ISSN 2616-7433 Vol. 8, Issue 3: 32-38, DOI: 10.25236/FSST.2026.080305

Design and Implementation of a Personalized Learning System Integrating Knowledge Graph and Lightweight Agent

Jiacheng Jin[1,a], Min Wang[1,b,*], Yuxuan Wang[1,c], Zitong Hui[1,d], Meng Li[1,e], Nian Liu[1,f]

[1]Information School, Beijing City University, Beijing, China
[a]2022636349@qq.com, [b]wangmin@bcu.edu.cn, [c]wangyuxuan20050823@163.com, [d]18545830529@163.com, [e]18311360348@163.com, [f]2115703480@qq.com
**Corresponding author*

Abstract: *This paper focuses on the cutting-edge advancement of AI-empowered education, designing and implementing a personalized learning system empowered by knowledge graph and educational agent. For the Python course, semantic compression technology is used to distill student dialogues into cognitive labels in real time, building dynamic learning profiles and long-term companionship memory. Combined with intelligent agents and knowledge graph queries, heuristic tutoring is achieved, effectively suppressing AI hallucinations. The homework grading module features targeted traceability, automatically locating prerequisite knowledge nodes to form a closed loop of assessment and remediation. The system boasts both high-concurrency responsiveness and precise tutoring capabilities under low-computing-power conditions, significantly reducing teachers' workload and improving personalized learning outcomes.*

Keywords: *Knowledge Graph; Educational Agent; Personalized Learning; Traceability Collaboration*

1. Introduction

Against the backdrop of accelerating digital transformation, educational digitization has been established as a strategic pillar for achieving the goal of educational modernization, becoming a core driving force for systemic educational reform. In recent years, the deep integration of cutting-edge technologies such as knowledge graphs, cognitive diagnosis theory, and large language models has provided a new technical paradigm for personalized learning path planning, precise question answering, and dynamic learning assessment.

This paper focuses on the core area of AI-empowered education, aiming to reveal the limitations of current technologies in practical engineering applications and system architecture design through classification and critical analysis of existing frontier research, and to provide an innovative architectural design solution for building an efficient, lightweight, closed-loop personalized learning engine.

2. Project research background

Although the integration of artificial intelligence and online education has brought tremendous development opportunities, practical teaching applications and system engineering implementation still face many challenges and technical pain points. The rapid expansion of online educational resources has led to severe problems such as "information overload," "cognitive disorientation," and "resource disorientation" when learners face massive, fragmented information, making it difficult to accurately select learning content that meets their own needs. Furthermore, traditional teacher-centered or standardized classroom teaching methods often overlook individual differences in learners' knowledge reserves, learning abilities, cognitive levels, and learning styles, making it difficult to provide refined personalized guidance, thus limiting learning efficiency.

ISSN 2616-7433 Vol. 8, Issue 3: 32-38, DOI: 10.25236/FSST.2026.080305

2.1. Educational Knowledge Graph Construction and Companion Memory Application

The construction of educational knowledge graphs has gradually shifted from manual to automated approaches. Zhang Jiaqi [1] adopted LoRA fine-tuning of large language models to efficiently extract triples and relationships from textbooks. An Mengliang [2] proposed the TMN-BiLSTM-CRF model, which uses entity triggers to alleviate the problem of named entity recognition (NER) under low-resource conditions. Wen Chao et al. [3] took the "Grassland Soil Biology" course as an example to build a course knowledge graph and validated the feasibility of a question-answering system. In the area of personalized recommendation, Zhang Xuefei et al. [4] systematically reviewed the collaborative application framework of knowledge graphs and large language models. Tu Yixiao [5] proposed a collaboratively driven learning recommendation framework that supports dynamic learner profiling and personalized path generation. Zhai Jie et al. [6] constructed a dual-agent collaborative recommendation model, LightPLRec. However, most existing knowledge graphs remain "static". How to transform students' daily conversations into structured growth portfolios in real time remains a major challenge [7].

2.2. Current Status of Intelligent Q&A and Academic Tutoring Applications

To mitigate the "hallucination" problem of large language models in education, Zhang Wei et al. [8] proposed a three-stage retrieval strategy combining precise keyword matching, vector search, and relation search, leveraging knowledge graphs to improve answer accuracy [4]. Cao Wei et al. [9] developed teaching agents for academic tutoring through structured prompt engineering and workflow design. Nevertheless, complex retrieval-augmented generation architectures still rely on cloud-based high-performance computing, leading to high response latency. The work of Zhai Jie et al. [6] and Fang Haiguang et al. [7] shows that lightweight large language models, when combined with agents, can achieve efficient inference and personalized tutoring under low-computing-power conditions, providing theoretical support for lightweight deployment.

