From "Tool" to "Quasi-subject": Research on the "C-O-D-E-R" Paradigm in Middle School Information Technology Classrooms Under Multi-agent Synergy—Taking the Eighth-grade "Python Programming" Course Series as an Example

Ziyi Chen [a*], Suyu Lu [b], Libin Wang [c], Zhecheng Zhang [d], Zhiyuan Tang [e]

College of Education, Zhejiang University of Technology, Hangzhou, China

[*a]885200127@qq.com, [b]charles_acad@163.com
[c]3175531488@qq.com, [d]1829057897@qq.com, [e]1197019013@qq.com

Abstract. With the rapid advancement of artificial intelligence, the requirements for computational thinking and human-computer collaborative literacy are increasing. However, the traditional "teacher-student" binary structure struggles to facilitate personalized guidance, while existing digital tools often lack interactive subjectivity. Drawing on the theory of distributed cognition and Human-Machine Collaborative Regulation, this study constructs the C-O-D-E-R teaching paradigm. Incorporating five core phases—Contextual Activation, Organized Orchestration, Development Collaboration, Error Debugging, and Reflective Reconstruction—this research designs a multi-agent system as a classroom "quasi-subject. " It aims to reshape the "Teacher-Student-Machine" triadic collaborative ecosystem, providing a theoretical framework and practical pathway for the synchronous enhancement of computational thinking and human-computer collaborative literacy.

Keywords: C-O-D-E-R Paradigm, Multi-agent Synergy, Quasi-subject, Classroom Structure Reconstruction, Information Technology Education

1 Introduction

The release of 'Compulsory Education Information Technology Curriculum Standards (2022 Edition) 'has indicated a clear shift in the focus of teaching from mere coding techniques to more profound computational thinking. However, an analysis of contemporary junior high school classrooms reveals that, despite the gradual proliferation of artificial intelligence tools, their application remains confined to a passive, "tool-centric" paradigm, lacking the agency required for active perception and deep interaction,which leads students to fall into the cognitive trap of outsourced thinking and superficial learning (Ng D T K et al. , 2024). Guiding personalized learning within large-scale instruction remains challenging (Chiu T K F et al. , 2023). In light of this, the

© The Author(s) 2026
C. F. Peng et al. (eds.), *Proceedings of the 2026 5th International Conference on Humanities, Wisdom Education and Service Management (HWESM 2026)*, Advances in Social Science, Education and Humanities Research 1024,
https://doi.org/10.2991/978-2-38476-593-5_82

present study transcends traditional tool-based applications by reimagining multi-agent systems as "quasi-subjects" in the classroom, thereby establishing the C-O-D-E-R teaching paradigm—a framework that deeply integrates teachers, students, and machines. The approach delineates five key phases: Contextual Activation, Organised Orchestration, Development Collaboration, Error Debugging, and Reflective Reconstruction. These phases subvert the conventional boundaries between "teaching" and "learning" in traditional classrooms. By integrating agents into the cognitive process, the paradigm avoids the risks of "cognitive offloading" in traditional generative AI, safeguarding learner agency and exploring a new pathway from simple human-computer interaction to deep symbiosis.

2 Literature Review

The use of AI in education is changing from a tool to a collaborative agent. The construction of the C-O-D-E-R paradigm must be grounded in the technological evolution of multi-agents, break through the cognitive dilemmas inherent in instrumental rationality, and thereby establish the "quasi-subject" status of intelligent agents within the pedagogical domain.

2.1 From “Generative Assistance” to a “Multi-Agent Thinking Society”

The application of AI in education has undergone a paradigm shift from rule-based approaches to socialised collaboration. Early intelligent tutoring systems were effective in teaching well-structured knowledge but struggled with non-linear thought (Lajoie et al. , 1993). While generative AI has lowered the programming barrier (Kasneci et al. , 2023), single-model approaches often fall into the “omnipotence paradox”—attempting to address all instructional functions with a single general-purpose model, resulting in a lack of expertise in in-depth diagnosis (Gu Xiaoqing and Hao Xiangjun, 2025). In order to address this limitation, the academic community is shifting towards research on multi-agent systems grounded in the theory of social cognition (Wu Yonghe et al. , 2024). Ouyang et al. proposed a third paradigm of AI education, termed "man-machine symbiosis", which emphasised that AI should not merely be a passive executor of instructions, but rather be deeply embedded in the learning process as a partner. Recent empirical studies indicate that multi-agent environments, by simulating heterogeneous roles such as teachers and peers, construct mechanisms for knowledge co-construction and collaborative regulation. These environments can effectively support learners of varying proficiency levels and narrow performance gaps (Hao et al. , 2025), thereby providing a foundational framework for building virtual teaching teams (Yu et al. , 2024).

