edges represent surmise relations, logical or pedagogical dependencies among knowledge units [38, 41]. According to assessment theory, certain knowledge elements typically precede others in time: that is, a particular problem can only be solved, or a question can only be answered once its prerequisite elements have been mastered.

An important extension of KST is the Competence-based Knowledge Space Theory (CbKST), which is grounded in the observation that learners require specific skills to solve given problems. In CbKST, prerequisite relations can be inferred by identifying the skills necessary to answer questions or complete tasks [16].

Within KST, the set of all questions or problems that define mastery in a given domain is called the knowledge space. A learner's knowledge state is a subset of this space, representing the questions the learner is currently able to answer. Once the actual knowledge state is known, it becomes possible to determine both the *outer fringe*: the set of problems that the learner is ready to attempt next, and the *inner fringe*: the set of problems to revisit when comprehension difficulties occur [33, 41].

To adequately represent the dynamics of evolving knowledge domains, it would be necessary to incorporate the notion of time dependency and, consequently, to formalize temporal relationships within educational knowledge representations. Although the concept of time-dependent graphs has been extensively studied in other scientific domains [117], it has not yet been systematically applied to educational modeling. The introduction of such temporal dimensions could enable more realistic modeling of how knowledge evolves, decays, or transforms over time, an aspect that current systems largely overlook.

In a general sense, time is a temporal construct that can represent either a specific moment or a duration. A single moment is often referred to as a *time instance*, whereas a duration corresponds to a *time interval*. Temporal aspects of real-world entities can be formally represented, for example, by the OWL-Time ontology, a temporal extension of the Web Ontology Language (OWL) designed to describe temporal properties and relations [48].

1.4.3 Pedagogical-methodological studies

The third major research area is digital pedagogy, which explores instructional strategies and assessment techniques that guide both teaching and learner evaluation within ITS frameworks. To effectively use tutoring capabilities, motivation and self-regulated learning emerge as central factors. Even in adaptive and intelligent environments, the success of learning ultimately depends on the intrinsic drive of the learner to engage with and persist in the process. In rapidly evolving domains, where knowledge constantly changes and expands, self-motivation becomes a fundamental prerequisite for maintaining the continuity and adaptability of learning [122]. Self-Regulated Learning (SRL) refers to the learner's ability to actively control their own learning process through goal setting, self-monitoring, and self-reflection. The foundational SRL models [103] emphasize that successful learners are not passive recipients of information, but active participants who plan, monitor, and adjust their strategies based on feedback and task demands.

Through knowledge state prediction, the system can estimate what the learner already understands, what they are ready to learn next, and where potential gaps or misconceptions may exist. This process, known as Knowledge Tracing (KT), involves modeling a student's evolving knowledge state based on their historical interactions with the system. These models use various techniques, including Bayesian networks, Recurrent Neural Networks (RNNs), and Transformer architectures, to capture the complexity of learning processes [96, 56, 49].

In the KST model previously discussed, the adaptive knowledge path algorithm can select from the outer fringe the learning objects that the learner is ready to acquire [41]. Consequently, the process of determining a personalized learning path consists of two main components. First, the learner's actual knowledge state must be assessed, and based on this assessment, their knowledge state across all other learning objects in the model can be predicted. Second, both measured and predicted knowledge states must be represented in a format that facilitates the identification of learning objects suitable for immediate study. Importantly, knowledge acquisition is

not unidirectional; the model also accounts for forgetting, ensuring that the learner's evolving state accurately reflects both learning gains and decay over time.

Forgetting is a fundamental aspect of human cognition with direct implications for educational technology design [74]. Without reinforcement, previously acquired knowledge decays over time, a phenomenon classically described by Ebbinghaus's forgetting curve [69].

Adaptive learning systems increasingly incorporate time-based forgetting models, adjusting knowledge estimates based on the interval since the last interaction of a learner with a concept. Systems like ALEKS operationalize this by updating knowledge states to reflect temporal decay [67], mirroring the forgetting curve.

Recent models also consider forgetting consistency [68], which ensures logical degradation of knowledge within structured representations, such as knowledge graphs. Relatedly, positive knowledge correlation restricts forgetting algorithms to avoid illogical decreases in the understanding of related concepts.

Although some systems explore testing effects [67], where assessment itself can reinforce learning, the findings are mixed. Large-scale data suggest that this effect may benefit high-performing students, while lower performers risk disengagement when faced with repeated exposure to identical items.

As discussed previously, both the measured and predicted knowledge states must be represented within the model to enable the determination of a personalized learning path. Intuitionistic Fuzzy Logic (IFL) appears to be a suitable framework for this purpose. Originally introduced by Atanassov [21], IFL extends classical binary logic by incorporating a third component—indeterminacy in addition to truth and falsehood. This framework reflects the idea that for any given proposition, three cases must be considered: (1) it has been proven to be true; (2) it has been proven to be false; or (3) it remains undecided, and no known finite procedure exists to classify it as either. Unlike traditional fuzzy logic, which defines a degree of truth, IFL distinguishes explicitly between degrees of membership (truth), non-membership (falsity), and hesitation (indeterminacy) ([82]). This makes it particularly well suited

to model the uncertainty of learner knowledge states, especially when information is incomplete or evolving over time ([35]).

1.4.4 Knowledge sharing in general

As artificial intelligence becomes more involved in decision-making processes, there is a growing risk that humans may become overly dependent on these systems, potentially leading to a decline in expertise and mental autonomy. Research consistently emphasizes the centrality of meaningful work in human well-being [115], as it promotes satisfaction through the application and continuous refinement of one's skills [112]. Simultaneously, the accelerating pace of technological change creates an urgent societal demand for lifelong and universally accessible learning opportunities to sustain human adaptability. The emergence of AI Large Language Models (LLMs) such as ChatGPT, capable of performing complex tasks and engaging in natural language interaction [78], further underscores this necessity.

In contrast, Intelligent Tutoring Systems were explicitly developed to support active learning through personalization, adaptive feedback, and learner modeling. However, these systems are typically domain-specific and isolated, rather than integrated into everyday AI applications. This separation limits their ability to facilitate learning during human-AI interactions in the real-world.

Even though Explainable AI (XAI) aims to enhance AI transparency, providing explanations alone does not equate to fostering genuine understanding [26]. Although XAI clarifies how decisions are made [19], it rarely supports the conceptual understanding necessary for users to internalize and utilize AI-generated insights [53].

Furthermore, despite significant progress in the design and accessibility of ITS, educational support remains uneven, particularly for learners with limited resources [62]. Most approaches prioritize infrastructural or economic access rather than embedding pedagogical intelligence directly within AI agents themselves.

This reveals a critical gap: current AI systems lack mechanisms that actively promote user learning during interaction. Although explanations may enhance trust, they

do not teach; and while ITS architectures deliver effective pedagogy, they remain external to everyday AI tools.

1.4.5 Conclusion

To effectively operate in dynamic domains, each of the traditional ITS sub-components must be re-envisioned. The Domain Model must be capable of representing evolving knowledge structures, supporting temporal and version-dependent relationships, and maintaining parallel truths.

The Learner Model must adapt to knowledge that may change or become obsolete while still maintaining an accurate estimation of learner competence over time.

The Pedagogical Model must incorporate mechanisms to recommend updated learning paths as the domain evolves, ensuring continuity between past and current knowledge states.

The User Interface Model must facilitate transparent communication of these changes, helping learners understand how domain evolution impacts their learning trajectory.

Although some research efforts have begun to address aspects of temporal or dynamic knowledge modeling, for example, in adaptive hypermedia systems or lifelong learning frameworks, comprehensive solutions that integrate time dependency and domain evolution across all ITS components remain scarce. This gap highlights the need for a new generation of intelligent tutoring architectures capable of managing dynamic domains holistically rather than through isolated updates.

Finally, Bridging the gap between AI systems and ITS requires generalizing the concept of teaching so that every AI system, regardless of its domain, inherently operates as a tutoring entity capable of supporting human learning alongside task execution.

Within our institute, previous research on knowledge-intensive systems has been conducted under the supervision of Prof. Dr. László Kovács.

Walelign Tewabe Sewunetie focused primarily on Automatic Question Generation (AQG), which falls within the broader category of assessment techniques. In contrast,

the present dissertation does not directly address AQG. Instead, it approaches the assessment problem through the use of Generative Artificial Intelligence, leveraging its capability to dynamically construct evaluation items based on existing learning material.

Hussein Ali Ahmed Ghanim concentrated on ontology-based knowledge representation within tutoring systems. The current dissertation extends this line of research by incorporating the fundamental principles of Knowledge Space Theory. Although Ghanim's ontology-based model also decomposed learning materials into discrete knowledge units, these were typically fine-grained, defined at the conceptual level characteristic of ontological hierarchies. In contrast, the present work operates at a coarser level of granularity, using larger knowledge units to form a semantically interpretable structure. Furthermore, while ontology-based representations traditionally rely on hierarchical tree structures, the proposed Evolving Knowledge Space Graph moves beyond this constraint by introducing a graph-based representation. This enables visualization and spatial organization of knowledge within a virtual learning space, which helps learners develop a better systemic understanding of the domain.

Neither Sewunetie's nor Ghanim's research addressed the challenge of dynamically changing domains or the problem of generating adaptive personalized learning paths and tracking learner progress over time, aspects that constitute the core contribution of the present dissertation.

Tanja Sieber's work, on the other hand, explored the evolution of industrial technical documentation and its sharing between departments, which partially relates to the issue of dynamically evolving domains. Her approach proposed a document management system based on ontology; however, her research did not concern educational applications and remained confined to the ontological modeling framework. The model proposed in this dissertation introduces a novel perspective by combining the concept of abstract time with temporal graph theory, offering an innovative approach to managing and representing evolving knowledge within adaptive tutoring systems.

Chapter 2

Implicit knowledge sharing in AI systems

The core claim advanced in this chapter is that, given their growing resemblance to human-level intelligence, AI systems should implicitly provide teaching and knowledge-sharing capabilities, acting as built-in tutors within their operational domains.

Despite significant advances in intelligent tutoring systems (ITS), access to high-quality educational support remains uneven, particularly for learners with limited resources. Recent work has investigated the use of decentralized technologies, such as the Ethereum blockchain, to democratize tutoring services and mitigate educational inequality by providing low-cost scalable solutions [62]. Although promising, such approaches primarily address logistical and economic accessibility, rather than embedding tutoring capabilities directly into AI agents themselves.

This reveals a critical gap: current AI systems lack integrated mechanisms to foster user learning during real-time interaction. Explanations, while valuable for building trust, do not constitute teaching. Similarly, ITS solutions offer effective pedagogy, but remain external systems rather than being embedded within everyday AI tools. As a result, a void persists precisely where learning could and should occur.

To address this gap, this chapter proposes a novel framework that embeds an implicit tutoring mechanism directly into AI systems through the introduction of the Knowledge-Sharing-Bridge (KSB). The KSB converts conventional AI agents into hybrid entities that can serve two purposes: to execute tasks and to instruct users.

By combining core XAI functions with interactive, personalized teaching elements, KSB enables continuous, contextual learning without requiring users to leave the environment where the AI operates. Previous research emphasizes that intrinsic and extrinsic motivations significantly influence the intention of individuals to share knowledge, particularly within formal virtual communities [20]. The proposed KSB component seeks to leverage these motivational insights by designing AI systems that not only explain but also encourage and facilitate user learning, acting as a motivational partner within the interaction.

The KSB comprises four interlinked components: *Explain* (XAI engine), *Report* (operational analytics), *Control* (user configurability), and *Teach* (instructional guidance). This integration allows AI systems not only to justify their actions, but also to act as informal tutors, gradually enhancing user competence. Using structured knowledge representations, such as word clouds, the proposed framework makes complex decision logic intuitively accessible and pedagogically valuable.

This chapter presents the high-level design and a prototype implementation of the KSB-enabled Teaching AI framework. The proposed solution fills a critical void in current AI applications, offering a novel pathway to blend automation with embedded learning, ensuring that AI systems not only inform, but educate their users in real time.

2.1 Framework Development via Theoretical Modeling

This section explores the framework development process by listing the guiding principles that determined the architecture. It defines and examines the components and their interactions of the KSB framework. In addition, it outlines the validation and analysis steps of the theoretical framework model.

The KSB framework was developed using the Theoretical Modeling (TM) approach grounded in principles from Intelligence Augmentation, emphasizing the comple-

mentary relationship between human cognition and AI systems, where technology is designed to enhance, rather than replace, human understanding and decision-making.

Theoretical analysis formed the basis for identifying a key gap in current AI applications: the lack of universal, implicit, and continuous learning opportunities embedded within the systems themselves. Thus, the KSB framework was designed not as an external educational tool but as an internal component of any AI system that interacts directly with users, making learning a result of usage rather than a separate, explicit process.

2.2 High Level Design principles guiding the KSB framework

XAI, while improving transparency, focuses on explanation rather than active teaching. This gap necessitated a framework that seamlessly embeds teaching functionality into AI systems. The development of the KSB framework was guided by the following core design principles:

- Intelligence Augmentation (IA) over AI: The framework prioritizes improving human intelligence and autonomy rather than replacing it.
- Implicit learning over explicit training: Learning should occur in the flow of using technology, reducing barriers such as cost, time, and motivation.
- Universality and accessibility: The KSB is designed to be integrated into any AI system, regardless of the domain, enabling lifelong learning for all users.
- Transparency and trust: By explaining and reporting decisions, AI can foster a more trusted relationship with users.

These principles seek to transform AI from a marginalizing force to an empowering tool. The necessity for KSB arose from multiple theoretical and practical observations. Despite rapid advancements in AI, most systems lack the ability to teach users how they function, leading to dependency rather than empowerment. Existing ITS focus narrowly on academic domains and are not embedded within general-purpose

AI systems. Furthermore, emerging challenges related to AI overautomation and loss of meaningful human work highlighted the urgent need for AI systems to play a more supportive and educational role in human society. Consequently, the KSB was conceptualized as an internal AI module designed to transfer knowledge from the AI to the user through intuitive and context-sensitive interactions.

2.3 Rationale behind KSB subcomponents

The KSB framework, depicted in Figure 2.1, was designed using a High Level Design (HLD) process, defining the overall architecture and subcomponent interactions. The inclusion of the four subcomponents *Explain*, *Report*, *Control*, and *Teach*, as well as the remaining components, was guided by the need to facilitate user learning.

- **Explain**: Integrates XAI techniques to make the AI's decisions interpretable. This supports cognitive understanding and enhances user trust. It is not enough to provide low level explanation; it is advisable to translate the explaining result into a human understandable format. For instance, in a legal environment explainability means legal explanation.

- **Report**: Offers statistical and performance feedback to users, helping them track system behavior and identify improvement areas in their own interaction or decision-making. To prepare thoughtful decisions in terms of AI control it is crucial to monitor the working of the system as well as to follow the communication between the user and the AI agent. The system must enable users to analyze their interactions, which necessitates the inclusion of the Report subsystem in the KSB. Established metrics already exist for evaluating AI agents (e.g., success rate, accuracy), and it can be anticipated that additional measures will emerge in response to the increasing demand for control.

- **Control**: Empowers users with configurable options that promote autonomy and self-regulation, aligning with principles from self-directed learning. To implement controllability, AI developers must put a set of rules into force in the Control subsystem so that the external users can intervene in the working

of the system in a predetermined way. To avoid demonization of AI technology there must be much larger control possibility provided to the users than today, however it means a great challenge to the system's security. Therefore, to avoid malicious interactions, careful implementation of the Control subsystem regarding security issues is crucial.

- **Teach**: Provides personalized, context-sensitive instructional content, enabling users to develop procedural knowledge on how to replicate or modify the AI-driven task. In a healthy synergy, AI learns from humans and humans learn from AI. Teach is the subsystem that facilitates human learning by providing premeditated feedback in a teaching manner. As opposed to the Explain subsystem, where the aim is to understand the AI's response, the Teach subsystem provides information how to learn the skills of the AI agent. For example, if a legal document is being rejected by the AI-classifier agent the Explain subsystem can point on to the key factors why the document was rejected, while the Teach subsystem gives information how the document needs to be constructed to get it accepted.

- **Internal gateway**: Having an internal gateway makes it possible to scale the internal components of the KSB so that can be extended and customized. Either by adding more components or more instances from existing components the internal gateway can encapsulate the communication and can realize internal security features that protect the subcomponents from malicious impacts. By providing private API the system can be integrated seamlessly into various AI systems.

- **User interface and the integration layer**: Users are communicating the AI agent using the user interface (UI). In the proposed framework the UI must be extended for the user to be able to interact with the KSB and to access learning materials, get explanation, realize control or to query statistical information. These functionalities can be implemented separately from the core AI functionality making it possible to apply advanced learning capabilities. The integration layer provides public API to implement the core functionality

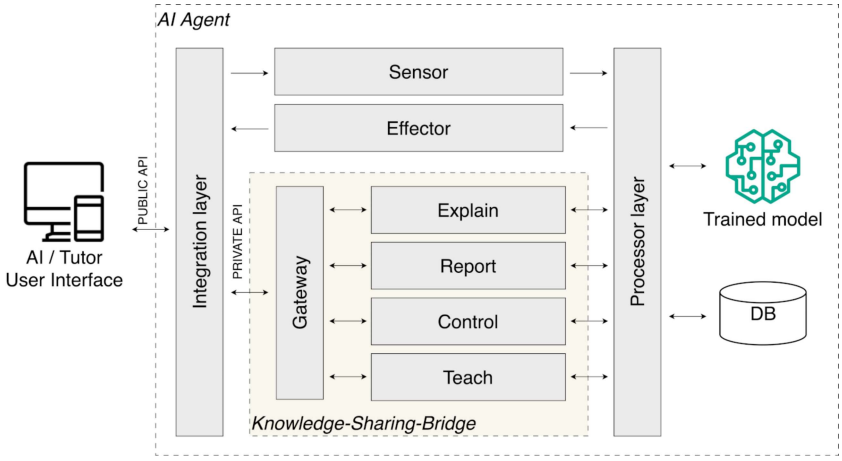


Figure 2.1: KSB framework

as well as to access the KSB functionalities behind the gateway. It ensures seamless integration with various AI systems and platforms, adhering to industry standards.

- **Core AI functionality**: The proposed model is describing a simplified AI agent that consists of the trained AI model as well as a processor layer that implements the business logic of the system. Usually the input/output is realized by the sensor and effector subcomponents. Furthermore, it is important to mention that the system always need a database (DB) where the core functionality related data, user related information or system settings are stored. In the proposed model the KSB related data is also located in this database.

Together, these modules transform AI from a static decision maker into a dynamic educational agent that can provide support to users in real time, enabling skill development and knowledge enhancement. The inclusion of KSB in general AI systems introduces a new model of **continuous implicit learning (CIL)**. Unlike formal educational systems or traditional e-learning platforms, the KSB supports just-in-time knowledge delivery, addressing the knowledge needs of users as they arise during real-world tasks. This aligns closely with the goals of lifelong learning and adaptive learning environments but broadens their reach to include all AI-driven interactions, not just educational applications.

2.4 Theoretical validation

The framework was subjected to Theoretical Validation (TV). The TV of the KSB framework involved a comprehensive analysis of its advantages and potential challenges. Key advantages include universal application, user empowerment, and enhanced trust and transparency. However, challenges such as the complexity of XAI integration and the need for high-quality automated tutoring solutions were also identified. The research acknowledges limitations, including the high-level description of the framework and the use of a small group of university students for preliminary evaluation.

In this section, the practical implementation of the KSB framework is explored through a prototype implementation developed as part of this research, which demonstrates how the *Explain*, *Report*, *Control*, and *Teach* components operate in an integrated AI environment. In addition, this section highlights how the KSB components contribute to the overarching goals of user empowerment and intelligence augmentation.

2.4.1 Workflow

The purpose of the Answer Validator (AV) is to automatically evaluate a textual response from a customer service employee. AV is a simple language-model-based AI system that acts as a virtual customer service trainer. The prototype KSB was implemented as part of the AV module.

The workflow of the system is as follows: the prototype UI shows a question to the employee, then the employee submits a textual answer using the UI to the AV module through the public API that is implemented as a REST service. The REST response received contains the evaluation result as well as a Universal Unique Identifier (UUID) to identify the communication flow. The UI then extracts the UUID and requests an explanation as well as teaching information from the KSB through the public API. The integration layer forwards the request using the private API to the KSB gateway. The gateway routes the request to the *Explain* and *Teach* sub-

components. Both components fetch the necessary information from the DB by the UUID and use the capability of the AI language model to generate a response. The user interface displays both information to the user and expects a corrected response. The process continues until the answer reaches the acceptance criteria level of the AV module.

2.4.2 XAI engine of the Explain and Teach modules

The proposed domain tutoring system is different from the general tutoring systems in many aspects. The main differences include the following elements:

- local scope problem domain
- small knowledge topic focusing on a specific problem
- flexible content
- open interface

The fundamental elements of the framework comprise the modules *Explain* and *Teach*, which generate a clear explanation of the prediction process carried out by the neural network (NN) and provide guidance to the user on how to enter an input that the NN recognizes as a correct response. The XAI engine of the *Explain* module will analyze the NN architecture and generate an interpretable representation of the NN knowledge model. Considering the usual knowledge representation formats used in expert systems, we can highlight the Word-Cloud tool, which shows the key concepts related to the decision process.

2.4.3 Proposed algorithm of Word-Cloud generation of the prototype AV module

To understand the evaluation algorithm of the prototype AV module, let us first observe the AV process itself. In the open-text AV domain, there are two main use cases. In case A, exact words need to be used to answer, while in case B the meaning of the answer is important, but the words themselves can differ. In other words, in case A the answer can be verified by comparing the words one by one while in case

B the semantics of the answer needs to be matched. Case B is more common in real life. For example, a case A question may look like this: *What do the initials HAL for the HAL 9000 computer mean in the film 2001: A Space Odyssey?* The right answer looks like this: *Heuristically programmed Algorithmic computer.* The question in case B may look like this: *How do you greet a customer?.* It can be a question in a customer service environment. The right answer could be: *Hello, how may I help you?.* But semantically, it is also acceptable to answer like this: *Hi, can I help you?.*

The algorithm of the implemented AV module evaluates textual answers using cosine similarity (X, A) between the expected answer $X = \{x_1, x_2, \ldots, x_m\}$ and the actual answer $A = \{a_1, a_2, \ldots, a_n\}$, where m is the number of expected answer tokens and n is the number of actual answer tokens. The implemented logic can be used in both cases A and B. For text representation, the system uses the *STSB-ROBERTA-LARGE* language model [90].

The implemented prototype *Explain* subcomponent evaluates the given answer words one by one and displays them in a Word-Cloud following this pattern: the bigger the word in the cloud, the further it takes the answer away from the expected answer. To calculate word relevance, the algorithm skips tokens a_i of the answer one by one and generates new answers $A_i = \{a_1, \ldots, a_{i-1}, a_{i+1} \ldots, a_n\}$ and calculates the similarity between the resulting new answer and the expected answer (X, A_i). The similarity value in this case is in correlation with the relevance of the skipped token. In other words, the larger the word in the cloud, the more you need to change that word to get to the correct answer.

The prototype *Teach* subcomponent works similarly to the *Explain* subcomponent. The only difference is that it calculates the relevance of the words in the expected answer and provides a cloud with only the few most relevant expected words. This gives a hint to the user about what terms should be included in the answer to get acceptance from the AV module.

The *Report* subcomponent provides information about the number of questions, evaluated questions, and statistics about the result scores in JSON format.

Using the *Control* subcomponent, the learner can change the evaluation scheme to another one that better fits the learner's needs. For example, there is a predefined evaluation scheme that gives a binary answer, such as GOOD/WRONG, or another one that can give a grade based on the similarity score.

2.5 Experiment with the prototype AV module

The prototype AV system implemented with built-in KSB was evaluated with a group of seven university students. The following evaluation objectives (EO) were defined:

- EO1: To evaluate how overall team performance is affected by the KSB component.
- EO2: To evaluate how individual user performance is affected by the KSB component.

EO1 and EO2 are motivated to empirically demonstrate the usefulness of the proposed KSB component by assessing the results of students when they can access the KSB component and when they have access only to the AV module itself.

During the experiment, a set of open-ended questions was presented to the students. The students started to provide answers and received evaluation responses from the AV module. At a certain point, the administrator altered the system settings, so the students started receiving not only evaluation responses but also explanation and teaching Word-Cloud responses from the KSB component. The aim was to observe how students perform when the KSB component is available and when it is not available, as depicted in Figure 2.2 and 2.3.

Remarks regarding EO1: Within the team, there are better and less performing students, just like in any team. When students were able to use only the AV module (KSB enabled = FALSE), the better performing students obtained higher scores and started to achieve better results faster than the others. When KSB was activated (KSB enabled = TRUE), the gap between students becomes smaller and the overall performance of the team becomes balanced, as Figure 2.2 shows.

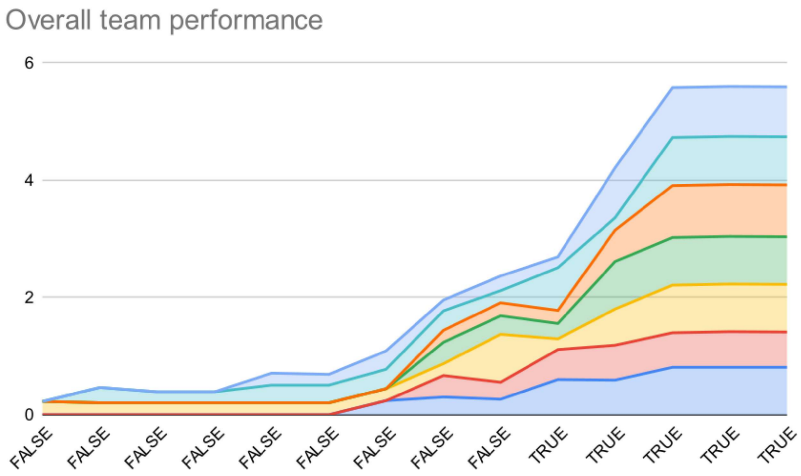


Figure 2.2: Evolving team performance over time (Colors representing individual learners)

Remarks regarding EO2: Since the questions presented were new to the students, even the better performers were struggling with them, but when the KSB module became available, all the students started to perform at a much higher level, as shown in Figure 2.3.