2.3. Current Status of Teaching Assessment and Homework Grading Applications

Agent technology is driving the automation of educational assessment. Duan Dongsheng et al. [10] designed an assessment system that combines dynamic knowledge graphs with agents: agents decompose assessment objectives and build a gradient-integrated evaluation database, thereby improving evaluation efficiency under high-concurrency scenarios. Fang Haiguang et al. [7] emphasized that a meta-diagnosis mechanism can provide holistic diagnostics and targeted feedback to students. An Mengliang [2] used knowledge tracing together with predecessor/successor relations in a knowledge graph to achieve intelligent guidance in the learning process. Zhang Xuefei et al. [4] pointed out that the collaboration between knowledge graphs and large language models can enhance the interpretability of reasoning. However, most current homework grading systems remain at the superficial level of "right/wrong judgment" or simple grammar correction, lacking a bidirectional "trace-and-collaborate" capability with the underlying knowledge graph. Consequently, they are unable to explore the root causes of student errors and achieve a closed loop from accurate assessment to targeted remediation-an area that requires further research.

2.4. Development Prospects

Currently, China is accelerating the digital transformation of education, and AI-empowered education has become an important part of the national strategy. The "Outline of the Plan for Building a Leading Country in Education (2024–2035)" [11] issued by the Central Committee of the Communist Party of China and the State Council clearly proposes the strategic deployment of "promoting AI-assisted educational change," emphasizing the need to "create large models for AI in education," "promote the construction of smart campuses, and explore effective ways to empower large-scale personalized teaching and innovative teaching through digitalization." Under this macro-policy background, personalized learning systems integrating knowledge graphs and educational agents have encountered unprecedented development opportunities.

ISSN 2616-7433 Vol. 8, Issue 3: 32-38, DOI: 10.25236/FSST.2026.080305

3. System Function Analysis

3.1 Requirement Analysis

The system addresses two major pain points of general large language models: susceptibility to hallucinations and reliance on high computational power. It introduces a master control agent and semantic compression technology, running on low computational power via the Ollama quantization engine. Core functions include knowledge graph construction, master control agent-based dynamic diagnosis, intelligent grading, and targeted traceability, forming a closed loop from assessment to remediation.

3.2 Core Functions

Focusing on learners' knowledge mastery status and personalized tutoring needs, the platform integrates four core modules: knowledge graph construction, master control agent dynamic diagnosis, intelligent grading, and targeted traceability. Together, they form a closed-loop process from knowledge organization to learning intervention. Based on self-reflection and dynamic confidence auditing, a high-quality domain knowledge graph is built to suppress hallucinations. Relying on the master control agent and semantic compression, the system extracts "cognitive labels" in real time, generates dynamic student profiles, and provides low-latency heuristic tutoring. During grading, it automatically traces prerequisite knowledge points, enabling targeted remediation of weak areas and achieving an integrated "assessment-evaluation-remediation" loop.

4. System Design Analysis

4.1 System Design Framework

The overall architecture is divided into four layers, as shown in Figure 1:

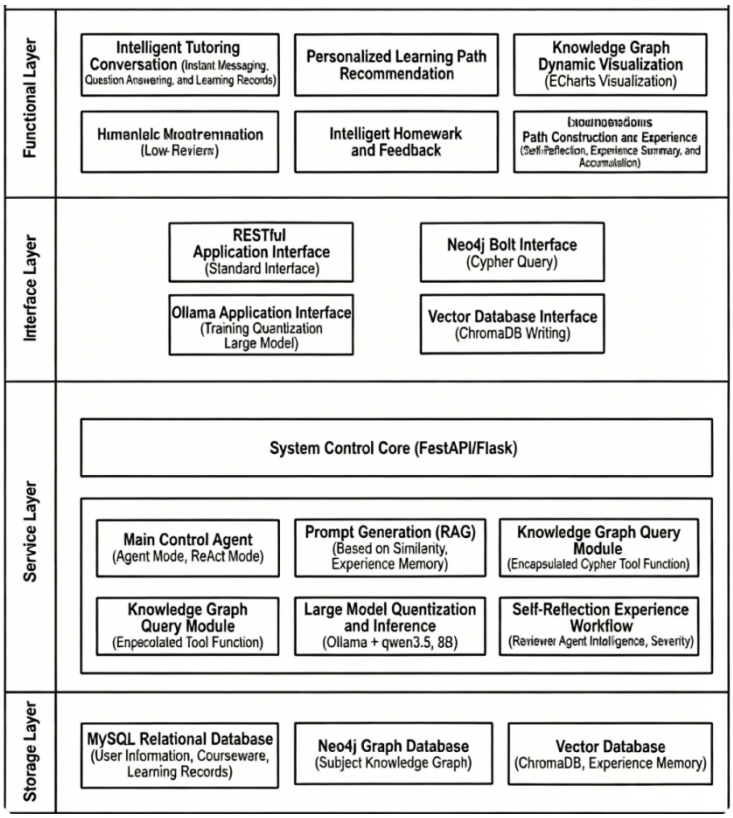

Figure 1: System Architecture Diagram

(1)Functional Layer: Provides intelligent tutoring dialogue, knowledge graph visualization, automated graph construction and validation, personalized learning path recommendation, homework

ISSN 2616-7433 Vol. 8, Issue 3: 32-38, DOI: 10.25236/FSST.2026.080305

grading with traceability, and human-AI collaborative validation.

(2)Interface Layer: Encapsulates RESTful APIs, Neo4j Bolt interface, Ollama API, and vector database interface to enable system communication and data access.

(3)Service Layer: Built on FastAPI/Flask, includes the master agent, retrieval-augmented generation (RAG), knowledge graph query engine, LLM quantization engine, and a self-reflective validation workflow.

(4)Storage Layer: Handles data persistence using MySQL for user and learning profile data, Neo4j for the domain knowledge graph, and ChromaDB for companion memory.

4.2 System Module Design and Interface Design

4.2.1 System Module Design

To realize the core functions of the "Smart Learning Engine" intelligent tutoring system, the overall system design is divided into seven main modules: User Management Module, Knowledge Graph Exploration Module, Intelligent Companion Module, Homework Assessment Module, Companion Memory Module, Teacher Workbench Module, and Administrator Backend Module. Each module collaborates through data interfaces and the AI service layer to ensure real-time flow of learning data and intelligent feedback, as shown in Figure 2.

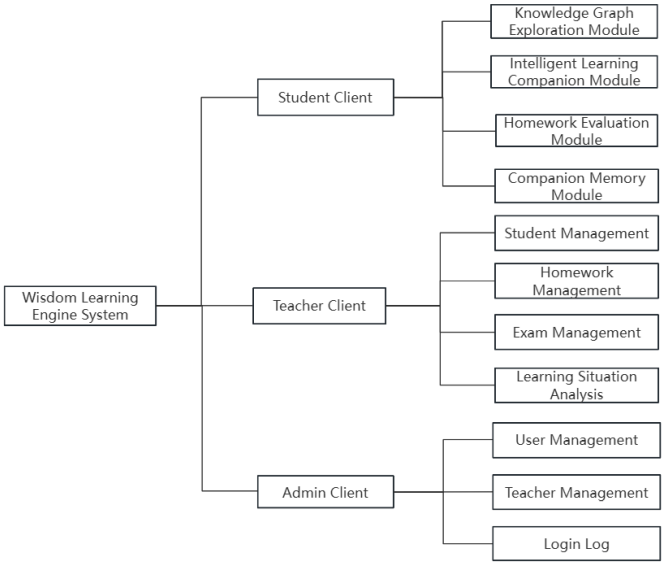

Figure 2: System module design and interface design

4.2.2 Knowledge Graph Exploration Module

Visually displays knowledge points and mastery status (mastered/learning/to consolidate). Supports search, filtering, and one-click focus on weak nodes, helping students precisely locate knowledge gaps and achieve targeted remediation.

4.2.3 Intelligent Companion Module

Integrates a local large language model that guides students through heuristic questioning without giving direct answers. It automatically detects cognitive boundaries and dynamically displays weak points. Students can converse anytime to receive personalized learning support and develop programming thinking.