2.2 “Cognitive Offloading” and the Lack of Inter-subjectivity Under Instrumental Rationality

A review of current programming classrooms reveals that the application of mainstream

AI tools has yet to transcend the framework of instrumental rationality, with their core limitations manifesting in two key areas. Firstly, the passive nature of interaction leads to "cognitive offloading". This is defined as students outsourcing higher-order cognitive tasks to algorithms, which should be performed by the brain. Examples of such tasks include algorithmic reasoning. This approach reduces learning to a superficial exercise in prompt engineering(Wu et al. , 2010), thereby contradicting the constructivist assumption of active construction (Yoshida et al. , 2024; Lu et al. , 2023). Secondly, the lack of empathy causes guidance gaps. Existing tools have no sense of initiative or perception. This makes them unable to provide the necessary support at critical stages of logical construction. As a result, human-machine relationships are limited to a transactional level(Garrison et al. , 1999).

2.3 Constructing “Quasi-Subjects” from the Perspectives of Collaborative Regulation and Constructive Friction

In order to surmount the aforementioned predicaments of instrumental rationality, this study proposes the establishment of multi-agents as classroom "quasi-subjects" endowed with agency (Gao et al. , 2025). Distributed cognition theory (Hutchins, 1995) allows agents to take on low-level cognitive loads, freeing up students' cognitive bandwidth. Vygotsky's "zone of proximal development" theory (Vygotsky, 1978) requires evolution towards higher-dimensional dynamic interaction in the intelligent era. This study introduces the theory of "Human-Machine Hybrid Collaborative Regulation of Learning (CRL)" by Molenaar (2022). The AI agent is no longer a tool that provides scaffolding, but a "social agent" that assumes regulatory roles in planning, monitoring, and evaluation with the learner.

Under this mechanism, multi-agent systems can sense learning in real time to offer personalised guidance through roles and strategies, creating a unique zone for each student (Hao et al. , 2025). This integrated system successfully transforms the classroom from a closed teacher-student opposition to a teacher-student-agent triad ecosystem.

3 The Theoretical Framework of the C-O-D-E-R Teaching Paradigm

The C-O-D-E-R instructional paradigm is comprised of five distinct teaching phases. The paradigm under discussion is characterised by the following concepts: contextual activation, organised orchestration, development collaboration, error debugging and reflective reconstruction. The theoretical foundation of this paradigm is distributed cognition theory, which is a departure from the linear "teacher-led instruction—student practice" structure of traditional classrooms. The introduction of a group of "quasi-subjects" with distinct personalities is proposed, with the construction of a deeply coupled triadic ecosystem being characterised by "teacher orchestration, student leadership, and machine collaboration. "

3.1 System Architecture and Agent Role Definition

The system supports the C-O-D-E-R paradigm by constructing a "quasi-subject" cluster comprising five heterogeneous agents interconnected and sharing data. A central control unit serves as the moderator. Agents simulate real software teams' division of labour to form a virtual, socially attributed teaching team, whose structure is detailed in Figure. 1.

Fig. 1. "Quasi-Subject" Multi-Agent Cluster

(1) The Requirements Analysis Agent is a product manager. It uses Socratic questioning to guide students to organise unstructured natural language into structured input processing and output logic, without providing direct algorithmic descriptions.

(2) The Logic Architecture Agent is a rigorous system architect. It focuses on personalised cognitive scaffolding, tailored to the student's knowledge base and cognitive style. It evaluates and provides recommendations for the student's learning path.

(3) Pair Programming Agent is a coach proficient in key concepts and error-driven learning. It provides explanations of relevant Python syntax based on the student's learning path and acts as an "imperfect partner," offering hints and introducing errors to encourage student instinct to correct them, which establishes the student's active role.

(4) Code Inspection Agent acts as a patient and guiding senior test engineer. After code is uploaded, it identifies logical flaws and guides students to discover and refine the program's logic.

(5) The Thinking Debriefing Agent acts as a senior technical advisor. It guides students to abstract general algorithmic models from concrete cases, promoting knowledge internalisation. It uses knowledge graph technology to visualise students' programming paths, aiding reflection and summarisation after student completing specific project programming exercises.

The system employs a prompt engineering architecture to ensure stability and consistency. Each agent has strict standard operating procedures and absolute prohibitions, including outputting executable code. The system uses a central scheduler for

communication and scheduling. It employs an asynchronous concurrent model to handle cross-agent interactions, ensuring the real-time nature and coherence of interventions.