The experiment with the university student group revealed several important insights. Figure 2.2 demonstrates that enabling the KSB component significantly reduced the performance gap between high- and low-performing students. This suggests that KSB subcomponents support personalized learning by adapting to user needs and helping weaker users catch up, thus promoting more equitable outcomes across a group. Figure 2.3 further indicates that all users, regardless of the initial skill level, benefited from the interactive feedback of the KSB, ultimately improving the quality of their responses.

These findings imply that the KSB framework has a strong potential to be integrated into AI systems where decision explanation and user guidance are required. In domains such as customer service training, onboarding, or general workplace learning, the ability to teach users on the fly in a context-sensitive and non-intrusive way could be transformative. Future applications may extend to healthcare, finance, and any domain where trust, understanding, and human-AI collaboration are vital.

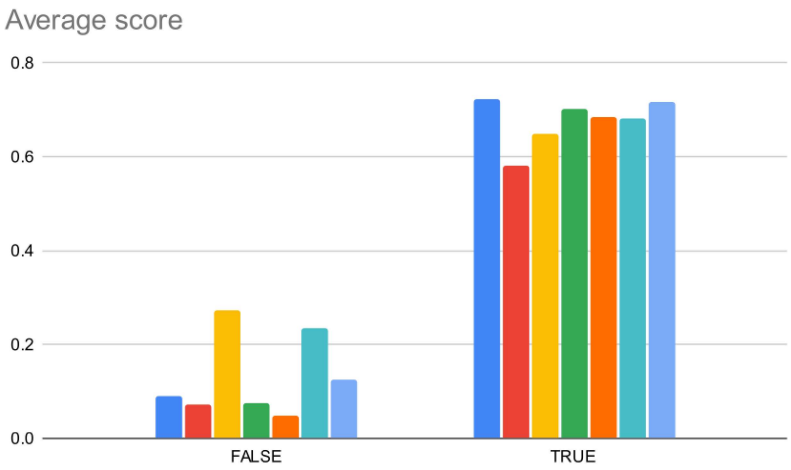


Figure 2.3: Evolving individual performance over time (Colors representing individual learners)

Furthermore, the modules *Explain* and *Teach* powered by XAI principles introduce transparency into a traditionally opaque process. The use of Word-Clouds provides interpretable visual cues, making it easier for users to understand why their responses are incorrect and how to improve. The modules *Report* and *Control*, although more administrative in nature, contribute to an adaptable and trackable learning experience, offering flexibility that aligns with continuous lifelong learning goals.

From a theoretical perspective, the prototype implementation validates the hypothesis that the integration of a knowledge-sharing component within AI systems can bridge the gap between performance evaluation and human learning. Compared to the original AV-only scenario, the KSB-enhanced system demonstrates that explainability and guidance not only improves outcomes, but also empowers users by making AI decisions comprehensible and actionable. The KSB is designed as a modular microservice-based component that is behind an integration layer that harmonizes the KSB with the core functionality. This architectural choice supports scalability and facilitates integration into larger AI systems or enterprise environments by enabling loose coupling and independent deployment of knowledge-sharing functionality. Although the system currently targets a relatively narrow domain with a small test group, the implications are far-reaching. These early results, though limited in scope, suggest that knowledge-sharing AI frameworks could be universally benefi-

cial in enhancing human learning in various contexts. However, scalability, domain adaptation, and further refinement of XAI components will be critical in future developments.

2.6 Thesis 1.

I introduced the Knowledge-Sharing-Bridge framework that embeds an implicit tutoring mechanism directly into Artificial Intelligence systems, which consists of four major subcomponents: *Explain*, *Report*, *Control*, and *Teach*. The modules *Explain* and *Teach*, supported by a dedicated Explainable Artificial Intelligence engine, demonstrate the feasibility of combining explainability with pedagogical guidance. The framework shows potential for applications across domains where human-AI collaboration is critical, including education, corporate training, and technical support. The framework incorporates structured knowledge representations, such as word clouds, that make complex decision logic more transparent, accessible, and pedagogically meaningful. The prototype implementation provides empirical support for the hypothesis that a knowledge-sharing component integrated into Artificial Intelligence systems can bridge the gap between passive consumption of technology and active human learning. The results demonstrate that explainability, when coupled with guidance, not only enhances task outcomes but also empowers users by making AI-driven decisions more comprehensible, actionable, and conducive to sustained knowledge retention.

2.7 Author's publications related to the thesis

Q4: [9]

[2], [3], [4]

Chapter 3

Dynamic ITS data model

3.1 General description of the proposed model

This dissertation proposes the Evolving Knowledge Space Graph (EKSG) model. The following description specifies the logical modifications applied to existing models, such as CbKST, to construct the EKSG framework. The network that represents prerequisite relations in CbKST can be displayed in a precedence diagram. The units of study in ontology can be the replacement of question/problem nodes of this precedence diagram. Thus, using prerequisite relations and units of study, a directed graph can be constructed. The resulted graph combines the benefits of ontologies and CbKST. In addition, two important features are added to the proposed model. To cover the requirements of an evolving domain model, time dependency is introduced and the term abstract time is defined. Although the term time-dependent graph is well known and studied [117], it has not yet been incorporated into educational knowledge representations. The second feature is the use of evoking-hooks that evoke certain knowledge elements by external triggers similar to the solution implemented in the FrameNet system [77].

3.2 Abstract time

As opposed to the well-known and general meaning of time, in this dissertation time is considered as a higher-level abstraction. The reason why the model needs an abstract time property is to be able to represent evolving knowledge. There are

domains where knowledge is not changing that fast, such as mathematics, physics, or chemistry. However, in other fields, such as the computer programming domain, knowledge is constantly evolving. In such dynamic domains, truth itself may exist in parallel forms: for instance, in one version of a software, a particular task requires command A, while in another version the same task requires command B. Both statements are correct within their respective contexts, which highlights the necessity of introducing an abstract notion of time to properly represent and manage such knowledge.

From the perspective of change, the term "time" does not have to mean Gregorian date-and-time. Software scientists invented the concept of "version" to represent change. A version behaves like a time instance. The main properties and functions of the concept of time, such as **before**, **after**, and **interval**, are also interpretable in the concept of version.

Definition 1 *The proposed general abstract time in the given domain is an ordered finite set of* ***time instance*** *values that are strongly related to the domain. Formally:*

$$\mathbf{T}^{inst} = t_1^{inst}, t_2^{inst}, \ldots, t_n^{inst} \tag{3.1}$$

where t_i^{inst} *is an instance value, and for every* $1 \leq i < n : t_i^{inst} < t_{(i+1)}^{inst}$. *Here,* n *is the number of existing instance values.*

Definition 2 *Let* $\mathbf{T}^{int}$ *be a pairwise distinct set of abstract* ***time intervals*** *within the defined domain:*

$$\mathbf{T}^{int} = \mathbf{t}_1^{int}, \mathbf{t}_2^{int}, \ldots, \mathbf{t}_m^{int} \tag{3.2}$$

where $\mathbf{t}_i^{int} = [t_{(p_i)}^{inst}, t_{(k_i)}^{inst}]$ *be an abstract time interval where* $1 \leq p_i \leq k_i \leq n$ *and* $t_{(p_i)}^{inst}, t_{(k_i)}^{inst} \in \mathbf{T}^{inst}$ *therefore* $t_{(p_i)}^{inst} \leq t_{(k_i)}^{inst}$. *The following notation is introduced for the maximal interval:* $\mathbf{t}_*^{int} = [t_1^{inst}, t_n^{inst}]$.

Definition 3 *Let $\prec$ be the matching relation that works as follows:*

instance match:

$$\prec \subseteq \mathbf{T}^{inst} \times \mathbf{T}^{inst}, \tau \prec t_i^{inst} \doteq \tau = t_i^{inst} \tag{3.3}$$

interval match:

$$\prec \subseteq \mathbf{T}^{int} \times \mathbf{T}^{int}, \tau \prec \mathbf{t}_j^{int} \doteq \exists l(p_j \leq l \leq k_j) : \tau \prec t_l^{inst} \tag{3.4}$$

set of instances match:

$$\tau \prec \{t_{l_1}^{inst}, t_{l_2}^{inst}, \ldots, t_{l_p}^{inst}\}, (l_1 \leq l_2 \leq \cdots \leq l_p) \doteq \exists l_i(\tau \prec t_{l_i}^{inst}) \tag{3.5}$$

set of intervals match:

$$\tau \prec \{\mathbf{t}_{l_1}^{int}, \mathbf{t}_{l_2}^{int}, \ldots, \mathbf{t}_{l_k}^{int}\}, (l_1 \leq l_2 \leq \cdots \leq l_k) \doteq \exists l_j(\tau \prec \mathbf{t}_{l_j}^{int}) \tag{3.6}$$

where τ is the time instance that is being examined.

The maximal interval $\mathbf{t}_*^{int}$ is used as the default value when there is no time restriction in the model. Figure 3.1 shows an example abstract timeline in which all three definitions are visualized. The time delay is not interpreted in the proposed model.

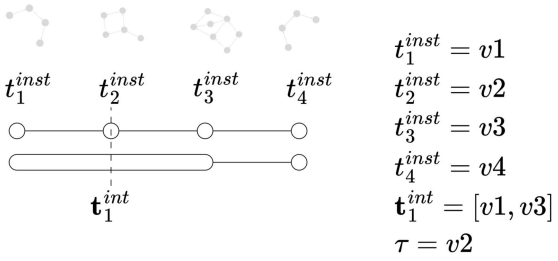


Figure 3.1: An example abstract timeline that shows different software versions which consists of v1, ..., v4 time instances, where $\tau \prec t_2^{inst}$ and $\tau \prec \mathbf{t}_1^{int}$

The matching relation enables the definition of additional operations that are essential for constructing a dynamic data model. Similarly to the classical concept of time (for example, the Gregorian calendar), the relations **before** and **after** can also be established within the abstract time framework.

Definition 4 *Let* $\rightarrow$ *denote the* ***before*** *comparison relation between abstract time instances.*

$$\rightarrow \subseteq \mathbf{T}^{inst} \times \mathbf{T}^{inst}, t_k^{inst} \rightarrow t_l^{inst} \doteq (1 \leq k < l \leq n) \tag{3.7}$$

meaning t_k^{inst} *is before* t_l^{inst} *and* t_l^{inst} *is after* t_k^{inst}.

Taking into account the interval matching relation defined in (3.4), it can be stated that: if $t_k^{inst} \rightarrow t_l^{inst}$ then $t_k^{inst} \prec [t_1^{inst}, t_{l-1}^{inst}]$ and $t_l^{inst} \prec [t_{k+1}^{inst}, t_n^{inst}]$ therefore $[t_1^{inst}, t_{l-1}^{inst}]$ is before t_l^{inst} and $[t_{k+1}^{inst}, t_n^{inst}]$ is after t_k^{inst}.

3.3 Evoking-hook

An evoking-hook is a textual element or representation that serves to connect the appropriate piece of text with a corresponding knowledge unit in the knowledge graph. While this can be a single word or a set of words, such as the keywords 'for' and 'in' in the Python programming domain that evoke the ForLoop knowledge unit, it may also take the form of a more elaborate description or prompt. In this broader sense, evoking-hooks provide a flexible mechanism for linking diverse forms of textual evidence to the appropriate knowledge units. A similar idea appears in the definition of FrameNet, where frame-evoking words are used to activate specific frame elements [77].

Definition 5 *An* ***evoking-hook*** *is a textual element or representation that establishes a connection between the domain item and the corresponding knowledge unit in the knowledge graph. Evoking-hooks may take different forms, ranging from single words and multiword expressions to longer descriptions or prompts that can be processed, for example, by a large language model.*

3.4 The EKSG model

Definition 6 *The proposed* ***EKSG model*** *is a labeled acyclic directed graph that can be described in the following way:* $\mathbf{G} = \{\mathbf{U}, \mathbf{R}\}$ *where* $\mathbf{U}$ *is a set of unit nodes*

representing atomic knowledge, test, and material fragments, and $\mathbf{R}$ *is a set of relations between the unit nodes.* $\mathbf{U}$ *is a triplet:* $\mathbf{U} = \{\mathbf{U}^K, \mathbf{U}^T, \mathbf{U}^M\}$*, where* $\mathbf{U}^K$ *is a set of* ***knowledge-units****, which are elements of a certain knowledge domain, represented by a set of four attributes:*

$$\mathbf{U}^K \subseteq \{(n, d, t, e) \mid n \in \mathbf{N}, d \in \mathbf{D}, t \subseteq \mathbf{T}^{int}, e \subseteq \mathbf{E}\} \tag{3.8}$$

where $\mathbf{N}$ *is the set of non-null and unique unit names. The name must reflect the meaning of the unit. The name acts as an external human-readable identifier;* $\mathbf{D}$ *is the set of unique non-null textual descriptions. The purpose of a description is to be able to better delimit the piece of knowledge represented by the unit;* $\mathbf{T}^{int}$ *is a set of abstract time intervals,* t *may be empty. If* t *is empty, then there is no time restriction in the model, which means that* $\mathbf{U}^K$ *is valid in all time intervals. If* $\mathbf{T}^{int}$ *contains no values, the model is reduced to a timeless graph;* $\mathbf{E}$ *is an evoking-hook set that may be empty.*

$\mathbf{U}^T$ *is a set of* ***test-units*** *which are items of an assessment set. An assessment item can be a question, a problem, or an exercise that is testing whether the learner has mastered a certain knowledge-unit. An assessment item is represented by a set of three attributes:*

$$\mathbf{U}^T \subseteq \{(d, m, t) \mid d \in \mathbf{D}, m : \mathbf{S} \to \{0, 1\}, t \subseteq \mathbf{T}^{int}\} \tag{3.9}$$

where $\mathbf{D}$ *is the set of non-null and unique textual descriptions;* $\mathbf{S}$ *is the set of learners who have completed the test-unit;* m *is a function that gives the information, in a dichotomous manner, if the learner* s_j *has mastered the related knowledge-unit or not. In practice, this information is obtained by presenting the learner with an assessment item, whose response serves as a basis to determine the value of* $m(s_j)$*;* $\mathbf{T}^{int}$ *is the set of time intervals defined as described in 3.8.*

$\mathbf{U}^M$ *is a set of* ***material-units*** *which are holding detailed notes about the knowledge-units. A material-unit is represented by a set of three attributes:*

$$\mathbf{U}^M \subseteq \{(d, n, t) \mid d \in \mathbf{D}, n \in \mathbf{I}, t \subseteq \mathbf{T}^{int}\} \tag{3.10}$$

where $\mathbf{D}$ *is the set of textual descriptions. A material description* d *is non-null and unique;* n *is a detailed note about a specific knowledge-unit;* $\mathbf{I}$ *is the set of all information-material available in the domain in the form of notes. One note can be in many different formats e.g., a hypertext or a multimedia content, etc.;* $\mathbf{T}^{int}$ *is the set of time intervals defined as described in 3.8.*

$\mathbf{R}$ *is a set of directed edges, representing the relations between the units:*

$$\mathbf{R} = \{\mathbf{R}^K, \mathbf{R}^T, \mathbf{R}^M\} \tag{3.11}$$

where $\mathbf{R}^K \subseteq \mathbf{U}^K \times \mathbf{L}^K \times \mathbf{U}^K$ *is a set of ordered triples* $[u^K_{i,t}, l^K_t, u^K_{j,t}]$ *in which* $u^K_{i,t}$ *is a knowledge-unit that is related to another knowledge-unit* $u^K_{j,t}$ *and* l^K_t *is a relation label. The direction of the relation edge is pointing towards* $u^K_{j,t}$*. The relation type between the knowledge-units is determined by the label* $l^K_t \in \mathbf{L}^K$*. Between knowledge-units there are two relation types exist in the proposed model:* $\mathbf{L}^K \subseteq \{PrerequisiteOf_t, IsA_t\}$*. In case of* $PrerequisiteOf_t$ *relation the knowledge represented by* $u^K_{i,t}$ *needs to be acquired before* $u^K_{j,t}$*. In other words,* $u^K_{j,t}$ *depends on* $u^K_{i,t}$*. In case of* IsA_t *relation the knowledge represented by* $u^K_{i,t}$ *is generalized by* $u^K_{j,t}$*.*

$\mathbf{R}^T \subseteq \mathbf{U}^T \times \mathbf{L}^T \times \mathbf{U}^K$ *is a set of ordered triples* $[u^T_{i,t}, l^T_t, u^K_{j,t}]$*, where* $u^T_{i,t}$ *is a test-unit that belongs to one certain knowledge-unit* $u^K_{j,t}$*. The relation type between the knowledge-unit and the test-unit is determined by the label* $l^T_t \in \mathbf{L}^T$*, where* $\mathbf{L}^T \subseteq \{Tests_t\}$ *representing that the actual test-unit* $u^T_{i,t}$ *is testing the mastery level of learner in terms of the connected knowledge-unit* $u^K_{j,t}$*. The edge is pointing towards the knowledge-unit* $u^K_{j,t}$*.*

$\mathbf{R}^M \subseteq \mathbf{U}^M \times \mathbf{L}^M \times \mathbf{U}^K$ *is a set of ordered triples* $[u^M_{i,t}, l^M_t, u^K_{j,t}]$*, where* $u^M_{i,t}$ *is the material-unit that belongs to one certain knowledge-unit* $u^K_{j,t}$*. The relation type between the knowledge-unit and the material-unit is determined by the label* $l^M_t \in \mathbf{L}^M$*, where* $\mathbf{L}^M \subseteq \{MaterialOf_t\}$ *representing that the material-unit* $u^M_{i,t}$ *is a detailed note*

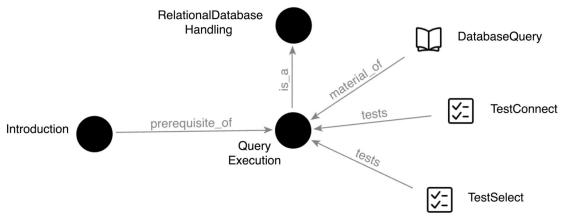


Figure 3.2: An example graph from the databases knowledge domain, that shows the *QueryExecution* KU which has *prerequisite_of* relation to the *Introduction* KU as well as *is_a* relation to the more abstract *RelationalDatabaseHandling* KU. The graph also shows two TUs and one MU connected to *QueryExecution*

that explains and teaches the connected knowledge-unit $u^K_{j,t}$ *to the learner. The edge is pointing towards the knowledge-unit* $u^K_{j,t}$*. Figure 3.2 is visualizing an example graph where the elements of the EKSG model can be observed.*

If the given time interval is empty, the default value is activated, which is t^{int}_*, which means that the model element (node or relation) is valid in all time instances. In case all time restrictions of the model elements are empty, the model reduces to a timeless graph. The fact that the time intervals t may be empty gives the EKSG model great flexibility.

There are several approaches to processing an evolving, time-dependent graph. In the following, the snapshot-based solution is introduced using discrete time-dependent units and time-independent relations.

Definition 7 *Given the* ***snapshot*** $\mathbf{U}^K_\tau = \{u \in \mathbf{U}^K \mid \tau \prec u.t\}$ *(where u.t means: attribute t of object u) contains all knowledge-units in* $\mathbf{G}$ *at time* τ*. The same logic can be applied to material-units* $\mathbf{U}^M_\tau = \{u \in \mathbf{U}^M \mid \tau \prec u.t\}$ *and test-units* $\mathbf{U}^T_\tau = \{u \in \mathbf{U}^T \mid \tau \prec u.t\}$*. These representations are constituting the* τ*th instance of* $\mathbf{G}$ *graph, therefore these instances can be called snapshots. A snapshot of* $\mathbf{G}$ *at* τ *is denoted by:* $\mathbf{G}_\tau = \{\mathbf{U}^K_\tau, \mathbf{U}^M_\tau, \mathbf{U}^T_\tau, \mathbf{R}\}$*. Consequently, the described EKSG model can be considered as a sequence of static knowledge graphs* $\mathbf{G} = \{\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_\tau, \ldots, \mathbf{G}_k\}$*.*

3.5 Knowledge difference in abstract time

From the perspective of ITS, it is crucial to identify the set of knowledge that a learner must master to achieve the final learning goal. Considering time dependency, it can be assumed that the learner has acquired a specific set of knowledge in a given time instance. For example, in a system domain with two versions, $v1$ and $v2$, the learner may have mastered the knowledge corresponding to version $v1$. The key question then becomes: what additional knowledge elements are missing from the learner's skill set to master the same system in version $v2$?

Definition 8 *Given* $\mathbf{G}_{\tau_1}$ *and* $\mathbf{G}_{\tau_2}$ *two snapshots of the EKSG graph, where* $\mathbf{G}_{\tau_1} = \{\mathbf{U}_{\tau_1}, \mathbf{R}_{\tau_1}\}$ *and* $\mathbf{G}_{\tau_2} = \{\mathbf{U}_{\tau_2}, \mathbf{R}_{\tau_2}\}$. *The* ***knowledge difference*** *of* $\mathbf{G}_{\tau_2}$ *and* $\mathbf{G}_{\tau_1}$ *is a subset of knowledge units defined as follows:*

$$\mathbf{d}(\mathbf{G}_{\tau_2}, \mathbf{G}_{\tau_1}) = \{\mathbf{U}_{\tau_2} \setminus \mathbf{U}_{\tau_1}, \mathbf{U}_{\tau_1} \in \mathbf{G}_{\tau_1}, \mathbf{U}_{\tau_2} \in \mathbf{G}_{\tau_2}\} \tag{3.12}$$

which means all knowledge units from snapshot τ_2 *that is not in snapshot* τ_1.

The classic graph difference [47] of the EKSG snapshots provides another notable result. Let $\mathbf{G} = \mathbf{G}_{\tau_2} - \mathbf{G}_{\tau_1}$ be the graph difference of EKSG model, where $\mathbf{G} = \{\mathbf{U}, \mathbf{R}\}$ is an EKSG subgraph, that is calculated as follows:

$$\mathbf{U} = \mathbf{U}_{\tau_2} \cup \mathbf{U}_{\tau_1}; \mathbf{R} = \mathbf{R}_{\tau_2} \setminus \mathbf{R}_{\tau_1} \tag{3.13}$$

which means that all knowledge units from both snapshots are included, with duplications eliminated, together with the relations that exist only in snapshot τ_2 but not in snapshot τ_1. As a final step, the knowledge units without any relations must be removed from the result. With this method, the result contains not only the differences in knowledge units, but, due to the edges, it also captures the inner fringe of the EKSG graph. The inner fringe of the EKSG graph comprises all knowledge units that the learner can master immediately before entering an unknown area, as explained by Falmagne et al. in [41].

3.6 Quantifying knowledge complexity

The volume of a knowledge difference provides essential information about the effort required for the learner to transition from one time instance to another (e.g., from one system version to another). In the proposed model, this volume is quantified by counting the knowledge units contained in the difference set $\mathbf{U}$, as follows:

$$\mathbf{v}_{\tau_2,\tau_1} =\mid \mathbf{d}(\mathbf{G}_{\tau_2}, \mathbf{G}_{\tau_1}) \mid \tag{3.14}$$

In the EKSG model each knowledge unit must cover approximately the same amount of information, therefore the overall amount of information that the learner has to master is in correlation with the number of knowledge units.

The prerequisite relation is a fundamental phenomenon of KST knowledge representation, which is also observable in human sciences [70]. Analyzing the neighborhood of each knowledge unit gives two other important measures [47]. The number of outgoing *PREREQUISITE_OF* edges shows how fundamental that certain knowledge unit is. The incoming *PREREQUISITE_OF* edges predict how complex a knowledge unit is. Given $\mathbf{r}$ the *PREREQUISITE_OF* relation between two knowledge units: u_i, u_j, where $u_i, u_j \in \mathbf{U}$ and $\mathbf{r} \in \mathbf{R}$ and the direction of the edge of the relation points towards u_j. The calculation of the neighborhood set of u_i where u_i is in *PREREQUISITE_OF* relation to the neighbors can be done as follows:

$$\mathbf{N}_{u_i}^{out} = \{u_j \mid [u_i, u_j] \in \mathbf{r}\} \tag{3.15}$$

The calculation of the neighborhood set of u_j where the neighbors are in *PREREQUISITE_OF* relation to u_j can be done as follows:

$$\mathbf{N}_{u_j}^{in} = \{u_i \mid [u_i, u_j] \in \mathbf{r}\} \tag{3.16}$$

It can be stated that if $\mathbf{n}_{u_i}^{out} =\mid \mathbf{N}_{u_i}^{out} \mid$ is relatively high then u_i is a fundamental

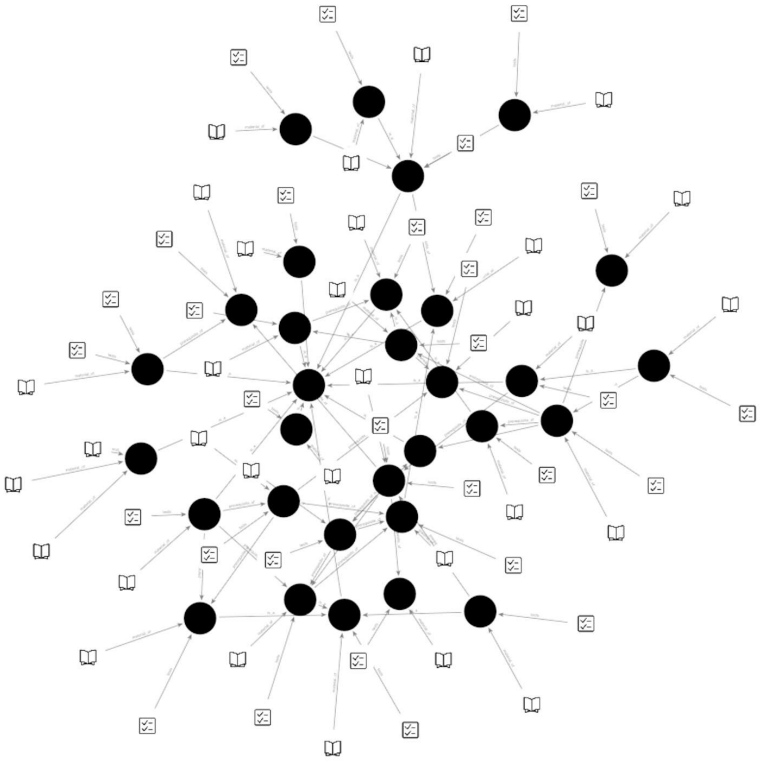


Figure 3.3: EKSG model of the selected 31 Python knowledge units

knowledge unit therefore it worth providing more material as well as more tests to help the learner mastering this unit on a higher level.