4.2.4 Homework Assessment Module

Automatically grades homework and quizzes, identifies error types, tags associated knowledge points, and generates specific improvement suggestions. Grading results are synchronized to the Companion Memory Module, forming a closed learning loop.

4.2.5 Companion Memory Module

Records learning history along a timeline, manages weak points , automatically generates learning

ISSN 2616-7433 Vol. 8, Issue 3: 32-38, DOI: 10.25236/FSST.2026.080305

reports, and provides a “Start Review” button to support efficient revision and consolidation.

4.2.6 Student Management Module

This module manages student personal accounts and information, including student login and registration, personal information editing, and personal learning analysis. Students can view their academic data, modify usernames, and edit personal profiles in their personal center. The system automatically generates personal learning analysis reports based on learning records, supporting self-assessment of learning progress.

4.2.7 Homework Management Module

This module is used by students to view homework lists assigned by teachers, assignment details, and submit assignments. Students can submit assignments online, view deadlines, grading status, and scores, and resubmit or view historical assignment records.

4.2.8 Exam Management Module

This module supports students in taking exams online, including objective questions and programming questions. The system automatically times and grades objective questions, and programming questions support code execution testing. After the exam, students can view scores and answers, and weak points are automatically synchronized to the companion memory.

4.2.9 Learning Analytics Module

This module generates personal learning analysis reports based on student learning data, displaying knowledge point mastery radar charts, learning time trends, homework/exam score changes, etc., in graphical form. Students can intuitively understand their strengths and weaknesses and receive personalized learning suggestions. On the teacher side, teachers can also view the overall learning situation of class students, enabling them to adjust teaching progress accordingly.

4.2.10 User Management

This module is responsible for centralized management of all student accounts in the system, located in the administrator backend. Administrators can add, delete, modify, and query student accounts, and support batch import and export.

4.2.11 Teacher Management

This module is responsible for reviewing teacher accounts, maintaining information, and assigning permissions, located in the administrator backend. Administrators can view information of all registered teachers and perform management operations.

4.2.12 Login Logs

This module records the login behaviors of all users (students, teachers, administrators) in the system, providing security auditing and anomaly monitoring capabilities, located in the administrator backend.

5. System Testing

5.1 System Functional Testing

The system function test includes four items, as detailed in Table 1.

Table 1: System function test.

Test Item	Input	Test Result
User login and registration	Enter correct student ID/password to log in; register a new account	Pass
Intelligent companion dialogue	Ask a Python question; the AI tutor responds in a questioning manner	Pass
Identify keywords from dialogue to dynamically unlock knowledge points in the kg	Display mastered/unmastered knowledge points in different colors; click a graph node to view textbook source information about that knowledge point	Pass
Programming code comparison and result tracing	Upload Python code for comparison; after grading, display error type, associated knowledge points	Pass

ISSN 2616-7433 Vol. 8, Issue 3: 32-38, DOI: 10.25236/FSST.2026.080305

5.2 Core feature demonstration

5.2.1 Python knowledge graph visualization exploration interface

Graph-based visualization of Python knowledge points with color-coded mastery states. It supports search, filtering, and weak-node focusing to help students pinpoint knowledge gaps, as shown in Figure 3.

Figure 3: Educational Knowledge Graph

5.2.2 AI Python tutor intelligent companion dialogue interface

This interface integrates a large language model agent, guiding thinking through heuristic questioning; it identifies weak points via cognitive boundary detection and displays them dynamically, cultivating programming thinking skills, as shown in Figure 4.