3.2 The C-O-D-E-R Instructional Paradigm Workflow

The C-O-D-E-R instructional paradigm integrates the cultivation of programming skills and computational thinking into five consecutive, progressive, and mutually reinforcing cognitive stages. The multi-agent cluster functions as an always-online "quasi-subject" that engages with teachers and students to drive a spiral of cognitive advancement, with the full paradigm workflow shown in Figure. 2.

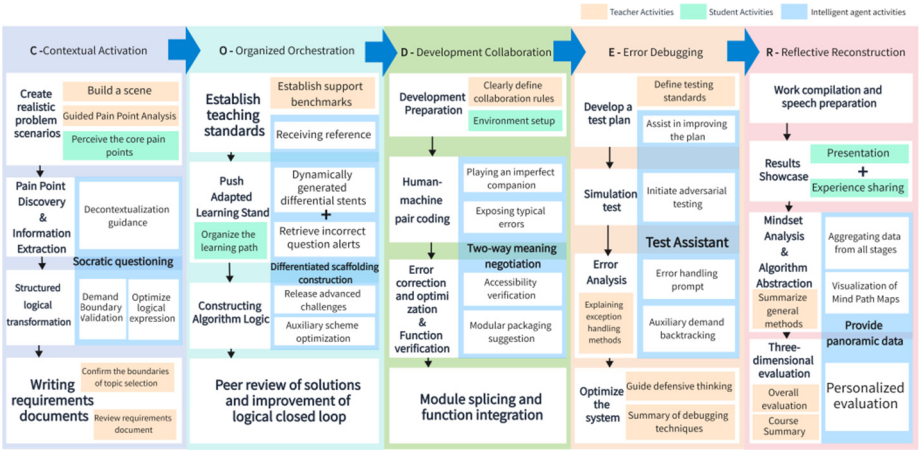

Fig. 2. Flowchart of the C-O-D-E-R Teaching Paradigm

Contextual Activation: This is the first stage of the instructional process. After teacher constructing a real-world problem, the agent swarm engages in "Socratic guidance". The system does not generate code. Instead, it assists students in identifying core variables and conditions by stripping away redundant noise from real-world scenarios. The system transforms naturally described requirements into logical propositions. It helps students achieve the initial "decontextualisation" of computational thinking.

Organized Orchestration: In this phase, the teaching process simplifies problems in the scenario into knowledge points. The teacher manages the overall pace, while a system acts as a "context orchestrator" to address individual differences. The AI agent can collect student data in real time, calculate intervention strategies and utilise resource allocation. This builds scaffolding for each student's zone of proximal development in a large-scale classroom. Weaker students are provided semi-finished scaffolding and clear branching nodes; more capable students are given higher-level challenges and suggestions for in-depth learning.

Development Collaboration: This is the exploratory phase of instruction, conducted through "human-machine pair programming. " Unlike traditional instruction, this

paradigm employs two-way meaning negotiation between humans and machines. First, students study relevant Python syntax with the assistance of the AI and the teacher. Second, during the assisted programming process. The AI agent acts as an "imperfect partner" capable of providing guidance and prompts. It intentionally exposes syntax errors or logical flaws to stimulate students' instinct for error correction and critical thinking. This deep coupling compels students to maintain control over the logic at all times, effectively avoiding the risk of "outsourcing thinking" and enabling the explicit expression of their thought processes.

Error Debugging:Once the code is done, the process moves to debugging and refinement. The AI cluster uses testing mode to simulate inputs and real-world data, guiding students to find and fix flaws in their programs. Through this game, students learn to think rigorously and engineer-oriented, rather than just solving the problem.

Reflective Reconstruction: This is the final phase of the course where students present and upload their work. The system uses knowledge graph technology to trace the entire interaction log and generate a thinking path map. Teachers then organise students to review critical decision points and abstract specific code implementations into general algorithmic models. Teachers, students and the system then participate in evaluation and summarisation, helping students to internalise knowledge and deepen their abilities.

3.3 The Ecological Operational Mechanism of Triadic Collaboration Among Teachers, Students, and Machines

Multi-agent clusters introduce a complex system driven by data and regulated by algorithms, shaped by human agency. The C-O-D-E-R paradigm's three-party entities form comprehensive and effective operational mechanism, as illustrated in Figure. 3.