Another observation is that if $\mathbf{n}_{u_j}^{in} =\mid \mathbf{N}_{u_j}^{in} \mid$ is relatively high, it means that u_j is too complex, so it is worth considering splitting it up into smaller units.

Consequently, all three values: $\mathbf{v}_{\tau_2,\tau_1}$, $\mathbf{n}_{u_i}^{out}$ and $\mathbf{n}_{u_j}^{in}$ are significant measures of the complexity of the knowledge the learner must master.

3.7 Illustrative Example 1: Python domain

The proposed EKSG model was evaluated by implementing 31 knowledge units in the Python programming language domain. For implementation, the Neo4J Graph database was utilized. The model can be observed in Figure 3.3.

The evaluation investigated whether the knowledge structure of a selected excerpt

from the Python documentation could be adequately represented within the model. The Python methods *str.removeprefix()* and *str.removesuffix()* are presented in Figure 3.4. These methods were introduced in the Python version $v_{3.9}$. Once they became available, the programmer only needed to know what a method is and how to call it to achieve the expected result. According to 3.8, the related knowledge unit since the Python version $v_{3.9}$ is represented as in 3.17, while before Python version $v_{3.9}$ it is represented as in 3.18.

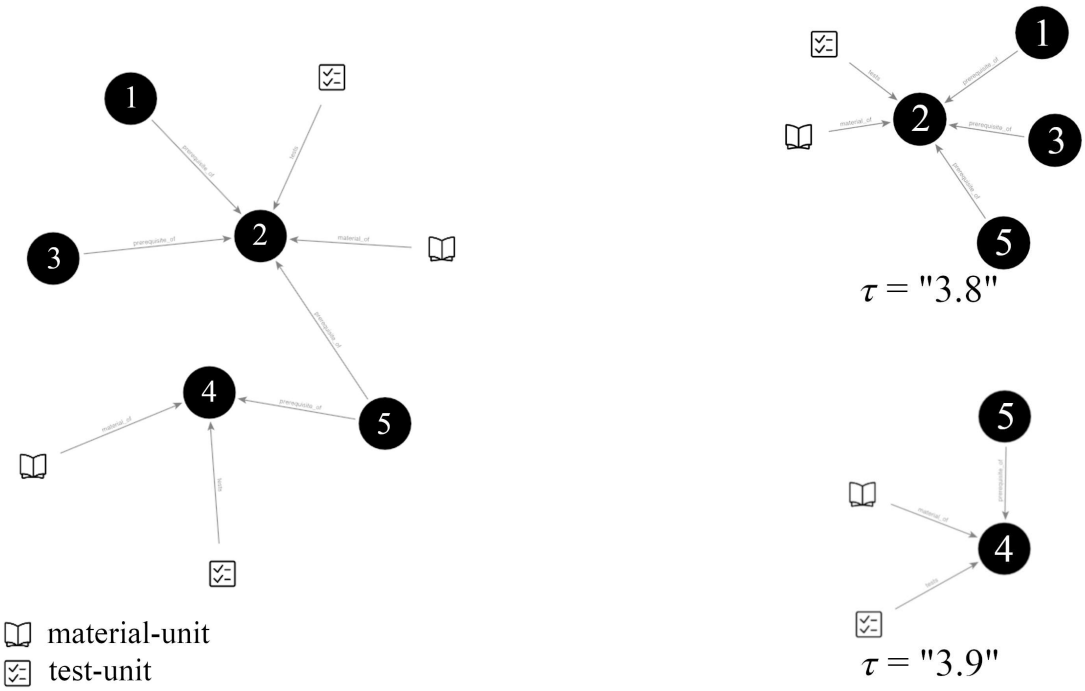
$$\begin{aligned}
&\{n = \text{"RemovePrefixSuffixFunction"},\\
&\ d = \text{"Removing specified text from prefix or suffix..."},\\
&\ t = [\text{"3.9"}, \text{"3.10"}, \text{"3.11"}, \text{"3.12"}],\\
&\ e = [\text{"prefix"}, \text{"suffix"}, \text{"removeprefix"}, \text{"removesuffix"}]\}
\end{aligned} \tag{3.17}$$

$$\begin{aligned}
&\{n = \text{"PrefixSuffixHandling"},\\
&\ d = \text{"You need to implement your own removal..."},\\
&\ t = [\text{"3.6"}, \text{"3.7"}, \text{"3.8"}],\\
&\ e = [\text{"prefix"}, \text{"suffix"}, \text{"startswith"}, \text{"endswith"}]\}
\end{aligned} \tag{3.18}$$

The knowledge required to perform the remove-prefix or remove-suffix operation can be observed in Figure 3.4a, where the knowledge is queried using the evoking hook keywords. The difference between versions $v_{3.8}$ and $v_{3.9}$ has a structural difference which is illustrated in Figure 3.4b.

According to the domain expert, prior to version $v_{3.9}$ the resulting *PrefixSuffixHandling* knowledge unit had the following prerequisites: *NumberOfItems*, *Condition*, and *StringDataType*. Since version $v_{3.9}$, the *RemovePrefixSuffixFunction* has only a single prerequisite relation: *StringDataType*.

Within the implemented data model, the corresponding knowledge structure can be retrieved using a simple Cypher query, as demonstrated in Figure 3.5.



(a) Query: $u.e == "prefix"$ (b) Query: $\tau \prec u.t$

Figure 3.4: Different knowledge structures in Python version $v_{3.8}$ and $v_{3.9}$ for removing prefix and suffix from a string. Black numbered nodes are knowledge-units as follows: 1: *NumberOfItems*, 2: *PrefixSuffixHandling*, 3: *Condition*, 4: *RemovePrefixSuffixFunction*, 5: *StringDataType*

In the EKSG framework, evoking-hooks establish connections between knowledge units and external knowledge representations, such as program source code. For example, in this case, they link to code snippets that implement prefix and suffix removal.

In the KST model, knowledge units must be atomic in order to construct correct prerequisite relations, which is essential for defining personalized learning paths. In contrast, the EKSG model adopts a more realistic approach, in which knowledge units encompass larger portions of knowledge. The size of such units is less precisely defined, but it can be approximately characterized as the amount of knowledge that can be conveyed within one or two pages of text.

```
MATCH (n)-[r]→(p:KnowledgeUnit)
WHERE ANY   (evoker IN p.evokers
                WHERE evoker IN ["prefix", "suffix"]
            )
AND ANY     (version IN p.timeInstances
                WHERE version = "" OR version = "3.8"
            )
RETURN n, r, p
```

Figure 3.5: Cypher query of determining the knowledge structure in version $v_{3.8}$ for prefix and suffix handling

3.8 Illustrative Example 2: Java domain

As a second example illustrating the applicability of the proposed model, this study considers a hypothetical software system, `ExampleProgram`, implemented in Java and structured as a collection of classes. Figure 3.6 provides a visual representation of versions as well as development branches of `ExampleProgram` the content of which is constantly evolving over time.

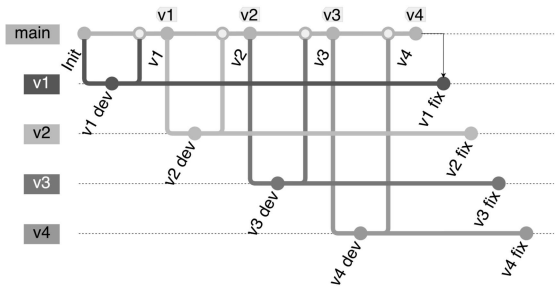


Figure 3.6: The ExampleProgram was developed in parallel across four separate branches, each representing a distinct version of the system. These versions are marked using GitHub TAGs, which serve as identifiers for abstract-time instances within the EKSG model in this example as visualized in the branching structure

The `ExampleProgram` system encodes domain-specific knowledge through the implementation of algorithms within individual Java classes, as illustrated in Figure 3.7. Each Java class encapsulates a discrete knowledge fragment, which we define as a knowledge unit within the EKSG model. These knowledge units are not isolated, they are linked through usage relationships, as classes often invoke methods or rely on data structures defined in others. This interdependence allows us to construct

a prerequisite network between knowledge units. For example, the class `ServiceV1` relies on the functionality provided by the `BasicFeature` class. Thus, understanding the algorithm implemented in `ServiceV1` presupposes familiarity with the concepts encapsulated in `BasicFeature`.

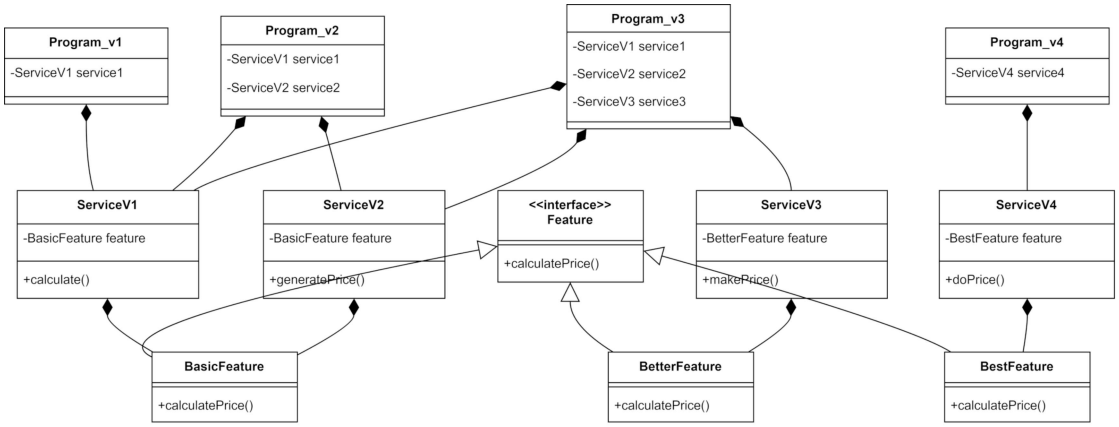


Figure 3.7: The class diagram illustrates the complete class structure implemented across all four versions of the ExampleProgram system

This form of dependency modeling serves as a simplified but practical approach to representing prerequisite relationships in the EKSG model. Over time, as the software evolves, new Java classes may be introduced or removed, reflecting changes in the underlying domain knowledge. As shown in Figure 3.1, different versions of `ExampleProgram` correspond to different sets of available classes. Table 3.1 summarizes the presence of each class in all versions of the demo software.

In particular, while most classes either appear or disappear across versions, one class, the Program, remains present throughout, though its internal implementation changes over time. To reflect this temporal variation in knowledge content, we refer to these evolving knowledge units as `Program_v1`, `Program_v2`, and so forth, each representing the state of the Program class in a specific version.

In the context of abstract-timeline, software versions, such as those created and labeled by a version control system (VCS), serve as abstract-time instances. Each version tag represents a meaningful state of the software associated with a set of Java classes that define the system at that particular stage.

Figure 3.1 demonstrates how each version operates as a discrete abstract-time value,

Table 3.1: Java classes represent knowledge units; software versions correspond to abstract-time instances in the EKSG model

Class	Relevant versions
Feature	v1, v2, v3, v4
BasicFeature	v1, v2, v3
BetterFeature	v3
BestFeature	v3
ServiceV1	v1, v2, v3
ServiceV2	v2, v3
ServiceV3	v3
ServiceV4	v4
Program_v1	v1
Program_v2	v2
Program_v3	v3
Program_v4	v4

anchoring the knowledge state of the system within a specific point on this conceptual timeline.

To support intuitive understanding of the distinction between chronological time and abstract time, a three-dimensional visualization is provided in Figure 3.8. In this figure, the x-axis represents chronological time, measured in weeks, and corresponds to the actual progression of the learner's activity, that is, how learning unfolds over real time.

The y-axis shows abstract-time instances, which reflect structural changes in domain knowledge, in this particular case resulting from the appearance of new software versions or major conceptual updates. Each abstract-time instance is associated with a snapshot of the EKSG model, capturing the state of knowledge at that point in the domain's evolution.

The z-axis captures how the level of knowledge of the learner changes not only over chronological time, but also in response to the structural progression of domain knowledge itself.

This visualization highlights how learners may engage with evolving content at different stages, and that the structure of what needs to be learned (abstract time) does not necessarily align with when the learning occurs (chronological time). This

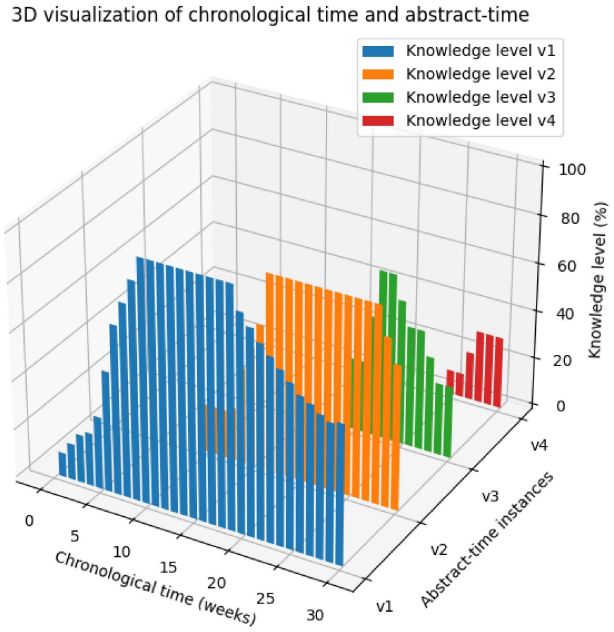


Figure 3.8: The diagram illustrates the relationship between the learner's knowledge level (z-axis), the progression of learning over real time (x-axis: chronological time), and the evolution of domain knowledge (y-axis: abstract-time instances)

conceptual distinction is especially important in dynamic domains, where learning takes place in parallel with the continuous evolution of domain knowledge.

3.8.1 Operations, conditions and metrics

This subsection presents practical examples that demonstrate the application of the formal concepts described previously. These examples aim to illustrate how notions such as the before-after relation or the complexity of knowledge can be interpreted in concrete contexts.

For example, take Definition 4. If there is a need to collect all knowledge related to the development of the `ExampleProgram` system *after* version $v1$, then all knowledge units are needed that are covered by the following time interval: $[v2, v4] = \{v2, v3, v4\}$.

If the learner knows the `ExampleProgram` system in version $v1$. What set of know-

ledge is missing from the learner's skill set to be able to understand version $v2$? The answer is calculated by $\mathbf{d}(\mathbf{G}_{v2}, \mathbf{G}_{v1})$ as described in Definition 8. The `Feature`, `BasicFeature`, `ServiceV1` and `Program_v1` are already known at version $v1$ so the missing skillset is: {`Program_v2`, `ServiceV2`}.

The graph difference between $v2$ and $v1$ gives the following result: {(`ServiceV1`, `Program_v2`), (`ServiceV2`, `Program_v2`), (`BasicFeature`, `ServiceV2`)}. Which means `ServiceV1` and `BasicFeature` are the inner fringe nodes of the result graph.

As Equation 3.14 shows the quantitiy of knowledge difference between $v2$ and $v1$ is $v_{v2,v1} = 2$ because two knowledge units need to be mastered.

According to Equation 3.15 the $\mathbf{N}^{out}_{ServiceV1} = 3$ because it has 3 outgoing `PREREQUISITE_OF` relations towards `Program_v1`, `Program_v2` and `Program_v3`.

Based on Equation 3.16 the $\mathbf{N}^{in}_{ServiceV1} = 1$ because it has one neighbor `BasicFeature` that has a `PREREQUISITE_OF` relation towards `ServiceV1`.

3.9 Key novel components in the proposed EKSG model

The EKSG model extends the classical Knowledge Space Theory by integrating concepts from ontology and temporal graph theory to more accurately represent knowledge in fast-evolving domains. In KST, prerequisite relations among problems can be represented as a precedence diagram. By replacing problem or question nodes with ontology-based study units, a directed acyclic graph can be constructed that combines the logical structure of KST [38] with the semantic richness of ontological modeling [44, 76]. This hybrid representation enables the EKSG to maintain both formal prerequisite relations and conceptual interconnections between knowledge units.

Beyond this integration, the proposed model introduces two major innovations that distinguish it from existing knowledge representation frameworks, such as Hierarch-

ical Task Networks (HTN) [98], Generalized Intelligent Framework for Tutoring (GIFT) [101], or the already mentioned ontology-based model and the KST model.

- *Abstract temporal dependency:* To address the dynamics of continuously evolving domains, the EKSG incorporates a notion of abstract time, which allows knowledge states and their relationships to vary across discrete temporal snapshots. Although the concept of time-dependent graphs is well established in graph theory [117], it has not previously been incorporated into educational knowledge representation models. This extension enables the system to model domain evolution and to reason about changes in knowledge across different time instances (e.g., software versions).
- *Evoking-hooks for external knowledge linkage:* The EKSG introduces the so called evoking-hooks semantic anchors that connect knowledge units to external knowledge representations, such as documentation segments, source code, or examples. Unlike simple keyword-based references, evoking-hooks can take the form of descriptive AI prompts or contextual cues interpretable by large language models, thereby enabling dynamic, context-sensitive retrieval of relevant knowledge. This idea parallels the frame-evoking elements used in FrameNet [77], but is adapted here to support educational knowledge representation and automated content association.

Together, these features establish the EKSG as a flexible and temporally aware framework capable of capturing both the structural and semantic evolution of knowledge in dynamic learning environments.

3.10 Thesis 2.

I introduced the Evolving Knowledge Space Graph model, which integrates the Knowledge Space Theory with ontological structures, while also incorporating the notion of abstract time, larger-grained knowledge units, and flexible connections to external knowledge representations through evoking-hooks. The Evolving Knowledge Space Graph model enables the representation of evolving domains where multiple

parallel truths may coexist (e.g., different software versions) and where knowledge differences and knowledge complexity can be quantified. By integrating these extensions, the Evolving Knowledge Space Graph provides a more realistic and adaptable framework to model knowledge evolution, learner adaptivity, and personalized learning paths in dynamic educational and professional contexts.

3.11 Author's publications related to the thesis

Q3: [7]

[8], [12]

Chapter 4

Dynamic ITS architecture

The creation of the EKSG model established a theoretical foundation for representing knowledge in dynamic learning domains. This model served as the conceptual basis for the implementation of a functional Intelligent Tutoring System (ITS) capable of supporting detailed tracking and analysis of the learning process. My primary objective is to gain in-depth insight into learner activity, not merely through isolated assessments, but by observing the full learning trajectory across interconnected knowledge units.

However, most available tutoring systems focus primarily on static test-based evaluation [28, 54], offering limited support to analyze the dynamic process of both learning and forgetting. To address this gap, I designed my own architecture, named Graph4Learn (G4L), which fully integrates the EKSG model and enables real-time tracking of learner actions, adaptive content delivery, and knowledge state evolution.

4.1 The Graph4Learn architecture

To make the theoretical foundations of the EKSG model operational in the G4L ITS, I implemented a working system architecture in which domain knowledge is represented and queried using a graph database. The core structure of the EKSG is a directed graph, where the nodes and edges encode key elements of the instructional model.

In the implemented system, three types of nodes are used to represent and organize information, as seen in Table 4.1.

Table 4.1: Types of nodes in the EKSG model

Node type	Description
Knowledge Unit	Represent individual concepts or skills within the domain. These nodes are the central entities in the graph and are connected to each other via directed edges labeled `prerequisite_of`, which express the prerequisite relationships among knowledge units.
Material Unit	Each Knowledge Unit node is linked to a Material Unit node, which refers to the corresponding instructional material. The material itself is stored externally in the file system as a PDF document, while the graph stores metadata and references needed to retrieve and display the content.
Test Unit	Each Knowledge Unit is also linked to a Test Unit node. This node represents the evaluative component of the knowledge unit, for example quizzes or test items, used to assess the understanding of the learner. The actual test content is stored in the file system using the YAML format, while the graph node captures descriptive data to query and align with the EKSG.

In addition to the graph-based representation of domain knowledge, the system also uses a relational database to manage data related to learners and system-level operations. This complementary data storage supports user-specific tracking, personalization, and evaluation processes. The main components stored in the relational database can be seen in Table 4.2.

These data sources are integrated and managed by the central G4L system, which comprises several functional subsystems. These subsystems ensure both the management of user data and the delivery of adaptive learning experiences. The main components can be seen in Table 4.3.

Together, these components form a cohesive system as Figure 4.1 shows, which enables dynamic, personalized learning in rapidly changing knowledge domains. The modular architecture also supports experimentation with various adaptation strategies and predictive models.

Table 4.2: Components stored in the relational database

Component	Description
System configuration data	Including global parameters and system-level settings used by adaptive algorithms and content delivery.
Learner activity logs	Record user interactions with the system. This includes access times, selected learning materials, responses to test units, and time spent on tasks, all of which are used for both formative assessment and behavior analysis.
Knowledge state data	Reflects each learner's current understanding with respect to the knowledge units in the EKSG. This data is dynamically updated based on learner interactions and test results.
Teacher related data	Metadata and parameters for prediction algorithms, model configurations, and system-level instructional strategies. This supports the integration and evaluation of various adaptivity mechanisms across learners and contexts.

4.2 Generative AI assistant

To facilitate the transformation of learning content into the EKSG model format, I used a generative artificial intelligence assistant [1], as can be seen in Figure 4.3. Using existing documentation of educational material, the Generative AI (GenAI) assistant automatically generated the prerequisite relational structure of knowledge units, alongside the initial submaterials for each unit. This machine-generated knowledge base was then subjected to a refinement process by a domain expert to achieve its final, pedagogically sound form. Similarly, the creation of quiz questions and answers was also performed by the GenAI assistant, drawing directly from the established content structure. These AI-generated assessment items were subsequently reviewed and validated by the domain expert. The primary pedagogical objective of this entire semiautomated content creation process was to rapidly establish a coherent and comprehensive knowledge base, crucial to the deployment of an adaptive learning environment in a dynamic domain.

[1]The Generative AI assistant used in this implementation was OpenAI's ChatGPT, accessed through its web interface, to assist in generating and structuring educational content.

Table 4.3: G4L subsystems

Subsystem	Description
Learner Subsystem	It handles user registration, authentication, and learner profile management. It ensures secure access and interactions with the system.
Teacher Subsystem	It is responsible for learning content selection and knowledge visualization. It presents learners with an interactive graph-based view of the domain knowledge, allowing them to explore the structure, choose specific topics to study, view the associated materials, and complete relevant assessments. Based on the learner's performance, this subsystem suggests personalized learning paths and identifies knowledge units that may require repetition or review. It also provides instructors with insights into learner progress and tools for configuring or fine-tuning the underlying adaptive algorithms.
Knowledge Retention Subsystem	It operates independently from user-triggered activity and performs scheduled updates of learner knowledge states. These updates take into account both the passage of chronological time and the frequency of repetition, using a configurable forgetting simulation algorithm to adjust the learner model accordingly.

4.3 Graphical representation of knowledge state using an adapted IFL triangle

I used a customized Atanassov intuitionistic fuzzy logic (IFL) interpretation triangle ([21]) to visualize and model the current knowledge state of individual knowledge units. This representation not only displays directly measured knowledge levels but also supports estimations derived from related units via a predictive algorithm. This approach facilitates customized learning path recommendations based on the learner's current knowledge state.

In my proposed implementation, depicted in Figure 4.2 the horizontal leg of the triangle represents the proportion of correctly answered assessment items, while the vertical leg corresponds to incorrect responses. The actual knowledge state of the

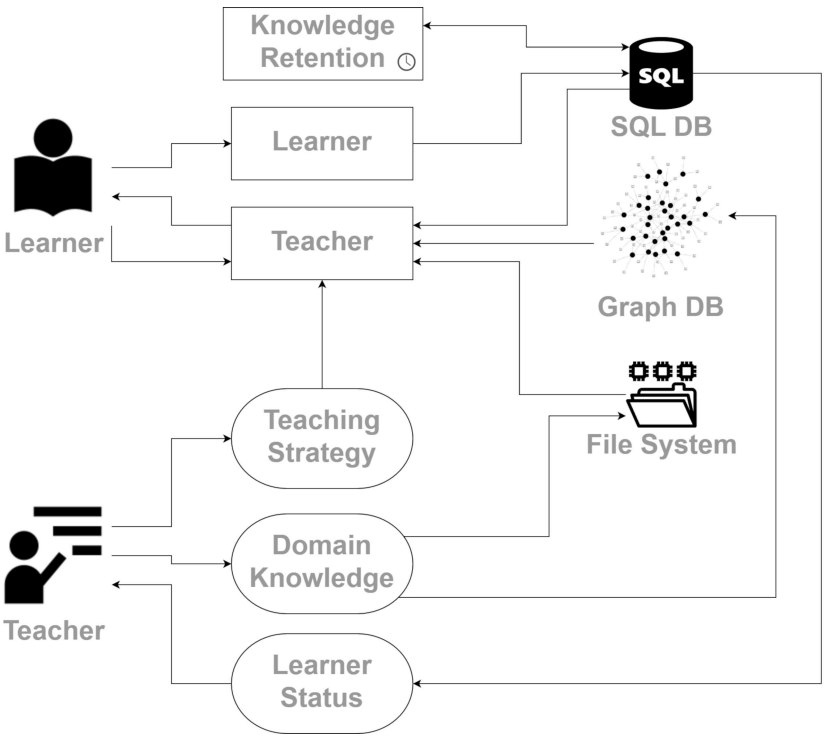


Figure 4.1: The architecture of the G4L system

learner is denoted by a solid dot within this triangle. The triangle vertices define three epistemic categories: point unknown (origin, marked by "?"), point *does not know* (top of the vertical leg, marked by "✗"), and point *knows* (end of the horizontal leg, marked by "✓"). The estimated knowledge state is shown with a hollow point. Therefore, the triangle is partitioned into three disjoint regions: region **KNOWS**, region **DOES_NOT_KNOW**, and region **UNKNOWN**, by a defined turning point that is visualized by the intersecting dashed lines, classifying the learner's knowledge state.