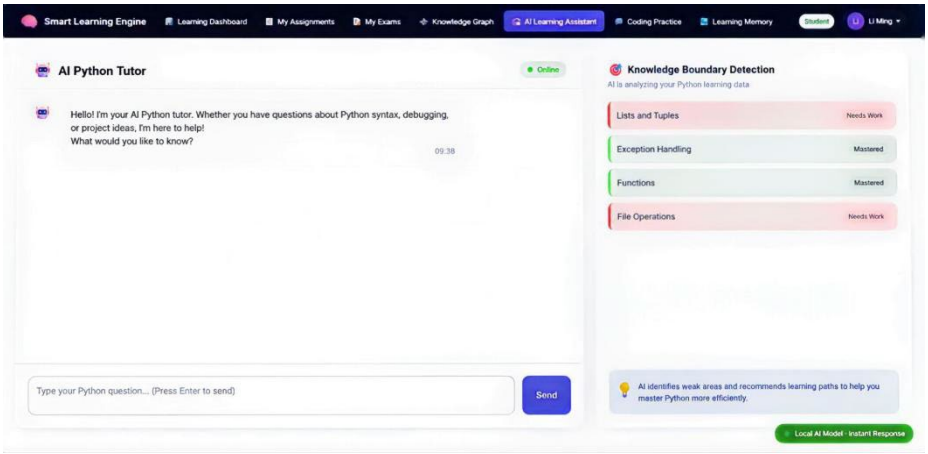

Figure 4: Educational Agent

6. Conclusions and prospects

6.1 Conclusion

The system integrates KG and lightweight agents, enabling dynamic diagnosis, heuristic tutoring, and traceable grading. Experiments show high concurrency under low computation, reduced hallucinations, improved learning outcomes, and reduced teacher workload.

6.2 Future Work

Future optimization and expansion will focus on the following aspects:

(1) Expand the multi-disciplinary knowledge graph, and study cross-disciplinary knowledge point

ISSN 2616-7433 Vol. 8, Issue 3: 32-38, DOI: 10.25236/FSST.2026.080305

associations and transfer learning path recommendations.

(2) Integrate multimodal interaction capabilities (e.g., voice questioning, code auto-completion, visual debugging assistance) to enhance learning immersion and interaction naturalness.

(3)Introduce deep knowledge tracing and reinforcement learning to achieve more refined dynamic planning of personalized learning paths that adapt to students' cognitive changes in real time.

Acknowledgment

This paper is supported by Beijing City University Student Innovation and Entrepreneurship Training Program Project.

References

[1] Zhang Jiaqi. Research and Implementation of Course Question Answering System Combining Large Language Model and Knowledge Graph [D]. Huazhong University of Science and Technology, 2024.

[2] AnMengliang. Design and Implementation of Intelligent Tutoring System Based on Knowledge Graph [D]. University of Chinese Academy of Sciences (Shenyang Institute of Computing Technology), 2021.

[3] Wen Chao, Zhang Dandan, Zhang Jing, et al. Exploration of Personalized Learning Model Based on Artificial Intelligence Knowledge Graph Construction: Taking "Grassland Soil Biology" Undergraduate Course as an Example [J/OL]. Pratacultural Science, 1-11 [2026-06-04]. https://link-cnki-net-s.webvpn.bcu.edu.cn:8000/urlid/62.1069.S.20251107.1033.004.

[4] Zhang Xuefei, Zhang Liping, Yan Sheng, et al. Personalized Learning Recommendation Based on Synergy of Knowledge Graph and Large Language Model [J]. Journal of Computer Applications, 2025, 45 (3): 773-784.

[5] Tu Yixiao. Personalized Learning Recommendation Driven by Synergy of Knowledge Graph and Large Language Model: Framework Construction, Application Verification, and Future Trends [J]. Information & Computer, 2025, 37 (17): 27-29.

[6] Zhai Jie, Li Yanhao, Chen Lexuan, et al. Dynamic Recommendation of Personalized Practical Learning Materials Based on Lightweight Educational Large Model [J]. Computer Science, 2026, 53 (2): 48-56.

[7] Fang Haiguang, Deng Yang, Li Zeyu, et al. Exploration of New Personalized Learning Based on AI Agents from the Perspective of Human-Machine Collaboration [J]. Modern Educational Technology, 2026, 36 (2): 40-50.

[8] Zhang Wei, Gan Lu, Du Mengqi, et al. Precise Q&A System for Course Knowledge Integrating Knowledge Graph and Large Language Model [J]. Modern Information Technology, 2026, 10 (8): 113-117.

[9] Cao Wei, Lu Shuping, Zhu Hongxu, et al. Research on the Development of Exclusive Agents Based on Personalized Teaching [J]. Vocational Technology, 2025, 24 (9): 37-43.

[10] Duan Dongsheng, Li Sihan. Design of Personalized Learning Evaluation System Combining AI-Driven Dynamic Graph and Agents [J]. Information System Engineering, 2025, (11): 157-160.

[11] The Central Committee of the Communist Party of China and the State Council. Outline of the Plan for Building a Leading Country in Education (2024–2035) [EB/OL]. (2025-01-19) [2025-04-28]. https://www.gov.cn/zhengce/202501/content_6999913.htm