To facilitate adaptive path recommendation using the Bayesian knowledge propagation algorithm, after each assessment, the algorithm identifies all knowledge units with estimated states outside the KNOWS region. For these units, the algorithm calculates their proximity (P) to the Knows vertex. Units are ranked by ascending proximity, and those closest to the Knows point are recommended for further learning as describen in Algorithm 1. This strategy prioritizes units that the learner is most likely to master with minimal effort, distinguishing my approach from classical KST. While KST typically selects units from the outer fringe defined by prerequisite

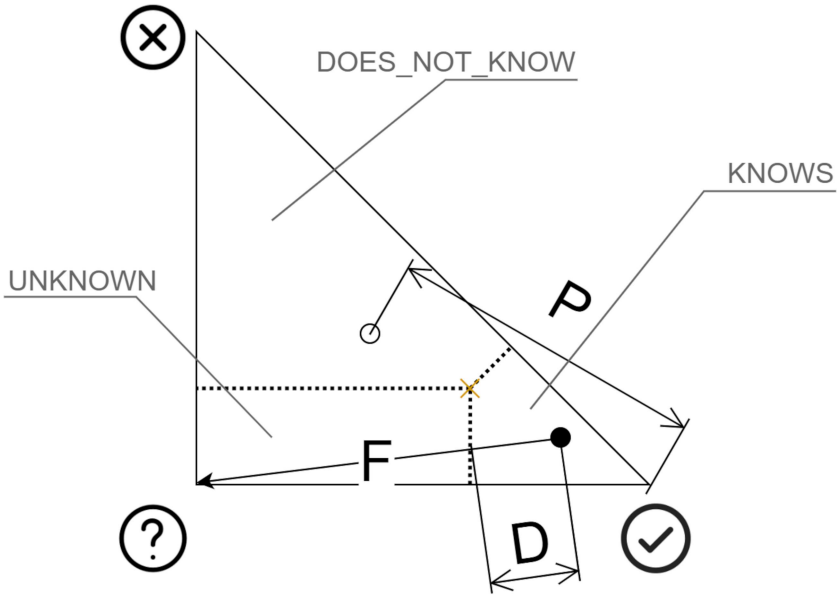


Figure 4.2: The proposed IFL Interpretational Triangle facilitates knowledge state visualization and adaptive path recommendation

relationships ([41]) and is effective for highly granular knowledge structures ([85]), my system focuses on medium to large knowledge units common in real-world educational domains with partial content overlap [83]. This justifies my deviation from the traditional adaptive recommendation based on KST.

Input: Assessment results and predicted states of learner L on knowledge units $\mathbf{U}_{\tau}^{K} = \{u_{1,t}^{K}, \ldots, u_{n,t}^{K}\}$

Output: Ranked list $\mathbf{R}_{\tau}$ of recommended knowledge units

foreach $u_{i,t}^{K} \in \mathbf{U}_{\tau}^{K}$ **do**

  $x_i \leftarrow$ proportion of correct responses on $u_{i,t}^{K}$;

  $y_i \leftarrow$ proportion of incorrect responses on $u_{i,t}^{K}$;

  place (x_i, y_i) in IFL triangle;

  $\hat{x}_i \leftarrow$ proportion of predicted correct responses on $u_{i,t}^{K}$;

  $\hat{y}_i \leftarrow$ proportion of predicted incorrect responses on $u_{i,t}^{K}$;

  place $(\hat{x}_i, \hat{y}_i)$ in IFL triangle;

  if $(x_i, y_i) \in KNOWS$ **then**

    mark $u_{i,t}^{K}$ as *mastered*;

  end

  else

    add $u_{i,t}^{K}$ to candidate set $\mathbf{C}_{\tau}$;

  end

end

if $\mathbf{C}_{\tau} \neq \emptyset$ **then**

  foreach $u_{i,t}^{K} \in \mathbf{C}_{\tau}$ **do**

    $P(u_{i,t}^{K}) \leftarrow$ distance of $(\hat{x}_i, \hat{y}_i)$ to KNOWS vertex;

  end

  sort $\mathbf{C}_{\tau}$ by ascending $P(u_{i,t}^{K})$;

  $\mathbf{R}_{\tau} \leftarrow$ ordered list of $\mathbf{C}_{\tau}$;

  recommend top-k units from $\mathbf{R}_{\tau}$;

end

Algorithm 1: Adaptive learning path recommendation

Forgetting is modeled as a vector shift, represented by a forget (F) arrow, moving the current knowledge point towards the vertex Unknown. The magnitude of this shift is computed using Ebbinghaus's forgetting curve ([40]), influenced by both the elapsed time and the number of repetitions. To prioritize units for review, the algorithm calculates their decay distance (D), the distance from the current knowledge state

point to the threshold beyond which the unit would leave the KNOWS region. Units with the smallest values of D are prioritized for review, reflecting the pedagogical assumption that these are most vulnerable to imminent forgetting as detailed in Algorithm 2.

Input: Current knowledge states of learner L on units $\mathbf{U}_\tau^K = \{u_{1,t}^K, \ldots, u_{n,t}^K\}$, elapsed time Δt, number of repetitions r

Output: Ranked list $\mathbf{R}_\tau^F$ of units prioritized for review

foreach $u_{i,t}^K \in \mathbf{U}_\tau^K$ **do**

  $(x_i, y_i) \leftarrow$ current knowledge state of $u_{i,t}^K$;

  $F_i \leftarrow$ forgetting shift magnitude from Ebbinghaus curve using $(\Delta t, r)$;

  update position: $(x_i, y_i) \leftarrow (x_i, y_i) + F_i \cdot \overrightarrow{\text{UNKNOWN}}$;

  if $(x_i, y_i) \in KNOWS$ **then**

    compute decay distance: $D(u_{i,t}^K) \leftarrow$ distance to boundary of KNOWS region;

    add $u_{i,t}^K$ to candidate set $\mathbf{C}_\tau^F$;

  end

end

foreach $u_{i,t}^K \in \mathbf{C}_\tau^F$ **do**

  assign priority weight $W(u_{i,t}^K) \leftarrow 1/D(u_{i,t}^K)$;

end

sort $\mathbf{C}_\tau^F$ by descending $W(u_{i,t}^K)$;

$\mathbf{R}_\tau^F \leftarrow$ ordered list of $\mathbf{C}_\tau^F$;

recommend top-m units from $\mathbf{R}_\tau^F$ for review;

Algorithm 2: Forgetting-aware review recommendation

4.4 Knowledge Space Theory knowledge state prediction

In Knowledge Space Theory, an individual's knowledge is represented by a *knowledge state* K. Let U^K denote the set of all knowledge units within a given domain.

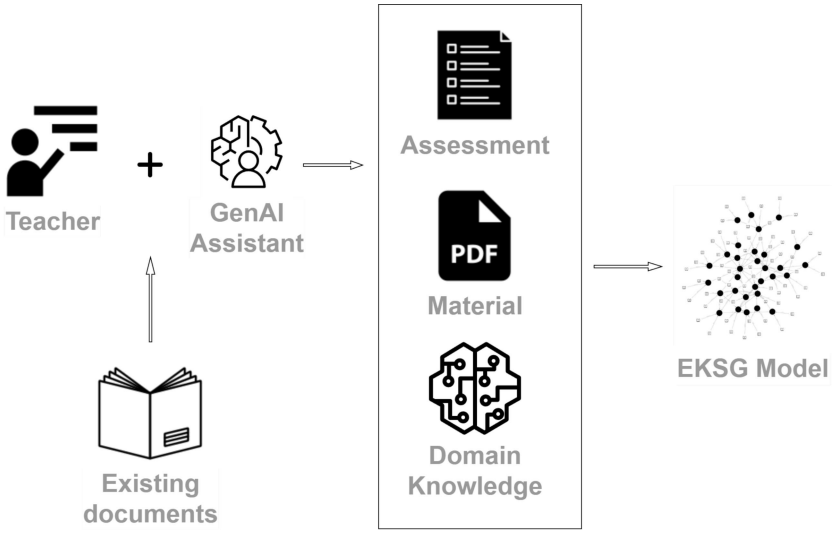


Figure 4.3: How the course knowledge is transformed into an EKSG model with the help of the GenAI assinstant

The knowledge state is then defined as a subset $K \subseteq U^K$, representing the specific elements of U^K that the learner has already mastered.

The collection of all attainable knowledge states within a population of learners forms a *knowledge structure*, denoted by $(U^K, \mathcal{K})$, where $\mathcal{K} \subseteq 2^{U^K}$. Importantly, $\mathcal{K}$ does not contain all possible subsets of U^K (2^{U^K} is the collection of all the subsets that can be obtained from U^K), but only those subsets that are *logically or pedagogically consistent* according to the prerequisite relationships defined within the domain.

Formally, if $a, b \in U^K$ and $a \triangleleft b$ denotes that a is a prerequisite for b, then any admissible knowledge state $K \in \mathcal{K}$ must satisfy the following consistency condition:

$$b \in K \Rightarrow a \in K \tag{4.1}$$

This expresses that a learner who has mastered a particular knowledge unit b must also have mastered all its prerequisite units, in this case a. Conversely, if a learner does not know a, then it cannot be expected that he knows any dependent units b that require a as a prerequisite [30].

This principle ensures that the learner's knowledge state is *structurally coherent* with respect to the prerequisite graph. In the *G4L* framework implementation within

the *EKSG* model, this rule governs the propagation of inferred knowledge states, once mastery of a node is detected, all prerequisite nodes are automatically marked as mastered, while non-mastery propagates in the opposite direction, preventing inconsistency in the learner model.

4.5 Bayesian Network knowledge state prediction

Existing Bayesian Knowledge Tracing Models (BKTM) have been proposed in the literature, for example, in [24] and [65]. To apply this model to the EKSG representation, I adopted a different approach.

Let $G = (U^K, E_{\text{pre}})$ be a directed acyclic graph representing the prerequisite structure of knowledge units, where $U^K = \{u_1^K, u_2^K, \ldots, u_n^K\}$ is the set of knowledge units and $E_{\text{pre}} \subseteq U^K \times U^K$ denotes the prerequisite relations among them.

Each knowledge unit $u_i^K \in U^K$ is associated with a binary random variable $K_i \in \{0, 1\}$, where $K_i = 1$ indicates that the learner has mastered the corresponding knowledge unit, and $K_i = 0$ otherwise.

The joint probability distribution over all knowledge units is defined as:

$$P(K_1, K_2, \ldots, K_n) = \prod_{i=1}^{n} P(K_i \mid \text{Pa}(K_i)) \tag{4.2}$$

where $\text{Pa}(K_i)$ denotes the set of parent nodes (prerequisite knowledge units) of K_i in the graph G.

Posterior update based on learner evidence

When the learner completes an assessment item related to u_j^K, the system observes evidence O_j, representing the probability of mastery inferred from the test performance. For example, if the learner achieved 80% correctness, then:

$$P(O_j \mid K_j = 1) = 0.8, \quad P(O_j \mid K_j = 0) = 0.2 \tag{4.3}$$

Given this observation, the posterior probability of the learner's mastery of u_j^K is updated using Bayes' theorem:

$$P(K_j = 1 \mid O_j) = \frac{P(O_j \mid K_j = 1)\, P(K_j = 1)}{P(O_j)} \tag{4.4}$$

where

$$P(O_j) = P(O_j \mid K_j = 1)P(K_j = 1) + P(O_j \mid K_j = 0)P(K_j = 0) \tag{4.5}$$

Propagation of beliefs

Once the posterior of K_j is updated, the belief propagation mechanism of the Bayesian network adjusts the conditional probabilities of all other connected nodes K_i in the graph, resulting in the following:

$$P(K_i = 1 \mid O_j) = f(P(K_i \mid \mathrm{Pa}(K_i)),\, P(K_j = 1 \mid O_j)) \tag{4.6}$$

where $f(\cdot)$ denotes the probabilistic inference function applied to the network structure.

This process allows the system to infer the probability of expected mastery of the learner for every knowledge unit based on the performance observed in related units.

Interpretation

In intuitive terms, if the learner demonstrates 80% mastery of a given knowledge unit u_j^K, the Bayesian Network updates its internal beliefs to estimate, for instance, that the learner has a 55% probability of mastering a related unit u_i^K, given the prerequisite dependencies and their conditional relations.

This mechanism provides a dynamically adaptive and probabilistic representation of the learner's predicted knowledge state within the EKSG model. While the classical *Bayesian Knowledge Tracing* model operates on a linear sequence of skills, the pro-

posed Bayesian Network approach generalizes this by modeling dependencies among multiple interrelated knowledge units. Thus, it can be seen as a *Bayesian Network-based* knowledge tracing model that combines the logical structure of Knowledge Space Theory with the probabilistic reasoning capabilities of Bayesian Networks.

4.6 Weighted Distance Dependent Induction knowledge state prediction

Building on foundational ideas from KST, the G4L system introduces yet another extended status propagation mechanism called Weighted Distance Dependent Induction (WDDI). While classical models like KST [25] infer knowledge states primarily along prerequisite chains (i.e., from a concept to its prerequisites), my approach expands this inference both forward and backward across the graph structure.

Specifically, knowledge status updates, triggered by learner interactions, are propagated not only to prerequisite nodes, but also to descendant nodes. This bidirectional propagation acknowledges that success or failure in a given knowledge unit can affect both the units it depends on and those that depend on it, even if the relationship is not immediate.

The WDDI algorithm operates directly within the intuitionistic logic framework, maintaining separate value levels for KNOWS, DOES_NOT_KNOW, and UNKNOWN states. To control the influence of more distant nodes, a damping factor is introduced that gradually reduces the propagation strength over longer paths. This weighted model allows for more nuanced prediction of learner knowledge across the graph and enhances adaptive content delivery beyond local node contexts.

State propagation

Let $G = (U^K, E_{\text{pre}})$ be a directed acyclic graph representing the prerequisite structure of knowledge units, where $U^K = \{u_1^K, u_2^K, \ldots, u_n^K\}$ is the set of knowledge units and $E_{\text{pre}} \subseteq U^K \times U^K$ denotes the prerequisite relations among them.

In the knowledge graph, each node is assigned to two kinds of intuitionistic knowledge level. One is based on the measured results related to the node:

$$\Lambda_m = \{(l_{mT}, l_{mF}, l_{mU})\} \tag{4.7}$$

and the second is the inferred level based on the measured levels of the neighborhood nodes:

$$\Lambda_i = \{(l_{iT}, l_{iF}, l_{iU})\} \tag{4.8}$$

In both cases, the condition $l_T + l_F + l_U = 1$ is met. The unit related to a knowledge level is denoted by an upper index, for example: $l_{mT}^{u_i^K}$

The calculation of the inferred knowledge level is based on the following considerations. The knowledge graph G can be managed as a probabilistic network similar to Bayesian networks. Let $I(u_i^K)$ denote the transitive incoming nodes of node $u_i^K \in U_i^K$, and the symbol $O(u_i^K)$ is for the set of transitive outgoing nodes. The knowledge level propagation is based on the following conditional probabilities:

- $P(l_{iT}^{u_i^K} | l_{mT}^{e \in I(u_i^K)})$: the inferred true level depends on the measured true levels of the transitive incoming nodes
- $P(l_{iF}^{u_i^K} | l_{mF}^{e \in I(u_i^K)})$: the inferred false level depends on the measured false levels of the transitive incoming nodes
- $P(l_{iT}^{u_i^K} | l_{mT}^{e \in O(u_i^K)})$: the inferred true level depends on the measured true levels of the transitive outgoing nodes

The first formula expresses the fact that the knowledge level of u_i^K depends on the knowledge levels of the prerequisite knowledge units. In the simplest form, we can say that if the student knows all prerequisite knowledge units, the problem u_i^K will be solved, too. The second formula denotes a similar relationship but refers to the "false" case (not known). If the student does not know some of the prerequisites, he cannot complete the task u_i^K either. The third formula is used to extend the

power of inference; if all outgoing knowledge units can be solved by the student, the common u_i^K prerequisite can be solved, too.

To simplify the set of required parameters in model construction, we apply the following propagation rules.

$$l_{iT}^{u_i^K} = (la_{iT}^{u_i^K} + lc_{iT}^{u_i^K})/2; l_{iF}^{u_i^K} = lb_{iF}^{u_i^K}; l_{iU}^{u_i^K} = 1 - l_{iT}^{u_i^K} - l_{iF}^{u_i^K} \tag{4.9}$$

This formula says that the inferred knowledge level of the unknown factor is calculated from the two other factors. The formula components are defined as the weighted minimum or maximum of the components in the following ways.

$$la_{iT}^{u_i^K} = wmin_{e \in I(u_i^K)}\{w_{u_i^K,e} l_{mT}^e\} \tag{4.10}$$

where $w_{u_i^K,e}$ is a weight factor which is inversely proportional to the distance between u_i^K and e. A typical weight formula is as follows:

$$w_{u_i^K,e} = \delta^{d_{u_i^K,e}} \tag{4.11}$$

where $\delta < 1$ is a positive damping factor and $d_{u_i^K,e}$ is the distance between the two nodes on the knowledge graph. Similarly,

$$lb_{iF}^{u_i^K} = wmax_{e \in I(u_i^K)}\{w_{u_i^K,e} l_{mF}^e\} \tag{4.12}$$

and

$$lc_{iT}^{u_i^K} = wmax_{e \in O(u_i^K)}\{w_{u_i^K,e} l_{mT}^e\} \tag{4.13}$$

In the formulas, the $wmin, wmax$ symbols denote the weighted minimum and maximum operators. These operations are a generalization of the standard minimum and maximum methods. The input for $wmin$ and $wmax$ is a set of values with the

corresponding relevance attributes. In my application, relevance values are treated as probability values in the calculation of the result. Having an input

$$L = \{(v_i, w_i)\} \tag{4.14}$$

we assume that

- $\forall i : 0 \leq w_i \leq 1$
- $max_i\{w_i\} = 1$

The weighted minimum is defined as

$$wmin(L) = \sum_i w_i^* v_i \tag{4.15}$$

where w_i^* denotes the relevance level of the case where v_i is the minimum value in the list. It is defined as

$$w_i^* = w_i - max_{v_j < v_i}\{w_j\} \tag{4.16}$$

If the $\{w_j\}$ set is empty, then $w_i^* = w_i$.

The definition of a weighted maximum has a very similar formula:

$$wmax(L) = \sum_i w_i^{**} v_i \tag{4.17}$$

with

$$w_i^{**} = w_i - max_{v_j > v_i}\{w_j\} \tag{4.18}$$

4.7 Persistent monitoring of the learning process

As Figure 4.1 shows, the EKSG-based G4L architecture supports persistent monitoring of the learning process by storing detailed log records in the database and updating the learner's knowledge state through a dedicated scheduler and knowledge evaluation events.

The system collects a wide range of log events as can be observed in Table 4.4. Each event is time-stamped and associated with both a learner and a related knowledge unit.

Table 4.4: Tracked event types

Event type	Description
user.profile.created	When a learner registers into the system.
topics.requested	When a learner accesses the central study page, which displays available topics and their associated knowledge graph, allowing for topic selection.
topic.changed	Indicating topic change.
learning.started	When a learner opens a material.
learning.finished	When a learner closes a material.
testing.started	When a learner opens a quiz page.
testing.submitted	When a learner requests quiz answer evaluation.
testing.finished	When a learner closes a quiz page with or without submitting their answers for evaluation
test_state.updated	when a learner submits the answers of the quiz, capturing question-level data such as the question identifier and the correctness of the submitted answer.
knowledge_state.updated	Whenever the learner's knowledge state is modified. This occurs in two primary scenarios: following the evaluation of a quiz, where it reflects changes in understanding based on performance, and during discrete forgetting events, when the system updates the knowledge state to account for potential knowledge decay over time.
page.requested	When any page of the system is requested.
recommendation.requested	Whenever the learner's knowledge state is updated, triggering the system to provide a personalized learning path.
knowledge_state .attempt.requested	Logged after a knowledge state recalculation has occurred so the system needs to visually update the knowledge graph. This involves potentially recoloring related node icon and repositioning the knowledge level indicator within the triangle to reflect the newly calculated knowledge state.

The logs are systematically collected and stored within the G4L system's rela-

tional database. The logging architecture is centered on three interconnected tables: `LEARNER_LOG`, `LEARNER_LOG_DATA`, and `CONFIG_LOG`.

The primary `LEARNER_LOG` table captures the core metadata for each event, acting as the main entry point for user-interaction data. It maintains a one-to-many relationship with the `LEARNER_LOG_DATA` table, which stores detailed key-value pairs associated with each log entry. This flexible structure allows for the recording of diverse parameters across various types of learner actions and system responses.

4.8 Core tutoring principles integrated by G4L

Competence-based Knowledge Space Theory (CbKST) ([15]), originally developed by Doignon and Falmagne [38], has provided a powerful framework for knowledge representation and adaptive learning. It has been successfully implemented in intelligent tutoring systems such as ALEKS ([25]) and ELEKTRA ([57]), allowing personalization based on a learner's current level of competence. ALEKS employs a graph-based structure internally but does not expose this to learners, while ELEKTRA emphasizes game-like metaphors and immersive narratives to support engagement and progression.

In contrast, G4L builds on the extended EKSG model ([7]), which builds on CbKST and makes the evolving structure of domain knowledge explicitly visible to learners, allowing them to track their progress within a dynamic, graph-based representation. This visibility not only supports navigation, the importance of which is underlined in the work of Telnov [107], but also serves as a cognitive tool that enhances the reflection and strategic planning of learners.

The G4L system also integrates the core principles of self-regulated learning, as articulated by Zimmerman [122], which emphasize the agency of the learner and proactive participation. In G4L, learners receive system-generated recommendations for their next steps, but these are not binding, allowing them to make informed decisions and take ownership of their learning path [79]. This design aligns with

SRL theory by fostering autonomy, metacognitive monitoring, and strategic control as also suggested by Steiner et al. [103].

The results indicate that the proposed visual graph-based approach allowed a coherent organization of knowledge units [27] with a significantly reduced authoring effort. These findings are consistent with previous work in domain modeling, such as Zouaq et al. [123], who applied ML and natural language processing techniques to generate domain concept maps and ontologies. Similarly, Limaye et al. [66] explored the automatic discovery of the prerequisite relation using burst analysis.

4.9 Thesis 3.

I developed the Graph4Learn system, grounded in the Evolving Knowledge Space Graph model, to integrate graph-based knowledge representation with a relational database, thus enabling real-time tracking of learner activity and adaptive learning path discovery. The system advances the state of the art by introducing the following key innovations. First, it implements an abstract time dependency to capture the dynamics of evolving curricula. Second, it adapts the Intuitionistic Fuzzy Logic model to represent learner knowledge states and forgetting. Third, it implements multiple knowledge state induction algorithms, including the proposed Weighted Distance Dependent Induction method to predict the knowledge state. Fourth, it applies an Intuitionistic Fuzzy Logic based algorithm for the recommendation of the adaptive learning path. Finally, it integrates a Generative AI powered assistant capable of generating curriculum content such as knowledge units, prerequisite relations, and assessment items.

4.10 Author's publications related to the thesis

Q4: [13]
[5], [14], [6]

Chapter 5

Prototype implementation and User Study

5.1 Implementation Details

The G4L system is implemented as a Python-based web application, employing a RESTful peer-to-peer architecture to facilitate communication among its distributed components. The concrete endpoints are listed in Table 5.1. This architectural choice significantly improves scalability and modularity, which are critical for effective adaptation to rapidly evolving technological landscapes. The EKSG model is realized within a Neo4j graph database, primarily selected for its inherent capability to efficiently represent and query the intricate relationships between knowledge concepts using Cypher queries. This graph-native approach offers a distinct advantage in managing the complex characteristic of a dynamic knowledge domain. The learner profiles and activity logs are systematically stored in a MySQL relational database, providing robust and efficient management of structured individual progress data. The specific learning content, tightly linked to the EKSG model, is presented in concise two-page PDF documents, optimizing accessibility and focus. Furthermore, quizzes are stored in YAML files, which ensures ease of authoring and readability for domain experts and instructors, thereby facilitating agile content updates essential in such dynamic fields. The details of the implementation are summarized in Figure 5.1.

The user interface of the Graph4Learn web application was implemented using the

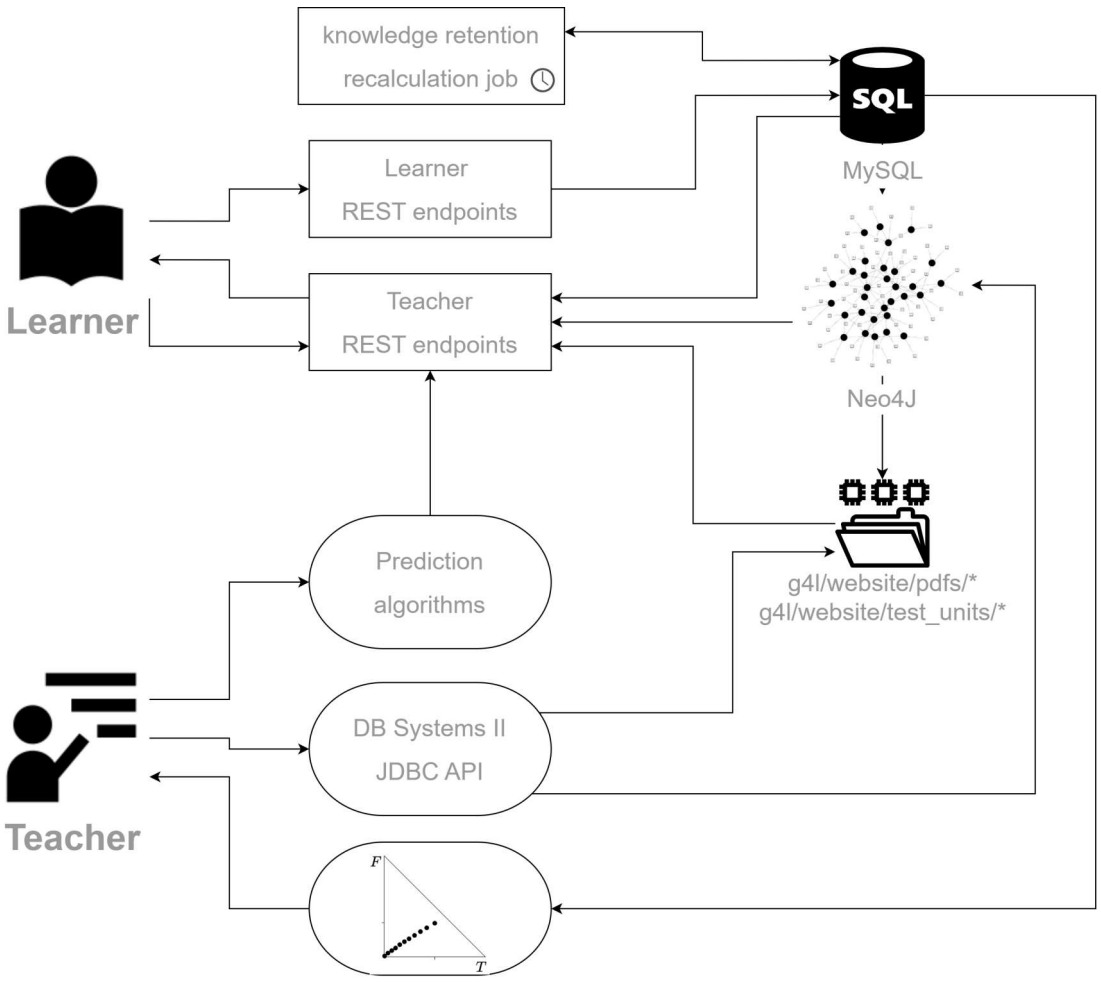


Figure 5.1: The implementation of the G4L architecture

Python Flask framework, which served as the backend to manage server-side logic and data exchange. The front-end can be seen in Figure 5.2 was developed with HTML templates and JavaScript, providing an interactive and responsive user experience. Additional details regarding the system's functionalities are available in the Appendix. The visualization of the EKSG model was realized using the NetworkX library, which enables dynamic rendering and manipulation of graph-based knowledge structures directly within the web interface.

This technological stack was selected for its flexibility, transparency, and suitability for rapid prototyping. Flask offers a lightweight yet powerful architecture that allows seamless integration between the data model and the user interface, while HTML and JavaScript enable fine-grained control over interactivity. The use of NetworkX made it possible to visualize complex graph relationships efficiently.

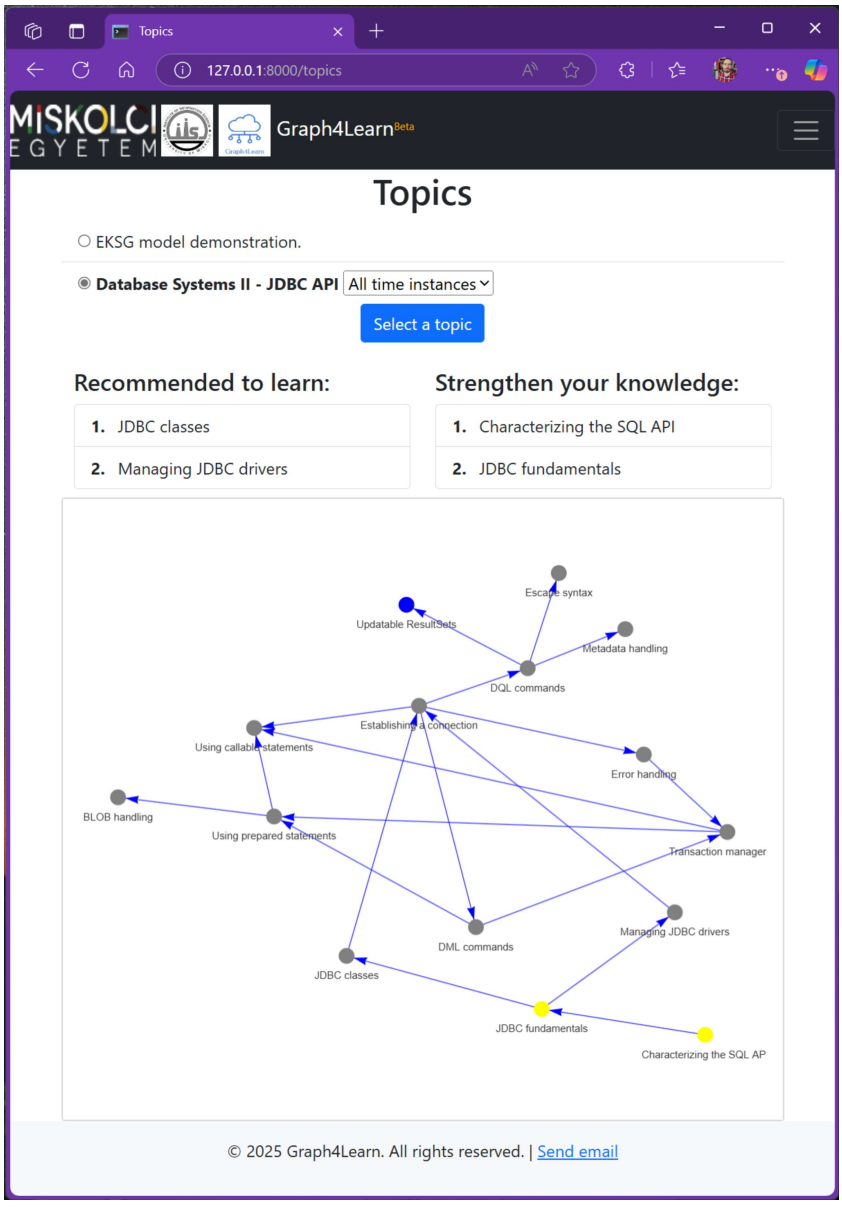


Figure 5.2: User Interface of G4L

Table 5.1: REST endpoints

Functionality	REST endpoints
Configuration	/settings
Knowledge Units	/topics
	/material_unit/<id>
	/material_unit/finished/<id>
	/migrate_knowledge_units
Assessment	/serve_quiz/<id>
	/serve_quiz/finished/<id>
	/submit_quiz
	/evaluation_attempt/<id>
	/results
Recommendation	/get_recommendations
Prediction model	/train_prediction_model
EKSG model	/export_knowledge_graph
	/save_all_node_positions
	/get_node_positions
Activity log	/learner-log

5.2 Research questions

To investigate the effectiveness and practical applicability of the G4L intelligent tutoring system, a mixed methods [92] user study was conducted involving 45 university students. Based on the G4L framework, I formulate the following research questions:

- **RQ1**: How effective is the EKSG model in structuring and representing domain knowledge and supporting learner navigation in a fast-evolving domain?
- **RQ2**: How do different adaptive learning algorithms compare in terms of learner performance and progression in an EKSG model based intelligent tutoring system?
- **RQ3**: How does the implementation of the EKSG model support real-time tracking of learner activity and knowledge state progression within the system?

5.3 Study design

The study was designed to address the three core research questions: First, I examined how well the EKSG model structures and represents domain knowledge while supporting learner navigation in a rapidly evolving domain (RQ1), using a qualitative approach based on a structured questionnaire to collect subjective feedback and perceptions of learners. Second, I evaluated the performance and usability of three implemented adaptive learning algorithms:

- Bayesian Network (BN) ([65])
- Knowledge Space Theory (KST) ([38])
- Weighted Distance Dependent Induction (WDDI)

in a comparative setup (RQ2), using a quantitative approach. The evaluation was based on the analysis of the algorithm performance and the retention of long-term knowledge by learners, measured through offline assessments carried out during the final exam. Finally, I analyzed the system's ability to track learner activity and knowledge states in real time (RQ3). This investigation followed a quantitative approach by evaluating structured and system-generated log data and knowledge state records ([89]), which allowed us to quantify and assess the extent and patterns of system usage.

5.4 Instructional content and participants

The selected instructional content was derived from the "JDBC API" subdomain of the Database Systems II course. The material was segmented into 15 knowledge units. Each KU was linked to an approximately two-page PDF file, considered a component of large granularity knowledge [83], through the relevant material unit. This document offered an explanation and conceptual dissection of the subject at hand.

To assess knowledge acquisition, each test unit was linked to a test bank of 10 multiple choice questions, of which 5 questions were randomly selected for each

test attempt. Both the order of questions and the order of answer options were randomized to prevent memorization. During the assessment, students received only their total score and no feedback was provided on which specific responses were correct or incorrect [91]. This design choice encouraged iterative work with the learning content and the tests.

Informed consent was obtained from all individual participants upon registration for the study. Participants were informed of the voluntary nature of their participation and that their data would be anonymized or pseudonymized prior to publication.

Upon registration in the G4L system, each participant was asked to complete a short demographic questionnaire, which also captured their prior familiarity with the course content and their confidence in online learning environments. A summary of these data is presented in Table 5.2. For an unknown reason, 3 participants were inactive during the experiment. The male-to-female ratio in my sample aligns with the prevailing gender distribution in Computer Science at our university.

The participants were automatically assigned by the system to one of the three predefined groups by the prediction algorithm used (BN, KST, WDDI) in a randomized but balanced fashion. Before the actual experiment, each student received a brief orientation session, including a hands-on demonstration of the system using a sample lesson.

After orientation, students were free to use the system at their own pace. However, dedicated practice sessions were incorporated into the practical labs of the course, ensuring that all students had structured time to engage with the G4L platform and learn using its features.

Table 5.2: Information about the participants (weighted familiarity: : yes = 3, no = 2, can not decide = 1; weighted online confidence: highly confident = 3, averagely confident = 2, not confident = 1)

Group	Registered	Inactive	Female (%)	Average Age	Familiarity	Confidence
BN	14	2	21	22.50	2.36	2.21
KST	16	1	6	22.10	2.25	2.50
WDDI	15	0	20	23.60	2.40	2.13

5.5 Rubric for system evaluation

To evaluate the G4L system, I developed a rubric based on the insights of the course teachers, my implementation experience, the system logs and the learner database [61]. This rubric defines six key aspects of the system which are presented in Table 5.3.

Table 5.3: Rubric linked to the appropriate RQ

	Criterion	Description
RQ1	Knowledge decomposition	How clearly the EKSG model decomposes the domain into coherent and meaningful knowledge units
	Prerequisite logic clarity	Clarity and interpretability of prerequisite relations in the graph
	Graph-based navigation usability	How intuitive and useful the graph is for learners when navigating the content
	Support for learner decision-making	Whether the structure helps learners make informed decisions on what to learn next
RQ2	Learning algorithm evaluation	To what extent the learning algorithm helps learners producing strong learning outcomes
RQ3	Learner activity tracking	How well the system provides information about the learner interactions

The performance of the system was then evaluated for each of these aspects using a 3-point Likert scale as presented in Table 5.4, resulting in a total possible score of 18 for the defined aspects.

5.6 Effectiveness of the EKSG model in structuring domain knowledge and supporting learner navigation

In relation to RQ1, I developed a set of questionnaire items to discover learners' experiences in relation to knowledge representation and learning activities. The

Table 5.4: Rubric

Criterion	1 - Poor	2 - Fair	3 - Good
Knowledge decomposition	The decomposition of the domain is unclear or inconsistent. The knowledge units appear arbitrary or fragmented and it is difficult for learners to understand the structure or relationships between them.	The decomposition results in mostly coherent and meaningful knowledge units, but there are areas where granularity is either too fine or too coarse, making it harder to align with specific learning goals or navigate the domain effectively.	The domain is clearly and logically decomposed into well-defined knowledge units. Each unit represents a distinct concept with meaningful boundaries, and the relationships between units support intuitive understanding and effective learning progression.
Prerequisite logic clarity	Learners struggle to understand why certain topics precede others, which negatively impacts navigation	The model includes prerequisite links that are mostly logical and understandable, but some dependencies seem arbitrary or insufficiently explained.	Learners can clearly see the logical progression and dependencies among topics, which supports effective learning paths.
Graph-based navigation usability	The graph-based navigation interface is difficult to use or understand.	Some learners can navigate effectively, but others struggle with understanding interaction mechanics.	Learners can easily locate knowledge units, understand their relationships, and navigate the graph efficiently.
Support for learner decision-making	Learners are unsure what to study next or whether they should review previous material.	The recommendations are not always clear, consistent, or helpful for all learners.	The system provides clear, personalized recommendations that help learners decide what to study next or what to revisit.
Learning algorithm evaluation	The algorithm results in limited knowledge retention or learner progress. Quantitative indicators show weak effectiveness.	Learners show reasonable progress, and the results are comparable to other algorithms, though without outstanding performance.	The algorithm consistently produces strong learning outcomes. It supports both progress and long-term retention, with top scores across key metrics.
Learner activity tracking	The system captures minimal learner interaction data, with limited detail or coverage.	Some learning trends can be identified, though certain aspects (e.g., timing, navigation behavior) may be underrepresented.	The system provides comprehensive and fine-grained logging of learner activity, including timestamps, navigation paths, test interactions, and content usage.

questionnaire was administered using Google Forms. The learners responded to the items on a 5-point Likert scale, where 1 indicated "strongly disagree" and 5 indicated "strongly agree." The specific questionnaire items are detailed in Table 5.6. In addition, an open-ended item was included to allow learners to provide general suggestions and express their overall opinions on the system. The Google Form was made available to students at the end of the semester and remained open for responses until the end of the examination period. The results are in Table 5.7.

To evaluate the system against my defined rubric, the average scores of the responses on the 5-point Likert scale were subsequently converted to the 3-point scale used in the rubric. This conversion was performed according to the mapping defined in Table 5.5.

Table 5.5: 5-point Likert scale conversion to 3-point scale

Average (1–5 scale)	Converted rubric score (1–3 scale)
1.0 – 2.4	1
2.5 – 4.3	2
4.4 – 5.0	3

This conversion allowed me to integrate the feedback of the learners directly into the rubric-based performance assessment of the G4L system.

Of the 45 registered learners, three did not participate in the experiment for unspecified reasons. A total of 28 learners completed the post-experiment questionnaire, corresponding to approximately 67% of the active participants. This response rate was deemed sufficient for publication and analysis.

Table 5.7 presents the aggregated results of the questionnaire. Each question is referenced by a unique ID, with the corresponding full text of the item provided in Table 5.6. The table includes the number of responses per rating option, the average scores for each rubric item, and the converted values based on a standardized 3-point scale. The details of the conversion process are described in Table 5.5.

The rubric items “Knowledge Decomposition” and “Graph-based Navigation Usability” received the highest average scores (4.41), with standard deviations of 0.71 and

Table 5.6: Questionnaire Items, response scale: strongly disagree (1) to strongly agree (5)

Criterion	Questionnaire item
Knowledge decomposition	1. The system clearly breaks down the learning material into smaller, understandable units.
	2. I understood how the different parts of the learning material were related to each other.
Prerequisite logic clarity	3. The system provided appropriate learning path suggestions.
	4. It was clear to me which topics were prerequisites for others.
Graph-based navigation usability	5. The graph made it easy for me to navigate what to study next.
	6. Overall, I found the system's structure to be useful.
Support for learner decision-making	7. The graph-based representation helped me understand the structure of the learning material.
	8. I felt that the system considered the evolution of my knowledge over time.
Overall (open ended)	9. What suggestions do you have for further developing the system?

0.78, respectively. These correspond to level 3 on the 3-point rubric scale. The item "Support for Learner Decision-Making" received a slightly lower average score of 4.21 (SD = 1.02), corresponding to level 2. The lowest-rated item was "Prerequisite Logic Clarity," with an average score of 4.16 (SD = 0.91), corresponding to level 2.

The items of highest rating, 'Knowledge Decomposition' and 'Graph-based navigation usability', in Table 5.7 suggest that the structural and visual design features of the system were well received by the participants. These components likely offered intuitive guidance, even for users with limited prior exposure to intelligent tutoring systems, as the confidence score shows in Table 5.2. In the longer term, these results suggest the model's adaptability and potential for rapid deployment. Given that the system provides a highly visual and graph-based structure for navigating knowledge, the strong ratings align well with its intended design. In addition, dedicated practice time and instructor guidance may have contributed to the learners' positive perceptions.

The average score for "Support for Learner Decision-Making" in Table 5.7 reflects more varied experiences. Although many learners valued the autonomy provided

Table 5.7: Questionnaire Items, response scale: strongly disagree (1) to strongly agree (5)

Criterion	ID	1	2	3	4	5	AVG	Score
Knowledge decomposition	1	0	0	1	11	16	4.41	3
	2	0	1	3	11	13		
Prerequisite logic clarity	3	1	0	7	13	7	4.16	2
	4	0	2	0	10	16		
Graph-based navigation usability	5	0	1	2	14	11	4.41	3
	6	0	1	2	5	20		
Support for learner decision-making	7	0	1	3	10	14	4.21	2
	8	1	3	3	6	15		

by the system as [103] also suggests, the larger spread of responses indicates that some may have required additional guidance or scaffolding. This variability probably results from differences in self-regulation skills, also emphasized by Zhang et al. in [120], learning preferences, or prior domain knowledge [73].

Similarly, the relatively lower score for "Prerequisite Logic Clarity" in Table 5.7 may indicate challenges experienced by learners with less background in interpreting conceptual hierarchies or graph structures as states [108].

This finding is in line with [121], that highlighted the importance of designing for a diverse range of cognitive and experiential profiles, possibly through multimodal explanations or adaptive visual supports suggested also in [36].

Taken together, these results emphasize the need to accommodate heterogeneity among learners, not only in content delivery, but also in system capabilities that support different levels of experience, confidence, and strategic learning behavior, also empasized in [105].

5.7 Learner performance and progression using different adaptive learning algorithms

To address RQ2, a multifaceted evaluation approach is employed.

First, I compared changes in individual knowledge unit mastery between initial and

final test sessions between different groups. The mastery level for a knowledge unit was calculated as the ratio of correct answers to all attempts, yielding a value between 0 and 1. Table 5.9 shows the analysis focused on a subset of knowledge units included in both evaluation phases.

Then, I evaluated the general knowledge state of the curriculum by adding the mastery levels of the 15 knowledge units. This aggregate measure was tracked from the initial to the final state. Additionally, during the study, I used offline quiz sheets to assess the average knowledge retention within each group, complementing the system's internal data. The results are in Table 5.10.

The conversion logic used to translate the three key metrics into the final rubric score is summarized in Table 5.8.

Table 5.8: Conversion table for RQ2 rubric

Rubric Score	Mastery change (%)[a]	Knowledge (max. 15)[b]	Knowledge retention (%)[c]
1	≤ 5	≤ 5	≤ 40
2	$6 - 20$	$6 - 10$	$41 - 70$
3	≥ 21	≥ 11	≥ 71

[a] Difference between the final and initial measured knowledge levels.

[b] Final aggregate knowledge state of the curriculum at the end of test sessions.

[c] The difference between the final measured knowledge level and the result of the offline examination, providing an external measure of long-term retention.

Comparison of initial and final test data indicates notable improvements in all groups (Table 5.9). The measured knowledge state increased in each group, with the BN group improving from 0.72 to 0.89, and the KST group from 0.92 to 0.97. The predicted knowledge state, derived from the learner model, showed a more pronounced change, especially for the BN and WDDI groups, suggesting enhanced inferred mastery levels over time. It is worth noting that the predicted knowledge state for the KST group remained at 1.000 throughout. This behavior reflects a key characteristic of the KST model: upon successful completion of an item associated with a know-

ledge unit, the model infers full mastery (1.000) not only for that unit, but also for all its prerequisite units in the predicted mastery values. Consequently, these values should be interpreted with caution, since these values do not reflect knowledge gain as observed in other models.

Table 5.9: Group-level changes in measured and predicted knowledge state from initial to final test sessions

Group	Initial		Final		Result	
	Measured	Predicted	Measured	Predicted	Change (%)	Score
BN	0.717	0.243	0.887	0.463	24	3
KST	0.919	1.000*	0.967	1.000*	5	2
WDDI	0.871	0.126	0.882	0.298	1	1

* KST predictions should be interpreted with caution: the model assigns full mastery to all prerequisite units once a higher-level unit is successfully completed.

I evaluated the overall mastery of the curriculum by aggregating the knowledge levels across all 15 knowledge units, which gave us a total of 15 as the total knowledge value. This cumulative metric allowed me to compare the progress of the learners from the initial to the final state within the system. To complement the system-generated data, I also distributed offline quiz sheets during the final examination to measure overall knowledge retention in each group. A summary of these results is presented in Table 5.10, showing both system-tracked mastery progression and externally assessed retention levels.

Table 5.10: Group-level changes in measured and predicted total knowledge state from initial to final test sessions and during exam (the curriculum's total knowledge value is 15.000)

Group	Initial		Final			Examination	
	Measured	Predicted	Measured	Predicted	Score	Offline	Score
BN	0.800	1.034	11.200	7.156	3	11.730	3
KST	6.600	8.000	15.000	10.000	3	10.000	2
WDDI	1.000	1.971	14.000	6.072	3	10.000	3

The three adaptive learning algorithms effectively supported learner knowledge state improvement, as detailed in Table 5.9. Munir et al. [72] also underlined that AI algorithms are used effectively in digital education. My results in Table 5.9 show

that the BN group exhibited the most substantial gain, with the measured knowledge state level increasing from 0.717 to 0.887, an improvement 24%. The other two groups also showed positive, albeit more moderate, gains. It is interesting to add the fact that the BN group began with moderate prior familiarity and average online confidence levels, as shown in Table 5.2. In contrast, the KST group started from a relatively high mastery level of 0.919, which still increased by 5%, suggesting the model's ability to support further learning even at advanced stages, as we could anticipate it based on [7]. Although the predicted values deviated from the measured values, both indicators reflected a consistent upward trend that was expected based on [28].

Further analysis of the value of the aggregated state of knowledge at the curriculum level revealed different learning trajectories [32]. As we can see in Table 5.10 the BN group improved from an initial composite score of 0.8 to 11.2. The WDDI group progressed from 1.0 to 14.0. while the KST group advanced from an already strong baseline of 6.6 to full curriculum mastery at 15.0. Regarding mid-term retention, the results of the offline exam indicated strong knowledge preservation in both the BN and WDDI groups. As shown by [22] knowledge retention typically declines over time. The study measured retention only within a limited mid-term window of a few weeks. In terms of knowledge retention, the KST group performed slightly below the other two, possibly due to ceiling effects associated with their already high measured test levels. The ceiling effects, studied in [102] and [113] are observed when participants start with very high scores, leaving little room for measurable improvement.

These performance differences may be explained by the variation in learner behavior between groups. For example, in Figure 5.3 we can observe significant differences in the time spent on the test sheets [63] and the total duration of engagement [52] in Figure 5.4. Furthermore, some learners practiced systematically with each knowledge unit, while others concentrated on a smaller subset, relying on repeated tests as shown in Figure 5.5. These distinct behavioral patterns probably contributed to the variation in group-level learning outcomes.

5.8 Real-time tracking of learner activity and knowledge state progression

To address RQ3, the system's ability to record and maintain real-time data related to learners' actions and transitions of knowledge states was analyzed.

To analyze learner interaction patterns, I extracted event logs from the `LEARNER_LOG` database table. Events were categorized based on their `event_name` field (e.g., `learning.started`, `testing.submitted`). I grouped the data by the algorithm used (BN, KST, WDDI), and calculated the average number of occurrences per learner for each event type. Moreover, I collected the number of quiz attempts and the number of knowledge state records to give a better overview of the system's ability of following the learners' activity. The results can be studied in Table 5.11 and Table 5.12.

To analyze the participation of learners in testing activities, I extracted log events related to test interactions (`testing.started`, `testing.submitted`, `testing.finished`). The events were grouped by learner and knowledge unit, and the time difference between the starting event and the nearest subsequent submission or completion event was calculated. Sessions lasting longer than the 99th percentile were considered outliers and excluded from the analysis to reduce noise caused by technical interruptions or idle time. The remaining durations were analyzed and visualized using a boxplot in Figure 5.3 to compare the distributions between the algorithms.

To assess the duration of learners' engagement with learning materials, I analyzed events logged during learning sessions. I extracted `learning.started` and `learning.finished` events from the `LEARNER_LOG` table and grouped them by learner and knowledge unit. The time spent was calculated as the difference between the start and finish events. Sessions without a valid end were excluded. To reduce the noise caused by inactive sessions or system errors, I discarded durations above the 95th percentile. The remaining data was analyzed to compare the learning engagement between the three groups and visualized using a boxplot in Figure 5.4.

To assess the level of participation of the learners in terms of active testing, I used the number of `knowledge_state.updated` events of type `measure` as a proxy indicator. I compute both the total number of updates per group and the average number of updates per learner within each group. In addition, I identified the most active learner in each group by counting individual-level update frequencies. This metric reflects the intensity of the testing activity and can be interpreted as a behavioral indicator of active learning, as Table 5.13 shows.

To examine the degree of knowledge acquisition over the course of the experiment, I compared the initial and final tests of the learners. I used two indicators for each group:

- **Measured knowledge state:** the proportion of correctly answered test items.
- **Predicted knowledge state:** the estimated mastery level inferred from the learner model.

The initial knowledge state was calculated based on the learners' earliest testing activities, typically covering a smaller subset of knowledge units. The final knowledge state was derived from the most recent tests at the end of the intervention period, generally incorporating a wider range of knowledge units. The averages at the group level for both indicators were calculated at both time points, Table 5.9.

To assess how participants responded to the personalized content suggestions of the system, the log entries labeled `recommendation.requested` were analyzed. These events indicate that the system provided a recommendation to the learner, either for new content (learning) or for review (repetition). To determine whether a recommendation was followed, the learner's next action was examined after the recommendation event. A recommendation was counted as followed if the first subsequent interaction of the same learner was aimed at the recommended KU and involved starting or repeating a learning session. This approach captures whether the learner chose to engage with the suggested content as their immediate next step, without applying any strict time limit. The total number of recommendation events and the average number per learner were calculated for each group. In addition, I calculated the proportion of learning and repetition recommendations that were followed, in-

dicating how well the system's suggestions aligned with user behavior. According to Table 5.14, most learners disregarded the recommendations in both cases, indicating that self-regulated decision-making took precedence over the suggested path. To analyze how often the learners interacted with the recommendation feature of the system, I examined the frequency of the same events.

The learning paths of two selected learners are analyzed based on their interactions with specific knowledge units. The extracted dataset from the `KNOWLEDGE_STATE` table included timestamped records of each learner's attempt on knowledge units, identified by their external IDs. The learning paths are visualized in Figure 5.5 as stepwise sequences of knowledge units. Instead of plotting absolute timestamps, the x-axis represented the sequential order of the learner's interaction steps, allowing comparison of learning progression patterns without time-scale distortion. The y-axis enumerated all KU in the topic, ensuring visibility of units even if a learner skipped them.

In order to assess the system's overall capability to support real-time tracking of learner activity (RQ3), a feature-based rubric analysis was conducted. The key dimensions of learning activity tracking are listed, such as event tracking, time tracking, recommendation adherence, and visualization of the learning trajectory, and the availability of these in current implementation. Each fulfilled capability received a checkmark. Based on the total number of possible features, a performance score was calculated by dividing the number of completed features by the total. If the system covered more than two thirds of the features, it received a score of Good (3); if between one third and two thirds, Fair (2); and if below one third, Poor (1). This scoring method enables a transparent and reproducible evaluation of tracking quality across learning systems, Table 5.15.

Table 5.11 provides an overview of the experimental groups, summarizing user activity and data during system usage. The metrics include the average number of log records per learner, the average number of knowledge state updates per learner, and the average number of quiz attempts per learner associated with each group.

Table 5.12 summarizes the average number of events per learner for each group.

Table 5.11: Overview of experimental groups and collected interaction data (aggregated per learner)

Group	Log records	Knowledge state records	Quiz attempts
BN	855.9	445.7	29.7
KST	505.1	134.9	27.6
WDDI	457.9	230.4	14.4

Table 5.12: Distribution of event types per learner across algorithms

Event name	BN	KST	WDDI
knowledge_state.attempt.requested	69.08	52.53	30.80
knowledge_state.updated	535.17	156.07	241.67
learning.started	21.20	14.90	10.83
learning.finished	14.40	12.50	11.64
page.requested	13.79	13.69	12.20
recommendation.requested	40.23	35.75	21.40
testing.started	43.58	33.80	18.00
testing.submitted	35.83	30.40	15.27
testing.finished	43.25	33.53	16.80
test_state.updated	168.92	148.80	74.53
topic.changed	3.85	3.63	2.87
topics.requested	7.46	7.25	6.13
user.profile.created	1.00	1.00	1.00

Figure 5.3 presents the distribution of the time learners spent on the test sheets, grouped by algorithm. The boxplot illustrates notable differences in test duration between groups. While most learners completed their test sessions within a consistent range, a few outliers were removed to enhance comparability.

Figure 5.4 illustrates the distribution of time learners spent on active learning sessions per knowledge unit, grouped by algorithm. After removing extreme values beyond the 95th percentile, the WDDI group showed the shortest median learning time, while the KST group exhibited the longest.

The average number of `knowledge_state.updated` events per learner was highest in the BN group, indicating the most frequent engagement with test activities Table 5.13. The most active learner, a member of the KST group, demonstrated a high level of engagement with 112 interactions.

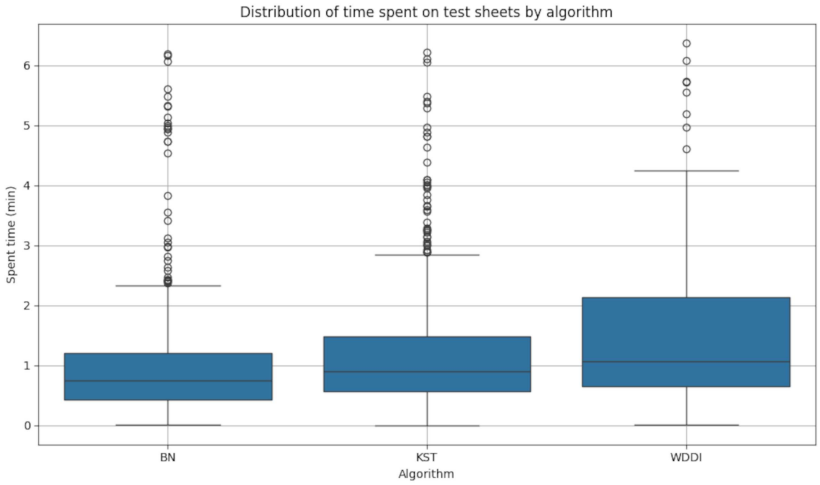


Figure 5.3: Distribution of time spent on test sheets by algorithm (outliers above 99th percentile removed)

Table 5.13: Testing activity across groups (no. of knowledge state update events)

Group	Average per learner	Most active learner	Most tested unit (no. of test submissions)
BN	34.7	95	DqlCommands (53)
KST	29.5	112	ConnectionEstablishment (43)
WDDI	15.4	58	SqlApiDescription (25)

The frequency of recommendation requests varied between the experimental groups. Table 5.14 presents the total number of `recommendation.requested` events, the average number per learner, and the percentage of learning and repetition recommendations followed.

Table 5.14: Recommendation usage statistics by group, including total and average request counts, and the percentage of followed learning and repetition recommendations

Group	Total	Per learner	Followed learn (%)	Followed repeat (%)
BN	523	40.23	18	8
KST	572	35.75	32	9
WDDI	321	21.40	29	7

Figure 5.5 depicts the learning trajectories of two selected learners. Using the step order on the x-axis, the plots highlight the sequence in which learners engaged with

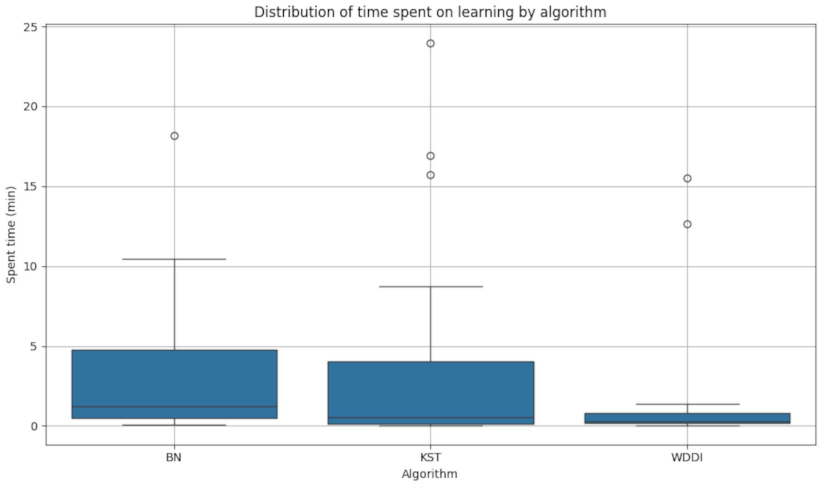


Figure 5.4: Distribution of time spent on learning sessions by algorithm (values above 95th percentile excluded)

the material, revealing patterns such as skipped units or revisits. All knowledge units were displayed on the y axis regardless of whether learners attempted them, providing a comprehensive overview of the topic structure and individual learner coverage. These findings demonstrate the utility of visualization of the sequential learning pathway in identifying learner behaviors and informing targeted educational interventions.

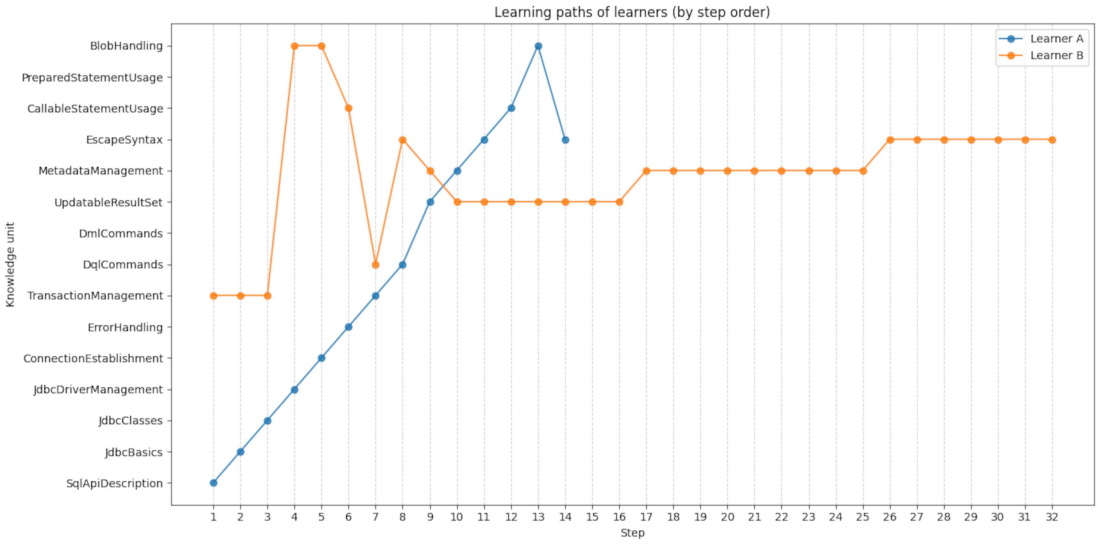


Figure 5.5: Learning path of two randomly selected learners

As shown in Table 5.15, the G4L system supports 11 tracked capabilities, resulting

in a fulfillment ratio of 85%. Based on the rubric tiered scoring logic (score = 3 if $> 66\%$), the system receives a learner activity tracking score of Good (3).

Table 5.15: Learner activity tracking capabilities of the G4L system. A check mark (✓) indicates that the feature is supported, an x mark (✗) otherwise.

Feature	Supported
User activity logging	✓
System usage tracking	✓
Event-level interaction data	✓
Time spent on test sheets	✓
Time spent in active learning sessions	✓
Most active learner identification	✓
Most frequently learned knowledge unit	✓
Knowledge gain measurement	✓
Adherence to learning recommendations	✓
Adherence to repetition recommendations	✓
Learning trajectory visualization	✓
Wrong answer tracking	✗
Recommendation response time tracking	✗
Total fulfilled features	11 / 13

Real-time tracking of learner activity and knowledge state progression is the cornerstone of modern educational technology [99], allowing personalized learning experiences and supporting improved educational outcomes. Numerous advanced models have been developed to monitor various dimensions of learner behavior and cognitive progress [104, 116, 29]. The G4L system relies exclusively on activity logs, which offer precise, time-stamped data on user interactions, such as time spent on learning and testing tasks, as well as the sequence in which knowledge units are accessed. In Table 4.4 there are 13 types of events that the G4L system is actively tracking.

Relying solely on log data provides several practical advantages. First, it eliminates the need for intrusive or additional assessment mechanisms, such as [88] or [45], making the system lightweight and scalable. Second, log data are objective and automatically generated, reducing the potential for human error or bias in data collection. It also allows for fine-grained analysis of behavior patterns, such as time on task, repetition cycles, and preferred learning paths, which are valuable for understanding engagement and self-regulation. Granularity is demonstrated by Table 5.12.

However, this approach also has notable limitations. Log data alone cannot capture the cognitive states, motivations, or emotional responses of learners, as emphasized in [45], factors that often influence learning success. Moreover, while activity patterns can correlate with learning outcomes, they do not necessarily provide causal information or explain why certain behaviors lead to better results [88]. Finally, as [97] showed, systems that rely solely on logs may overlook off-platform learning or unrecorded metacognitive strategies that contribute to overall knowledge acquisition.

Despite these limitations, the log data revealed that the G4L system effectively captured a wide range of learner interactions in real time. As Table 5.11 shows, with an average of more than 855 log records per learner in the BN group, over 505 in the KST and over 457 in the WDDI groups, the implementation demonstrated a strong capacity to track not only the usage and testing activity of the learners, but also their navigation patterns and evolving knowledge states. According to the learning activity tracking rubric detailed in Table 5.15, the system clearly qualifies as Good (3), providing detailed information on learning behavior in real time and progression of the state of knowledge. This level of detail enables a meaningful analysis of learning patterns and supports adaptive educational interventions.

5.9 Learner feedback and suggestions

The feedback of the learners on the G4L system was predominantly positive. Participants particularly appreciated the innovative and transparent graph-based structure, describing it as both engaging and easy to navigate (e.g., "The graph solution really appealed to me; it is innovative and easy to overview.").

However, several recurring suggestions emerged that are summarized in Table 5.16. These insights will inform ongoing system development, with particular attention to accessibility, feedback mechanisms, and user-centered design.

The use of a G4L GenAI assistant successfully supported prerequisite identification and knowledge decomposition, as reflected in the positive feedback presented in Table 5.7.

Table 5.16: Learner feedback and suggestions

Area	Summary
Accessibility	A common request was to allow access beyond the university network to improve convenience and usability.
Feedback and Error Awareness	Learners emphasized the need for more detailed test feedback, including information on incorrect responses and correct answers, to better support reflective learning.
Content and Assessment	Suggestions included expanding learning materials and increasing the number of test items to improve coverage and assessment reliability.
User Interface and Experience	Feedback pointed to UI/UX improvements, including clearer layouts, better component sizing, and refinement of input validation (e.g., instances where correct answers were not accepted).
Graph Visualization	One user noted that excessive edge crossings occasionally reduced clarity, indicating a need for optimization in graph layout.

5.10 Thesis 4.

An empirical investigation involving Bachelor of Science students demonstrated that an intelligent tutoring system based on the Evolving Knowledge Space Graph model and implemented through the Graph4Learn framework effectively supports adaptive and self-regulated learning in rapidly evolving domains.

Regarding RQ1, the results confirm that the EKSG model provides a robust structure to represent domain knowledge and support the navigation of the learner.

In response to RQ2, the comparative evaluation revealed that the Bayesian Network approach achieved the highest learning gains among the adaptive algorithms considered.

Finally, addressing RQ3, the study validated that the system enables real-time tracking of learner activity and knowledge state progression, improving adaptability and personalization.

5.11 Author's publications related to the thesis

Q1, preprint, under review: [10]

[11], [1]

Chapter 6

Summary

This dissertation introduces the Knowledge-Sharing-Bridge framework, which embeds an implicit tutoring mechanism directly into Artificial Intelligence systems. The framework consists of four interrelated modules: *Explain*, *Report*, *Control*, and *Teach*, that collectively enable AI to function not only as a decision-support tool but also as an active pedagogical agent. Powered by an Explainable AI engine, the *Explain* and *Teach* components demonstrate how explainability can be effectively combined with instructional guidance. The KSB framework integrates structured knowledge representations, such as word clouds, to enhance the transparency, accessibility, and pedagogical value of complex decision logic.

A prototype implementation validates the hypothesis that embedding knowledge-sharing mechanisms within AI systems bridges the gap between passive technology use and active human learning. The findings indicate that when explainability is coupled with guided instruction, AI systems not only improve task performance but also foster user comprehension, engagement, and long-term knowledge retention.

In a broader sense, this work reconnects with the ancient roots of learning, where knowledge was continuously shared between the wise and the curious. With the advent of Artificial Intelligence, such continuous, dialogic learning becomes possible again, this time through intelligent systems capable of explaining, guiding, and teaching.

In addition to the KSB framework, another major contribution of this research is the development of an Intelligent Tutoring System capable of adapting to rapidly

evolving curricula, such as those in computer science, while also supporting individual learner progression and self-regulated learning. The implemented Graph4Learn system was designed to meet the diverse needs of university students and adult learners.

The implementation is based on the Extended Knowledge Space Graph model, a graph-based knowledge representation framework also developed as part of this research. A relational database was integrated to enable real-time learning tracking and support adaptive algorithm decision making. The system introduced several innovations: (1) abstract temporal dependency to model evolving curricula, (2) adapted IFL to represent learner knowledge states and forgetting, (3) multiple adaptive learning algorithms, including our proposed WDDI method, and (4) the integration of a GenAI-based assistant to generate curriculum content, comprising knowledge units, prerequisite relationships, and quiz items.

The findings of the study indicate that the EKSG model effectively structured domain knowledge and supported learner navigation. Real-time tracking of both learner activity and knowledge progression was achieved successfully. Furthermore, our evaluation of adaptive algorithms in a coarse-grained curriculum setting revealed that the BN algorithm yielded the highest overall learning gains, outperforming both the KST and WDDI models.

Together, these findings highlight both the practical viability and theoretical relevance of the G4L system and the underlying EKSG model. By combining structured domain modeling, adaptive learning algorithms, and fine-grained learning activity tracking, the system offers a flexible and scalable solution to support personalized learning. These insights underscore the significance of graph-based representations and log-based analytics in the development of future intelligent tutoring systems.

6.1 Thesis 1.

I introduced the Knowledge-Sharing-Bridge framework that embeds an implicit tutoring mechanism directly into Artificial Intelligence systems, which consists of four major subcomponents: *Explain*, *Report*, *Control*, and *Teach*. The modules *Ex-*

plain and *Teach*, supported by a dedicated Explainable Artificial Intelligence engine, demonstrate the feasibility of combining explainability with pedagogical guidance. The framework shows potential for applications across domains where human-AI collaboration is critical, including education, corporate training, and technical support. The framework incorporates structured knowledge representations, such as word clouds, that make complex decision logic more transparent, accessible, and pedagogically meaningful. The prototype implementation provides empirical support for the hypothesis that a knowledge-sharing component integrated into Artificial Intelligence systems can bridge the gap between passive consumption of technology and active human learning. The results demonstrate that explainability, when coupled with guidance, not only enhances task outcomes but also empowers users by making AI-driven decisions more comprehensible, actionable, and conducive to sustained knowledge retention.

6.2 Thesis 2.

I introduced the Evolving Knowledge Space Graph model, which integrates the Knowledge Space Theory with ontological structures, while also incorporating the notion of abstract time, larger-grained knowledge units, and flexible connections to external knowledge representations through evoking-hooks. The Evolving Knowledge Space Graph model enables the representation of evolving domains where multiple parallel truths may coexist (e.g., different software versions) and where knowledge differences and knowledge complexity can be quantified. By integrating these extensions, the Evolving Knowledge Space Graph provides a more realistic and adaptable framework to model knowledge evolution, learner adaptivity, and personalized learning paths in dynamic educational and professional contexts.

6.3 Thesis 3.

I developed the Graph4Learn system, grounded in the Evolving Knowledge Space Graph model, to integrate graph-based knowledge representation with a relational database, thus enabling real-time tracking of learner activity and adaptive learning

path discovery. The system advances the state of the art by introducing the following key innovations. First, it implements an abstract time dependency to capture the dynamics of evolving curricula. Second, it adapts the Intuitionistic Fuzzy Logic model to represent learner knowledge states and forgetting. Third, it implements multiple knowledge state induction algorithms, including the proposed Weighted Distance Dependent Induction method to predict the knowledge state. Fourth, it applies an Intuitionistic Fuzzy Logic based algorithm for the recommendation of the adaptive learning path. Finally, it integrates a Generative AI powered assistant capable of generating curriculum content such as knowledge units, prerequisite relations, and assessment items.

6.4 Thesis 4.

An empirical investigation involving Bachelor of Science students demonstrated that an intelligent tutoring system based on the Evolving Knowledge Space Graph model and implemented through the Graph4Learn framework effectively supports adaptive and self-regulated learning in rapidly evolving domains.

Regarding RQ1, the results confirm that the EKSG model provides a robust structure to represent domain knowledge and support the navigation of the learner.

In response to RQ2, the comparative evaluation revealed that the Bayesian Network approach achieved the highest learning gains among the adaptive algorithms considered.

Finally, addressing RQ3, the study validated that the system enables real-time tracking of learner activity and knowledge state progression, improving adaptability and personalization.

6.5 Limitations of the dissertation

The research acknowledges limitations, including the high-level description of the KSB framework and the use of a small group of university students for the preliminary evaluation of the framework.

Although the proposed EKSG model and the G4L framework demonstrate promising results, the scope of the empirical evaluation was limited to a specific educational context and a learner population. Furthermore, the implemented system represents a prototype-level realization and additional large-scale validation would be required to assess long-term effectiveness across diverse domains.

6.6 Future research directions

Future work will focus on extending the application of the Graph4Learn system to multiple courses, one of them is currently under development and implementation.

Based on the experience gained so far, further refinements of the learner tracking database are also planned. In particular, modifications will be introduced to improve the representation and traceability of individual learning sessions. Although the current framework already allows for the quantitative evaluation of learning gains through changes in learners' knowledge states, future extensions will aim to formalize additional metrics related to learning engagement and learning efficiency. In particular, navigation patterns within the knowledge graph, deviations from recommended learning paths, and temporal characteristics of learning sessions provide a strong foundation for defining explicit engagement and efficiency indicators.

Finally, a key direction for future development is the integration of a tutoring dialog assistant. This assistant will leverage the underlying graph structure and assessment questions to provide context-aware hints and explanations, thereby enhancing learner support and interaction within the system.

6.7 Concluding remarks

This dissertation aimed to address the overarching research question of how an intelligent tutoring framework can support adaptive, personalized, and sustainable learning in fast-evolving knowledge domains while preserving the central role of educators.

The proposed Evolving Knowledge Space Graph model and the Graph4Learn frame-

work demonstrate that dynamic curriculum representation, actionable learner modeling, and adaptive learning strategies can be effectively combined within a single system. Through empirical evaluation, the work shows that such a framework can support learner progression, enable educator insight through transparent learner tracking, and remain robust amid continuous domain evolution.

Together, these contributions provide a coherent answer to the ORQ and establish a foundation for future research on intelligent tutoring systems.

Author's publications

[1] L. Csépányi-Fürjes, 'Performance analysis of low dimensional word embeddings to support green computing,' *PRODUCTION SYSTEMS AND INFORMATION ENGINEERING*, vol. 10, pp. 27–36, 2022 (cit. on p. 94).

[2] L. Csépányi-Fürjes, 'Szövegosztályozó módszerek vizsgálata ügyfélszolgálati kérések előfeldolgozásához,' in *Elemző és robotizált folyamatautomatizálási rendszer fejlesztése nagy terhelésű ügyfélszolgálatok számára : Kutatási jelentések 2022/1*, L. Kovács, Ed. Miskolci Egyetemi Kiadó, 2022, pp. 124–147 (cit. on p. 33).

[3] L. Csépányi-Fürjes, 'Controllable and explainable ai framework in the automatic assessment domain,' in *12th International Conference on Applied Informatics (ICAI 2023)*, Eszterházy Károly Katolikus Egyetem, 2023, pp. 1–3 (cit. on p. 33).

[4] L. Csépányi-Fürjes, 'Ügyfélszolgálati megkeresések automatikus feldolgozásának lehetőségei emberközpontú mesterségesintelligencia-módszerekkel,' in *Mesterségesintelligencia-kutatások a Miskolci Egyetemen : Elemző és robotizált folyamatautomatizálási rendszer fejlesztése nagy terhelésű ügyfélszolgálatok számára*, L. Kovács, Ed. Miskolci Egyetemi Kiadó, 2024, vol. 1, pp. 146–158 (cit. on p. 33).

[5] L. CSÉPÁNYI-FÜRJES and L. KOVÁCS, 'Capturing knowledge dynamics in the evolving knowledge space graph using temporal relations and knowledge aging,' *Acta Marisiensis. Seria Technologica*, vol. 22, pp. 14–22, 2 Dec. 2025, ISSN: 26684217. DOI: 10.62838/amset-2025-0012. [Online]. Available: https://amset.umfst.ro/?pag=vols/2025-2&doi=10.62838/amset-2025-0012 (cit. on p. 70).

[6] L. Csépányi-Fürjes and L. Kovács, 'Efficiency analysis of deeplearning4j neural network classifiers in development of transition based dependency parsers,' *ACTA MARISIENSIS. SERIA TECHNOLOGICA*, vol. 18, pp. 33–39, 1 2021 (cit. on p. 70).

[7] L. Csépányi-Fürjes and L. Kovács, 'Evolving graph based knowledge space model for tutoring systems,' *Pollack Periodica*, Jul. 2024, SJR: Q3, ISSN: 1788-1994. DOI: 10.1556/606.2024.01058. [Online]. Available: https://akjournals.com/view/journals/606/aop/article-10.1556-606.2024.01058/article-10.1556-606.2024.01058.xml (cit. on pp. 52, 69, 84).

[8] L. Csépányi-Fürjes and L. Kovács, 'Intelligens oktatórendszerek gráf alapú tudásállapot vizualizációja,' in *Oktatás - Informatika - Pedagógia 2024*, A. Buda, Ed., DE/BTK/Nevelés- és Művelődéstudományi Intézet, 2024, pp. 20–20 (cit. on p. 52).

[9] L. Csépányi-Fürjes and L. Kovács, 'Generalized domain tutoring framework for ai agents with integrated explainable ai techniques,' *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 40, p. 860, 2 Nov. 2025, SJR:Q4, ISSN: 2502-4760. DOI: 10.11591/ijeecs.v40.i2.pp860-870. [Online]. Available: https://ijeecs.iaescore.com/index.php/IJEECS/article/view/40674 (cit. on p. 33).

[10] L. Csépányi-Fürjes and L. Kovács, 'Intelligent tutoring in dynamic domains: A graph-based system for comparative analysis of adaptive algorithms with intuitionistic fuzzy logic and forgetting,' *Educational Technology Research and Development*, 2026, SJR:Q1, preprint, under review (cit. on p. 94).

[11] L. Csépányi-Fürjes and L. Vass, 'Megfelelően alkalmazva a generatív mesterséges intelligencia eszközök fejlesztik a hallgatók problémamegoldó képességeit,' *PRODUCTION SYSTEMS AND INFORMATION ENGINEERING*, vol. 12, pp. 60–71, 3 2024 (cit. on p. 94).

[12] L. Kovács and L. Csépányi-Fürjes, 'Feature reduction for dependency graph construction in computational linguistics,' in *CEUR Workshop Proceedings*, vol. 2650, 2020 (cit. on p. 52).

[13] L. Kovács, L. Csépányi-Fürjes and G. H. A. Ahmed, 'Automated assessment generation in intelligent tutoring systems,' in *Lecture Notes in Networks and Systems*. 2022, vol. 386 LNNS, pp. 808–819, SJR: Q4. DOI: 10.1007/978-3-030-93817-8_72. [Online]. Available: https://link.springer.com/10.1007/978-3-030-93817-8_72 (cit. on p. 70).

[14] L. Kovács, L. Csépányi-Fürjes and W. Tewabe, 'Transformer models in natural language processing,' University of Miskolc, Tech. Rep., 2024. DOI: 10.1007/978-3-031-54674-7_14 (cit. on p. 70).

References

[15] D. Albert, C. Hockemeyer, M. D. Kickmeier-Rust, A. Nussbaumer and C. M. Steiner, 'E-Learning Based on Metadata, Ontologies and Competence-Based Knowledge Space Theory,' in *Communications in Computer and Information Science*, vol. 295 CCIS, 2012, pp. 24–36. DOI: 10.1007/978-3-642-32826-8_3. [Online]. Available: http://link.springer.com/10.1007/978-3-642-32826-8_3 (cit. on p. 69).

[16] D. Albert and J. Lukas, *Knowledge Spaces: Theories, Empirical Research, and Applications*. 1999 (cit. on p. 15).

[17] A. Almasri *et al.*, 'Intelligent tutoring systems survey for the period 2000-2018,' *International Journal of Academic Engineering Research*, vol. 3, pp. 21–37, 2019. [Online]. Available: http://dstore.alazhar.edu.ps/xmlui/handle/123456789/456 (cit. on p. 8).

[18] J. R. Anderson, C. F. Boyle and B. J. Reiser, 'Intelligent Tutoring Systems,' *Science*, vol. 228, no. 4698, pp. 456–462, Apr. 1985, ISSN: 0036-8075. DOI: 10.1126/science.228.4698.456 (cit. on p. 9).

[19] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold and P. M. Atkinson, 'Explainable artificial intelligence: an analytical review,' *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 5, 2021, ISSN: 19424795. DOI: 10.1002/widm.1424 (cit. on p. 18).

[20] S. Assegaff, Kurniabudi and E. Fernando, 'Impact of extrinsic and intrinsic motivation on knowledge sharing in virtual communities of practices,' *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 1, no. 3, pp. 619–626, Mar. 2016, ISSN: 25024760. DOI: 10.11591/ijeecs.v1.i3.pp619-629 (cit. on p. 22).

[21] K. T. Atanassov, 'Intuitionistic fuzzy sets,' *Fuzzy Sets and Systems*, vol. 20, no. 1, 1986, ISSN: 01650114. DOI: 10.1016/S0165-0114(86)80034-3 (cit. on pp. 17, 56).

[22] D. R. Bacon and K. A. Stewart, 'How Fast Do Students Forget What They Learn in Consumer Behavior? A Longitudinal Study,' *Journal of Marketing Education*, vol. 28, no. 3, pp. 181–192, Dec. 2006, ISSN: 0273-4753. DOI: 10.1177/0273475306291463. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0273475306291463 (cit. on p. 84).

[23] M. Badaracco and L. Martínez, 'An Intelligent Tutoring System Architecture for Competency-Based Learning,' in 2011, pp. 124–133. DOI: 10.1007/978-3-642-23863-5_13 (cit. on p. 10).

[24] A. Badrinath, F. Wang and Z. Pardos, 'pyBKT: An Accessible Python Library of Bayesian Knowledge Tracing Models,' Tech. Rep., 2021. [Online]. Available: https://github.com/CAHLR/pyBKT (cit. on p. 62).

[25] W. Canfield, 'ALEKS: a Web-based intelligent tutoring system,' *Mathematics and Computer Education*, vol. 35, no. 2, pp. 152–158, 2001 (cit. on pp. 64, 69).

[26] D. Castelvecchi, 'Can we open the black box of AI?' *Nature*, vol. 538, no. 7623, 2016, ISSN: 14764687. DOI: 10.1038/538020a (cit. on p. 18).

[27] D. Chaves-Fraga, O. Corcho, F. Yedro, R. Moreno, J. Olías and A. De La Azuela, 'Systematic Construction of Knowledge Graphs for Research-Performing Organizations,' *Information*, vol. 13, no. 12, p. 562, Nov. 2022, ISSN: 2078-2489. DOI: 10.3390/info13120562. [Online]. Available: https://www.mdpi.com/2078-2489/13/12/562 (cit. on p. 70).

[28] F. Chen and Y. Cui, 'Utilizing student time series behaviour in learning management systems for early prediction of course performance,' *Journal of Learning Analytics*, vol. 7, no. 2, 2020, ISSN: 19297750. DOI: 10.18608/JLA.2020.72.1 (cit. on pp. 53, 84).

[29] M. Chen, K. Bian, Y. He, Z. Li and H. Zheng, 'Enhanced Learning and Forgetting Behavior for Contextual Knowledge Tracing,' *Information*, vol. 14, no. 3, p. 168, Mar. 2023, ISSN: 2078-2489. DOI: 10.3390/info14030168. [Online]. Available: https://www.mdpi.com/2078-2489/14/3/168 (cit. on p. 91).

[30] D. de Chiusole, L. Stefanutti, P. Anselmi and E. Robusto, 'Stat-Knowlab. Assessment and Learning of Statistics with Competence-based Knowledge Space Theory,' *International Journal of Artificial Intelligence in Education*, vol. 30, no. 4, 2020, ISSN: 15604306. DOI: 10.1007/s40593-020-00223-1 (cit. on p. 61).

[31] W. Clarysse and K. Vandorpe, *Information Technologies: Writing, Book Production, and the Role of Literacy*. Oxford University Press, Dec. 2009. DOI: 10.1093/oxfordhb/9780199734856.013.0029. [Online]. Available: https://academic.oup.com/edited-volume/28003/chapter/211776828 (cit. on p. 1).

[32] D. H. Clements and J. Sarama, 'Systematic review of learning trajectories in early mathematics,' *ZDM – Mathematics Education*, vol. 57, no. 4, pp. 637–650, Aug. 2025, ISSN: 1863-9690. DOI: 10.1007/s11858-024-01644-1. [Online]. Available: https://link.springer.com/10.1007/s11858-024-01644-1 (cit. on p. 84).

[33] E. Cosyn, H. Uzun, C. Doble and J. Matayoshi, 'A practical perspective on knowledge space theory: ALEKS and its data,' *Journal of Mathematical Psychology*, vol. 101, 2021, ISSN: 10960880. DOI: 10.1016/j.jmp.2021.102512 (cit. on p. 15).

[34] W. J. Courtenay, 'The Bible in medieval universities,' in *The New Cambridge History of the Bible*, Cambridge University Press, Apr. 2012, pp. 555–578. DOI: 10.1017/CHOL9780521860062.031. [Online]. Available: https://www.cambridge.org/core/product/identifier/CBO9781139050555A044/type/book_part (cit. on p. 2).

[35] D. van Dalen, 'Intuitionistic Logic,' in *The Blackwell Guide to Philosophical Logic*, Wiley, Aug. 2017, pp. 224–257. DOI: 10.1002/9781405164801.ch11. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/9781405164801.ch11 (cit. on p. 18).

[36] R. D. Deleña *et al.*, 'Predicting student retention: A comparative study of machine learning approach utilizing sociodemographic and academic factors,' *Systems and Soft Computing*, vol. 7, 2025. DOI: 10.1016/j.sasc.2025.200352. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105011178694&doi=10.1016%2fj.sasc.2025.200352&partnerID=40&md5=2f676f761e5139dc565187717674af8f (cit. on p. 81).

[37] S. Diaz Benavides, S. D. Cardoso, M. Da Silveira and C. Pruski, 'Analysis and implementation of the DynDiff tool when comparing versions of ontology,' *Journal of Biomedical Semantics*, vol. 14, no. 1, p. 15, Sep. 2023, ISSN: 2041-1480. DOI: 10.1186/s13326-023-00295-7 (cit. on p. 14).

[38] J. P. Doignon and J. C. Falmagne, 'Spaces for the assessment of knowledge,' *International Journal of Man-Machine Studies*, vol. 23, no. 2, 1985, ISSN: 00207373. DOI: 10.1016/S0020-7373(85)80031-6 (cit. on pp. 15, 50, 69, 75).

[39] A. Domínguez, J. Saenz-de-Navarrete, L. de-Marcos, L. Fernández-Sanz, C. Pagés and J.-J. Martínez-Herráiz, 'Gamifying learning experiences: Practical implications and outcomes,' *Computers & Education*, vol. 63, pp. 380–392, Apr. 2013, ISSN: 03601315. DOI: 10.1016/j.compedu.2012.12.020 (cit. on p. 11).

[40] H. Ebbinghaus, 'Memory: A Contribution to Experimental Psychology,' *Annals of Neurosciences*, vol. 20, no. 4, Oct. 2013, ISSN: 09727531. DOI: 10.5214/ans.0972.7531.200408. [Online]. Available: http://annalsofneurosciences.org/journal/index.php/annal/article/view/540 (cit. on p. 59).

[41] J.-C. Falmagne, E. Cosyn, J.-P. Doignon and N. Thiery, 'The assessment of knowledge, in theory and in practice,' Institute of Electrical and Electronics Engineers (IEEE), Mar. 2004, pp. 609–615. DOI: 10.1109/kimas.2003.1245109 (cit. on pp. 15, 16, 41, 58).

[42] G. Fenza, V. Loia and F. Orciuoli, 'Providing Smart Objects with Intelligent Tutoring Capabilities by Semantic Technologies,' in *2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, IEEE, Sep. 2016, pp. 103–109, ISBN: 978-1-5090-4124-4. DOI: 10.1109/INCoS.2016.110 (cit. on p. 12).

[43] P. Fournier-Viger, R. Nkambou and E. M. Nguifo, 'A data mining framework for the acquisition of procedural knowledge in intelligent tutoring systems,' in *Proceedings - ICCE 2008: 16th International Conference on Computers in Education*, 2008 (cit. on p. 13).

[44] H. A. A. Ghanim and L. Kovács, 'Ontology Supported Domain Knowledge Module for E-Tutoring System,' *Acta Cybernetica*, vol. 26, no. 3, pp. 455–474, Jul. 2024, ISSN: 2676-993X. DOI: 10.14232/actacyb.297804. [Online]. Available: https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/4316 (cit. on pp. 12, 50).

[45] L. Giraldi, M. Giovannetti and E. Cedrola, 'Unveiling emotional reaction and satisfaction in e-learning with face tracking,' *Educational Research and Evaluation*, pp. 1–33, May 2025, ISSN: 1380-3611. DOI: 10.1080/13803611.2025.2501530. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/13803611.2025.2501530 (cit. on pp. 91, 92).

[46] I. Š. Grgić, A. Grubišić, S. Stankov and M. Štula, 'An agent-based intelligent tutoring systems review,' *International Journal of Learning Technology*, vol. 14, no. 2, p. 125, 2019, ISSN: 1477-8386. DOI: 10.1504/IJLT.2019.101847 (cit. on p. 9).

[47] P. L. Hammer, U. N. Peled and X. Sun, 'Difference graphs,' *Discrete Applied Mathematics*, vol. 28, no. 1, pp. 35–44, 1990, ISSN: 0166218X. DOI: 10.1016/0166-218X(90)90092-Q (cit. on pp. 41, 42).

[48] Q. He, Y. Tang and Y. Z. Huang, 'Using OWL time ontology for workflow modeling and verification,' *Tongxin Xuebao/Journal on Communications*, vol. 27, no. 11, 2006, ISSN: 1000436X (cit. on p. 15).

[49] A. R. Hegde, A. S, A. C H, V. B V and C. Hegde, 'Enhanced Transformer: Knowledge Tracing with Incorporation of Temporal Features,' in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, IEEE, Mar. 2024, pp. 1–6, ISBN: 979-8-3503-6052-3. DOI: 10.1109/IATMSI60426.2024.10502970 (cit. on p. 16).

[50] H. Hegde *et al.*, 'A change language for ontologies and knowledge graphs,' *Database*, vol. 2025, Jan. 2025, ISSN: 1758-0463. DOI: 10.1093/database/baae133 (cit. on p. 14).

[51] F.-Z. Hibbi, O. Abdoun and E. K. Haimoudi, 'Knowledge Management in the Expert Model of the Smart Tutoring System,' in *Proceedings of the 3rd*

International Conference on Networking, Information Systems & Security, New York, NY, USA: ACM, Mar. 2020, pp. 1–4, ISBN: 9781450376341. DOI: 10.1145/3386723.3387895 (cit. on p. 13).

[52] D. L. Hoffman, F. Furutomo, A. Eichelberger and P. McKimmy, 'Matters of Frequency, Immediacy and Regularity: Engagement in an Online Asynchronous Course,' *Innovative Higher Education*, vol. 48, no. 4, pp. 655–677, Aug. 2023, ISSN: 0742-5627. DOI: 10.1007/s10755-023-09646-9 (cit. on p. 84).

[53] A. Ignatiev, 'Towards trustable explainable AI,' in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2021-January, 2020. DOI: 10.24963/ijcai.2020/726 (cit. on p. 18).

[54] T. Ivanova, 'Component based development of ontology-based intelligent tutoring systems,' in *2021 35th International Conference on Information Technologies, InfoTech 2021 - Proceedings*, 2021, ISBN: 9781665403245. DOI: 10.1109/InfoTech52438.2021.9548621 (cit. on p. 53).

[55] A. Johns, 'The coming of print to Europe,' in *The Cambridge Companion to the History of the Book*, Cambridge University Press, Dec. 2014, pp. 107–124. DOI: 10.1017/CCO9781139152242.009. [Online]. Available: https://www.cambridge.org/core/product/identifier/9781139152242%23c02373-934/type/book_part (cit. on p. 2).

[56] T. Kaser, S. Klingler, A. G. Schwing and M. Gross, 'Dynamic Bayesian Networks for Student Modeling,' *IEEE Transactions on Learning Technologies*, vol. 10, no. 4, 2017, ISSN: 1939-1382. DOI: 10.1109/tlt.2017.2689017 (cit. on p. 16).

[57] M. D. Kickmeier-Rust, N. Peirce, O. Conlan, D. Schwarz, D. Verpoorten and D. Albert, 'Immersive Digital Games: The Interfaces for Next-Generation E-Learning?' In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, PART 3, vol. 4556 LNCS, 2007, pp. 647–656. DOI: 10.1007/978-3-540-73283-9_71. [Online]. Available: http://link.springer.com/10.1007/978-3-540-73283-9_71 (cit. on p. 69).

[58] A. Klašnja-Milićević, B. Vesin, M. Ivanović and Z. Budimac, 'E-Learning personalization based on hybrid recommendation strategy and learning style identification,' *Computers & Education*, vol. 56, no. 3, pp. 885–899, Apr. 2011, ISSN: 03601315. DOI: 10.1016/j.compedu.2010.11.001. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360131510003222 (cit. on p. 11).

[59] M. Koopmans, 'Education is a Complex Dynamical System: Challenges for Research,' *The Journal of Experimental Education*, vol. 88, no. 3, pp. 358–374, May 2020, ISSN: 0022-0973. DOI: 10.1080/00220973.2019.1566199 (cit. on p. 13).

[60] P. Kyriakou, I. Hatzilygeroudis and J. Garofalakis, 'A tool for managing domain knowledge and helping tutors in intelligent tutoring systems,' *Journal of Universal Computer Science*, vol. 16, no. 19, 2010, ISSN: 0958695X. [Online]. Available: https://www.researchgate.net/publication/220738589_A_Tool_for_Managing_Domain_Knowledge_in_Intelligent_Tutoring_Systems (cit. on p. 12).

[61] C.-Y. Lee and T. Sloan Chener, 'A Comprehensive Evaluation Rubric for Assessing Instructional Apps,' *Journal of Information Technology Education: Research*, vol. 14, pp. 021–053, 2015, ISSN: 1547-9714. DOI: 10.28945/2097. [Online]. Available: https://www.informingscience.org/Publications/2097 (cit. on p. 77).

[62] S. Lgarch, M. Hnida and A. Retbi, 'Empowering E-learning through blockchain: an inclusive and affordable tutoring solution,' *International Journal of Electrical and Computer Engineering*, vol. 14, no. 5, pp. 5554–5565, Oct. 2024, ISSN: 20888708. DOI: 10.11591/ijece.v14i5.pp5554-5565 (cit. on pp. 18, 21).

[63] T. Li, L. M. C. Castro, K. Douglas and C. G. Brinton, 'Relationship between learning engagement metrics and learning outcomes in online engineering course,' in *2021 IEEE Frontiers in Education Conference (FIE)*, IEEE, Oct. 2021, pp. 1–5, ISBN: 978-1-6654-3851-3. DOI: 10.1109/FIE49875.2021.9637234 (cit. on p. 84).

[64] J. Liang *et al.*, 'Student Modeling and Analysis in Adaptive Instructional Systems,' *IEEE Access*, vol. 10, pp. 59 359–59 372, 2022, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3178744. [Online]. Available: https://ieeexplore.ieee.org/document/9784853/ (cit. on p. 10).

[65] S. F. Lima and S. M. D. Stump, 'How to obtain knowledge evidences in a student model based on Bayesian Network,' in *Proceedings of the 5th Iberian Conference on Information Systems and Technologies, CISTI 2010*, 2010 (cit. on pp. 62, 75).

[66] A. Limaye, S. S. Karkera, H. Khatri and V. T. Raisinghani, 'PREP: Prerequisite Relationship Extraction Using Position-Biased Burst Analysis,' in *Lecture Notes in Electrical Engineering*, vol. 888, Springer, 2022, pp. 217–228. DOI: 10.1007/978-981-19-1520-8_17. [Online]. Available: https://link.springer.com/10.1007/978-981-19-1520-8_17 (cit. on p. 70).

[67] J. Matayoshi, U. Granziol, C. Doble, H. Uzun and E. Cosyn, 'Forgetting curves and testing effect in an adaptive learning and assessment system,' in *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018*, 2018 (cit. on p. 17).

[68] J. Matayoshi and H. Uzun, 'Learning, forgetting, and the correlation of knowledge in knowledge space theory,' *Journal of Mathematical Psychology*, vol. 109, 2022, ISSN: 10960880. DOI: 10.1016/j.jmp.2022.102674 (cit. on p. 17).

[69] J. Matayoshi, H. Uzun and E. Cosyn, 'Deep (Un)Learning: Using Neural Networks to Model Retention and Forgetting in an Adaptive Learning System,' in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11625 LNAI, 2019, pp. 258–269. DOI: 10.1007/978-3-030-23204-7_22. [Online]. Available: https://link.springer.com/10.1007/978-3-030-23204-7_22 (cit. on p. 17).

[70] J. Máth and K. Abari, 'KNOWLEDGE SPACES AND HISTORICAL KNOWLEDGE IN PRACTICE,' Tech. Rep., 2011, pp. 126–152 (cit. on p. 42).

[71] K. Mishra and R. B. Mishra, 'Multiagent Based Selection of Tutor-Subject-Student Paradigm in an Intelligent Tutoring System,' in *Insights into Advancements in Intelligent Information Technologies*, IGI Global, 2012, pp. 47–71. DOI: 10.4018/978-1-4666-0158-1.ch004. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-0158-1.ch004 (cit. on p. 10).

[72] H. Munir, B. Vogel and A. Jacobsson, 'Artificial Intelligence and Machine Learning Approaches in Digital Education: A Systematic Revision,' *Information*, vol. 13, no. 4, p. 203, Apr. 2022, ISSN: 2078-2489. DOI: 10.3390/info13040203. [Online]. Available: https://www.mdpi.com/2078-2489/13/4/203 (cit. on p. 83).

[73] J. Muñoz-Mederos, E. Acosta-Gonzaga, E. F. Ruiz-Ledesma and A. Ramírez-Arellano, 'Do Learning Styles Enhance the Academic Performance of University Students? A Case Study,' *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, ISSN: 21565570. DOI: 10.14569/IJACSA.2021.0120686. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=12&Issue=6&Code=IJACSA&SerialNo=86 (cit. on p. 81).

[74] P. Nedungadi and M. S. Remya, 'Incorporating forgetting in the Personalized, Clustered, Bayesian Knowledge Tracing (PC-BKT) model,' in *2015 International Conference on Cognitive Computing and Information Processing(CCIP)*, IEEE, Mar. 2015, pp. 1–5, ISBN: 978-1-4799-7171-8. DOI: 10.1109/CCIP.2015.7100688. [Online]. Available: http://ieeexplore.ieee.org/document/7100688/ (cit. on p. 17).

[75] R. Nkambou, E. Mephu Nguifo, O. Couturier and P. Fournier-Viger, 'Problem-Solving Knowledge Mining from Users' Actions in an Intelligent Tutoring System,' in 2007, pp. 393–404. DOI: 10.1007/978-3-540-72665-4_34 (cit. on p. 13).

[76] E. Nyitrai and B. Varga, 'Ontology-based higher educational information systems,' *Pollack Periodica*, vol. 7, no. 3, 2012, ISSN: 17881994. DOI: 10.1556/Pollack.7.2012.3.14 (cit. on p. 50).

[77] E. Ovchinnikova, L. Vieu, A. Oltramari, S. Borgo and T. Alexandrov, 'Data-driven and ontological analysis of framenet for natural language reasoning,' in *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*, 2010 (cit. on pp. 34, 37, 51).

[78] A. L. Overono and A. S. Ditta, *The Rise of Artificial Intelligence: A Clarion Call for Higher Education to Redefine Learning and Reimagine Assessment*, 2023. DOI: 10.1080/87567555.2023.2233653 (cit. on p. 18).

[79] G. Palaigeorgiou and A. Papadopoulou, 'Promoting self-paced learning in the elementary classroom with interactive video, an online course platform and tablets,' *Education and Information Technologies*, vol. 24, no. 1, pp. 805–823, Jan. 2019, ISSN: 1360-2357. DOI: 10.1007/s10639-018-9804-5. [Online]. Available: http://link.springer.com/10.1007/s10639-018-9804-5 (cit. on p. 69).

[80] V. Parvathy and D. Mishra, 'Challenges of Multimedia Education in the Lives of Human Beings,' in 2023, pp. 149–159. DOI: 10.1007/978-981-99-6568-7_14 (cit. on p. 3).

[81] M. Pawar, V. Dhotare, N. Urunkar, Y. Andalgavkarkulkarni, D. Shahane and A. S. Pawar, 'Comparative Impact of AI and Search Technologies on Outcome-Based Learning in Engineering Education,' *Journal of Engineering Education Transformations*, vol. 38, no. IS2, pp. 591–598, May 2025, ISSN: 23492473. DOI: 10.16920/jeet/2025/v38is2/25073 (cit. on p. 8).

[82] B. Pe˛kala, 'Properties of Atanassov's intuitionistic fuzzy relations and Atanassov's operators,' *Information Sciences*, vol. 213, pp. 84–93, Dec. 2012, ISSN: 00200255. DOI: 10.1016/j.ins.2012.05.024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025512003830 (cit. on p. 17).

[83] R. Pelánek, 'Managing items and knowledge components: domain modeling in practice,' *Educational Technology Research and Development*, vol. 68, no. 1, pp. 529–550, Feb. 2020, ISSN: 1042-1629. DOI: 10.1007/s11423-019-09716-w. [Online]. Available: http://link.springer.com/10.1007/s11423-019-09716-w (cit. on pp. 10, 58, 75).

[84] R. F. Person, 'Education and the Transmission of Tradition,' in *The Wiley Blackwell Companion to Ancient Israel*, Wiley, Oct. 2015, pp. 366–378. DOI: 10.1002/9781118774199.ch20 (cit. on p. 1).

[85] A. Phillips, J. F. Pane, R. Reumann-Moore and O. Shenbanjo, 'Implementing an adaptive intelligent tutoring system as an instructional supplement,' *Educational Technology Research and Development*, vol. 68, no. 3, pp. 1409–1437, Jun. 2020, ISSN: 1042-1629. DOI: 10.1007/s11423-020-09745-w. [Online]. Available: http://link.springer.com/10.1007/s11423-020-09745-w (cit. on p. 58).

[86] C. Pierrakeas, G. Solomou and A. Kameas, 'An ontology-based approach in learning programming languages,' in *Proceedings of the 2012 16th Panhellenic Conference on Informatics, PCI 2012*, 2012. DOI: 10.1109/PCi.2012.78 (cit. on p. 14).

[87] R. Prendergast, 'On the Social Evolution of Knowledge,' in *Governing Markets as Knowledge Commons*, Cambridge University Press, Dec. 2021, pp. 58–88. DOI: 10.1017/9781108692915.004. [Online]. Available: https://www.cambridge.org/core/product/identifier/9781108692915%23CN-bp-2/type/book_part (cit. on p. 2).

[88] X. Qiao, X. Zheng, X. Sun, S. Li and Y. Zhang, 'Learners' States Monitoring Method Based on Face Recognition Technology,' *Procedia Computer Science*, vol. 202, pp. 172–177, 2022, ISSN: 18770509. DOI: 10.1016/j.procs.2022.04.024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050922005580 (cit. on pp. 91, 92).

[89] M. Raeiszadeh, F. Estrada-Solano, R. H. Glitho, J. Eker and R. A. F. Mini, 'A Data-Driven Approach for Adaptive Real-Time Log Parsing in Cloud Environments,' in *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, IEEE, Jul. 2024, pp. 173–178, ISBN: 979-8-3503-0948-5. DOI: 10.1109/MeditCom61057.2024.10621416. [Online]. Available: https://ieeexplore.ieee.org/document/10621416/ (cit. on p. 75).

[90] N. Reimers and I. Gurevych, 'Sentence-BERT: Sentence embeddings using siamese BERT-networks,' in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019. DOI: 10.18653/v1/d19-1410 (cit. on p. 29).

[91] Y. R. Rincón, A. Munárriz, M. J. Campión Arrastia and M. I. Goicoechea López-Vailo, 'Instructional design for tutoring on interactive platforms: creating educational interventions overcoming the digital gap,' *Educational technology research and development*, May 2025, ISSN: 1042-1629. DOI: 10.1007/s11423-025-10516-8. [Online]. Available: https://link.springer.com/10.1007/s11423-025-10516-8 (cit. on p. 76).

[92] P. Robinson, 'Designing and Conducting Mixed Methods Research,' *Australian and New Zealand Journal of Public Health*, vol. 31, no. 4, 2007, ISSN: 13260200. DOI: 10.1111/j.1753-6405.2007.00096.x (cit. on p. 74).

[93] M. Rosic, V. Glavinic and S. Stankov, 'Intelligent Tutoring Systems for the New Learning Infrastructure,' in *Intelligent Learning Infrastructure for Knowledge Intensive Organizations*, IGI Global, 2005, pp. 225–250, ISBN: 9781591405030. DOI: 10.4018/978-1-59140-503-0.ch008. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59140-503-0.ch008 (cit. on p. 9).

[94] G. V. Rybina and A. A. Grigoriev, 'Modern Architectures of Intelligent Tutoring Systems Based on Integrated Expert Systems: Features of the Approach to the Automated Formation of the Ontological Space of Knowledge and Skills of Students,' *Pattern Recognition and Image Analysis*, vol. 33, no. 3, pp. 491–497, Sep. 2023, ISSN: 1054-6618. DOI: 10.1134/S1054661823030409 (cit. on p. 9).

[95] S. Sani and T. N. Aris, 'Computational Intelligence Approaches for Student/-Tutor Modelling: A Review,' in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, IEEE, Jan. 2014, pp. 72–76, ISBN: 978-1-4799-3858-2. DOI: 10.1109/ISMS.2014.21 (cit. on pp. 10, 12).

[96] L. Sha and P. Hong, 'Neural Knowledge Tracing,' in 2017, pp. 108–117. DOI: 10.1007/978-3-319-67615-9_10 (cit. on p. 16).

[97] H. c. J. Shih and S. h. C. Huang, 'College students' metacognitive strategy use in an EFL flipped classroom,' *Computer Assisted Language Learning*, vol. 33, no. 7, 2020, ISSN: 17443210. DOI: 10.1080/09588221.2019.1590420 (cit. on p. 92).

[98] M. N. Siddiqui, A. Gupta, J. M. Reddig and C. J. MacLellan, 'HTN-Based Tutors: A New Intelligent Tutoring Framework Based on Hierarchical Task Networks,' in *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, New York, NY, USA: ACM, Jul. 2024, pp. 491–495, ISBN: 9798400706332. DOI: 10.1145/3657604.3664702 (cit. on p. 51).

[99] J. Song, Y. Wang, C. Zhang and K. Xie, 'Self-attention and forgetting fusion knowledge tracking algorithm,' *Information Sciences*, vol. 680, p. 121 149, Oct. 2024, ISSN: 00200255. DOI: 10.1016/j.ins.2024.121149. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025524010636 (cit. on p. 91).

[100] R. Sottilare and K. Brawner, 'Component interaction within the generalized intelligent framework for tutoring (GIFT) as a model for adaptive instructional system standards,' in *CEUR Workshop Proceedings*, vol. 2121, 2018 (cit. on p. 10).

[101] R. A. Sottilare, R. S. Baker, A. C. Graesser and J. C. Lester, *Special Issue on the Generalized Intelligent Framework for Tutoring (GIFT): Creating a Stable and Flexible Platform for Innovations in AIED Research*, 2018. DOI: 10.1007/s40593-017-0149-9 (cit. on pp. 10, 51).

[102] N. L. Staus, K. O'Connell and M. Storksdieck, 'Addressing the Ceiling Effect when Assessing STEM Out-Of-School Time Experiences,' *Frontiers in Education*, vol. 6, Jul. 2021, ISSN: 2504-284X. DOI: 10.3389/feduc.2021.690431 (cit. on p. 84).

[103] C. M. Steiner, A. Nussbaumer and D. Albert, 'Supporting self-regulated personalised learning through competence-based knowledge space theory,' *Policy*

Futures in Education, vol. 7, no. 6, 2009, ISSN: 14782103. DOI: 10.2304/pfie.2009.7.6.645 (cit. on pp. 16, 70, 81).

[104] S. Su, P. Zeng, C. Kang and T. Xin, 'Integrate Question Information With Learning Behavior for Knowledge Tracing,' *IEEE Access*, vol. 13, pp. 33 532–33 543, 2025, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2025.3543536. [Online]. Available: https://ieeexplore.ieee.org/document/10892115/ (cit. on p. 91).

[105] J. Tan, 'An Empirical Study of Adaptive Learning Algorithm Based on Intrinsic Motivation in English Online Teaching and Learning,' *Journal of Electrical Systems*, vol. 20, no. 6s, pp. 1695–1704, Apr. 2024, ISSN: 1112-5209. DOI: 10.52783/jes.3088. [Online]. Available: http://journal.esrgroups.org/jes/article/view/3088 (cit. on p. 81).

[106] S. D. Teasley and S. Lonn, 'Using learning management systems to support students' collaborative learning in higher education,' in *Proceedings of the 8th iternational conference on Computer supported collaborative learning - CSCL'07*, Morristown, NJ, USA: Association for Computational Linguistics, 2007, pp. 718–720, ISBN: 9780615154367. DOI: 10.3115/1599600.1599731 (cit. on p. 9).

[107] V. Telnov and Y. Korovin, 'Semantic web and knowledge graphs as an educational technology of personnel training for nuclear power engineering,' *Nuclear Energy and Technology*, vol. 5, no. 3, pp. 273–280, Sep. 2019, ISSN: 2452-3038. DOI: 10.3897/nucet.5.39226. [Online]. Available: https://nucet.pensoft.net/article/39226/ (cit. on p. 69).

[108] T. W. Teo and Z. Q. Peh, 'An exploratory study on eye-gaze patterns of experts and novices of science inference graph items,' *STEM Education*, vol. 3, no. 3, 2023, ISSN: 27671925. DOI: 10.3934/steme.2023013 (cit. on p. 81).

[109] B. Thalheim, 'Open Problems of Information Systems Research and Technology,' in 2013, pp. 10–18. DOI: 10.1007/978-3-642-40823-6_2. [Online]. Available: http://link.springer.com/10.1007/978-3-642-40823-6_2 (cit. on p. 8).

[110] A. A. Toptsis, 'Toward an Emotion Machine-based hybrid adaptive tutoring system,' in *The 3rd International Conference on Information Sciences and Interaction Sciences*, IEEE, Jun. 2010, pp. 133–137, ISBN: 978-1-4244-7384-7. DOI: 10.1109/ICICIS.2010.5534729 (cit. on p. 9).

[111] J. Traxler, 'Distance learning—Predictions and possibilities,' *Education Sciences*, vol. 8, no. 1, 2018, ISSN: 22277102. DOI: 10.3390/educsci8010035 (cit. on p. 8).

[112] T. Turja, J. Minkkinen and S. Mauno, 'Robotizing meaningful work,' *Journal of Information, Communication and Ethics in Society*, vol. ahead-of-print, Dec. 2021. DOI: 10.1108/JICES-06-2021-0063 (cit. on pp. 4, 18).

[113] B. Uttl, 'Measurement of Individual Differences,' *Psychological Science*, vol. 16, no. 6, pp. 460–467, Jun. 2005, ISSN: 0956-7976. DOI: 10.1111/j.0956-7976.2005.01557.x (cit. on p. 84).

[114] L. Vettraino, E. Guglielman, M. Guspini and V. Castello, 'Complex Learning: A Way for Rethinking Pedagogies and Processes in Technology-Enhanced Learning and Education,' in *2012 IEEE 12th International Conference on Advanced Learning Technologies*, IEEE, Jul. 2012, pp. 143–145, ISBN: 978-1-4673-1642-2. DOI: 10.1109/ICALT.2012.95 (cit. on p. 9).

[115] S. Voswinkel, 'Meaningful Work,' *Comparative Sociology*, vol. 19, pp. 741–755, Dec. 2020. DOI: 10.1163/15691330-12341529 (cit. on pp. 4, 18).

[116] M. Wang, C. Peng, R. Yang, C. Wang, Y. Chen and X. Yu, 'GASKT: A Graph-Based Attentive Knowledge-Search Model for Knowledge Tracing,' in 2021, pp. 268–279. DOI: 10.1007/978-3-030-82136-4_22. [Online]. Available: https://link.springer.com/10.1007/978-3-030-82136-4_22 (cit. on p. 91).

[117] Y. Wang, Y. Yuan, Y. Ma and G. Wang, *Time-Dependent Graphs: Definitions, Applications, and Algorithms*, Dec. 2019. DOI: 10.1007/s41019-019-00105-0 (cit. on pp. 15, 34, 51).

[118] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster and J. R. Movellan, 'The Faces of Engagement: Automatic Recognition of Student Engagementfrom Facial Expressions,' *IEEE Transactions on Affective Computing*, vol. 5, no. 1, pp. 86–98, Jan. 2014, ISSN: 1949-3045. DOI: 10.1109/TAFFC.2014.2316163. [Online]. Available: http://ieeexplore.ieee.org/document/6786307/ (cit. on p. 11).

[119] M. K. Worden and M. J. Bray, 'A theory of change approach to curricular revision: filling a curricular gap identified through curriculum mapping,' *BMC Medical Education*, vol. 25, no. 1, p. 533, Apr. 2025, ISSN: 1472-6920. DOI: 10.1186/s12909-025-07100-2 (cit. on p. 14).

[120] Y. Zhang, 'Analyzing the Impact of Learning Styles and Preferences on Vocational English Education Using Multiple Linear Regression Model,' in *2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII)*, IEEE, Aug. 2023, pp. 221–223, ISBN: 979-8-3503-2353-5. DOI: 10.1109/ICKII58656.2023.10332554. [Online]. Available: https://ieeexplore.ieee.org/document/10332554/ (cit. on p. 81).

[121] D. Zhao, 'The impact of AI-enhanced natural language processing tools on writing proficiency: an analysis of language precision, content summarization, and creative writing facilitation,' *Education and Information Technologies*, vol. 30, no. 6, pp. 8055–8086, Apr. 2025, ISSN: 1360-2357. DOI: 10.1007/s10639-024-13145-5. [Online]. Available: https://link.springer.com/10.1007/s10639-024-13145-5 (cit. on p. 81).

[122] B. J. Zimmerman, 'Becoming a Self-Regulated Learner: An Overview,' *Theory Into Practice*, vol. 41, no. 2, pp. 64–70, May 2002, ISSN: 0040-5841. DOI: 10.1207/s15430421tip4102_2. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1207/s15430421tip4102_2 (cit. on pp. 16, 69).

[123] A. Zouaq, R. Nkambou and C. Frasson, 'Document semantic annotation for intelligent tutoring systems: A concept mapping approach,' in *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2007*, 2007. [Online]. Available: https://cdn.aaai.org/FLAIRS/2007/FLAIRS07-076.pdf (cit. on pp. 12, 70).

Appendix
Graph4Learn System Screenshots

The following screenshots illustrate key components and functionalities of the Graph4Learn intelligent tutoring system developed as part of this research. The figures demonstrate the user interface, graph-based knowledge representation, and adaptive learning path features discussed in Chapter 4 and in Chapter 5.

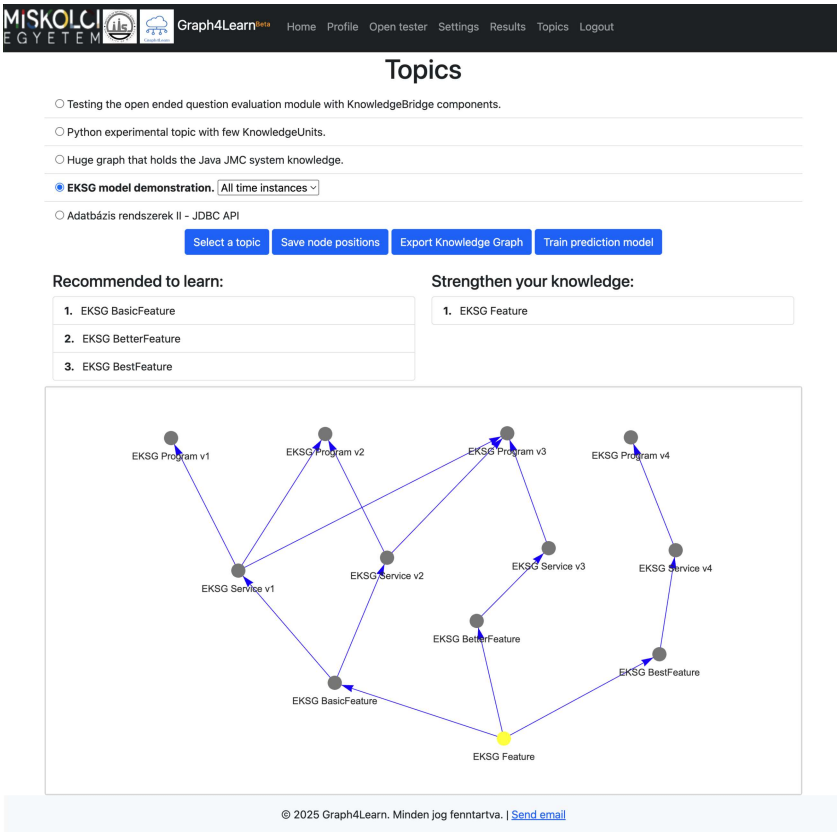
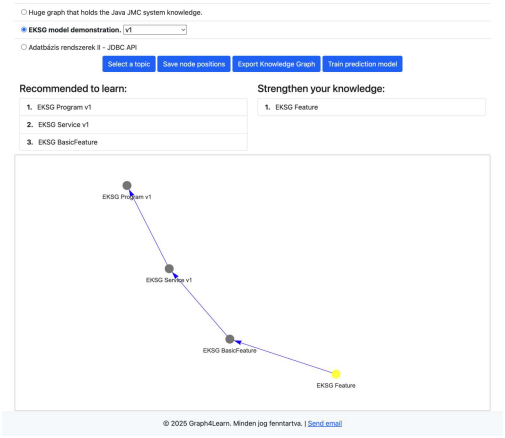
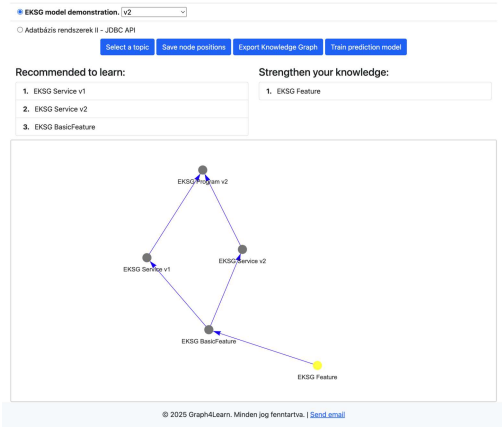


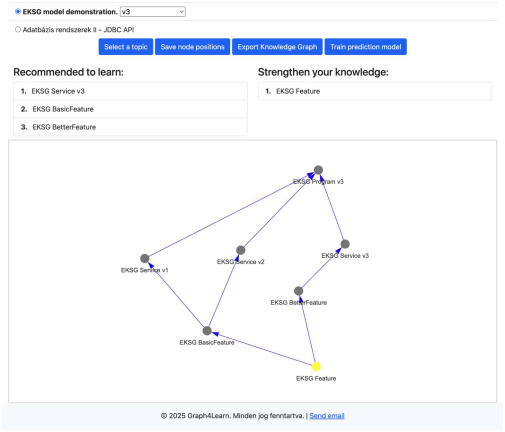
Figure 1: The *Topics* view of the Graph4Learn teacher interface. This screen allows the tutor to browse and select from available learning materials. Through the drop-down menu, different abstract time intervals of the evolving knowledge space can be selected; in this case, the entire time range is displayed. The interface also shows administrative controls, a table listing the next recommended learning steps, the knowledge units marked for revision, and the EKSG graph visualization depicting prerequisite relations among topics.



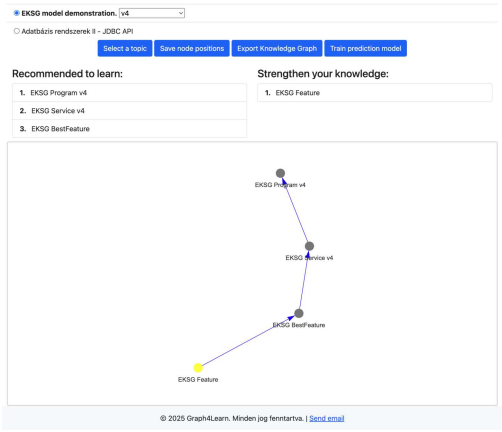
(a) Abstract time instance 1



(b) Abstract time instance 2

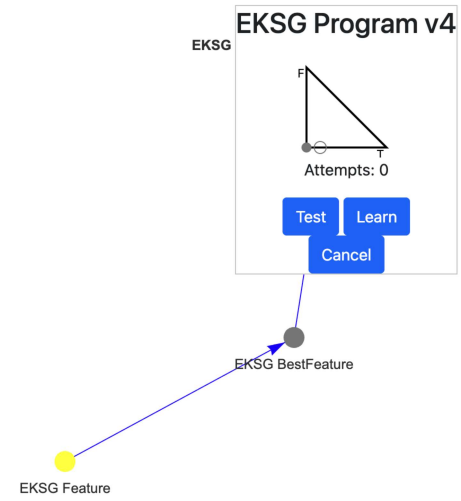


(c) Abstract time instance 3

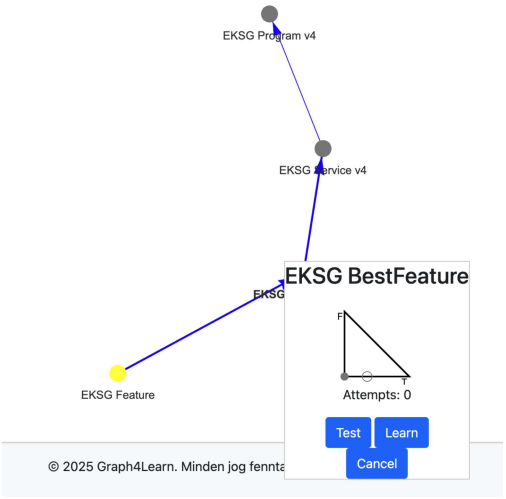


(d) Abstract time instance 4

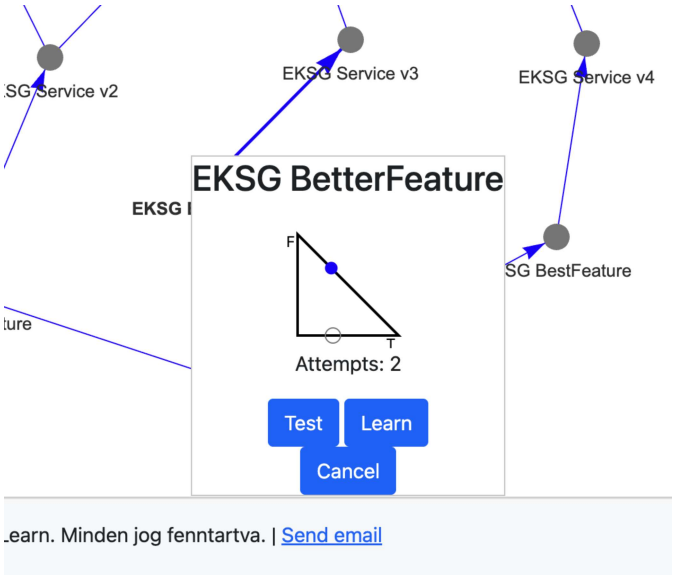
Figure 2: The four abstract time instances of the demo curriculum, representing the evolving states of the EKSG model over time. Each snapshot illustrates a different temporal stage of the knowledge graph, demonstrating how prerequisite relations and knowledge units change dynamically.



(a) Knowledge unit pop-up displaying measured and predicted knowledge levels using the enhanced IFL triangle.

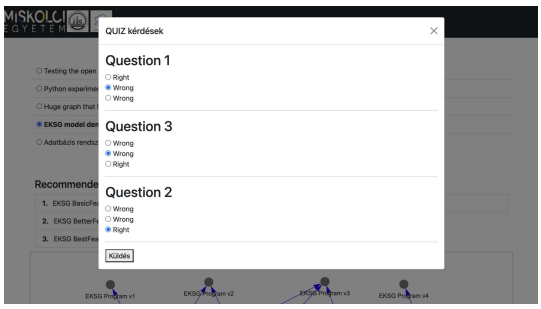


(b) Knowledge unit pop-up displaying measured and predicted knowledge levels using the enhanced IFL triangle.

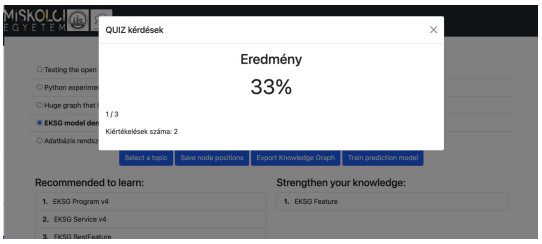


(c) An example showing the learner's number of quiz attempts and corresponding IFL triangle.

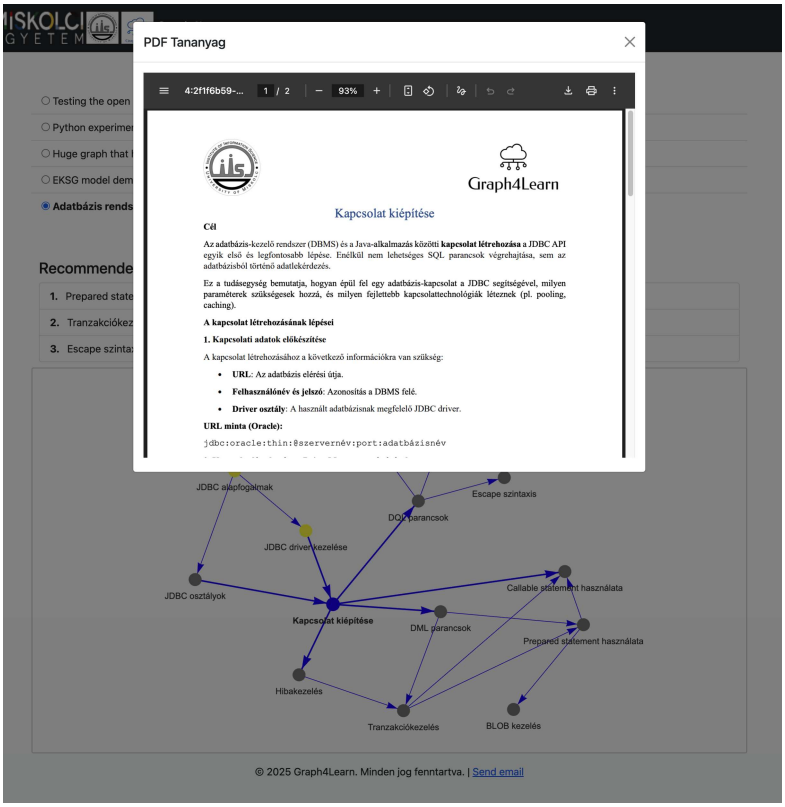
Figure 3: Interactive visualization of knowledge unit pop-up windows in the Graph4Learn system. Each pop-up integrates the enhanced IFL triangle, which visualizes both measured and predicted knowledge states for the learner. A solid circle indicates the measured, while an empty circle represents the predicted knowledge state. The interface also provides direct access to learning materials, quiz initiation, and displays the learner's number of prior attempts, supporting self-regulated learning and progress tracking.



(a) Starting a quiz for a selected knowledge unit. The learner can test their understanding through quiz questions.



(b) Quiz result interface showing the learner's achieved score.



(c) Example of a learning material (PDF) associated with a knowledge unit, accessible directly from the interface.

Figure 4: Quiz initiation, result feedback, and related learning material view in the Graph4Learn system.

Figure 5: Teacher's dashboard showing the progress of individual learners. Each row represents a student, displaying the corresponding IFL triangle that visualizes the measured knowledge states. This overview enables educators to monitor learning progression and identify where additional support may be needed.