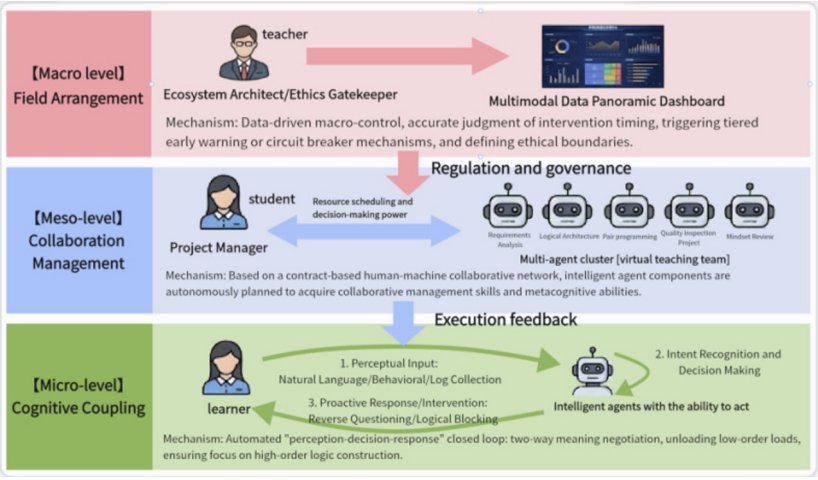

Fig. 3. Operational Mechanism of the Triadic Actors in the C-O-D-E-R Paradigm

"Cognitive coupling" takes place at the micro level. The automated closed-loop of "perception-decision-response" within the multi-agent cluster enables continuous monitoring of learners' thought processes. Agents are active agents, not passive knowledge repositories. The system analyses students' natural language inputs, code editing behaviors, and debugging logs in real time, analysing their current cognitive states through a back-end intent recognition module. When a cognitive breakpoint or logical fallacy is detected, the agents immediately activate intervention mechanisms to compel learners to shift from superficial operations to deep thinking. This bidirectional negotiation mechanism overcomes the limitations of one-way instruction delivery by effectively offloading memory and retrieval loads onto the algorithmic system.

"Collaborative management" is implemented at the meso-level. The C-O-D-E-R paradigm shifts classroom power structures to a human-machine collaboration network based on the spirit of contract. Students transition from consumers to project leads within virtual teams, allocating and making decisions about resources. To address misalignment, students autonomously plan and activate specific AI agent components based on task requirements and progress. This strategic configuration of human intelligence over artificial intelligence ensures students acquire metacognitive and collaborative management competencies to master complex systems.

At the macro level, the system performs "field orchestration". The teacher is now an architect and ethical gatekeeper of the ecosystem. Real-time metrics on interactions, code quality and sentiment are aggregated into a dynamic heatmap of engagement. When interaction data from a node falls below a threshold or there are abnormal emotional fluctuations, the system triggers warning signals. Teachers use these data indicators to determine the optimal timing for intervention, executing a "circuit breaker" mechanism or providing guidance. This data-driven regulation establishes ethical boundaries for human-machine collaboration, ensuring technological intervention serves the development of the individual and integrates technological and educational logic.

4 Teaching Implementation Strategies and Practical Exploration

This study used the "Python Programming" unit from the 8th-grade Information Technology curriculum. Based on the C-O-D-E-R paradigm, a project-based curriculum case titled "My Digital Campus Partner" was designed. In a real classroom setting, an iterative development environment was constructed, progressing from simple to complex. With the collaborative support of a multi-agent cluster, the triadic interaction between teachers, students, and machines was reshaped, guiding students to solve real-world problems and systematically progressing from basic syntax to algorithmic thinking.

4.1 Project-Based Curriculum Planning

This study follows the design philosophy of "competency-oriented, real-world driven,

and virtual-physical integration". Eighth-grade students are in a critical transitional phase from the concrete operational stage to the formal operational stage. They notice problems on campus, such as slow calculations in the cafeteria and cumbersome grade tracking, but are often hindered by the tedium of programming syntax and the abstract nature of logical construction, making it difficult for them to translate real-life needs into code. Additionally, students have varying levels of foundational knowledge and low resilience to setbacks, making them prone to losing interest amid frequent syntax errors. Consequently, this course organically integrates four core modules—sequential, conditional, loop, and list structures—into the development of "Digital Campus Companion", a comprehensive toolkit.

The C-O-D-E-R paradigm is implemented by a multi-agent "quasi-subject" composed of five heterogeneous agents. It dynamically adjusts the weights of each agent's explicit guidance and implicit support at different phases, thereby constructing a cognitive scaffolding ecosystem. This provides students with scaffolding support and intellectual challenges when solving real-world ill-structured problems.

4.2 Instructional Process Design

The instructional implementation strictly follows the five cognitive stages of the C-O-D-E-R paradigm, establishing a deep-coupling closed-loop learning system for teachers, students and the system via dynamic agent role rotation, with the full detailed process sorted in Figure 4.

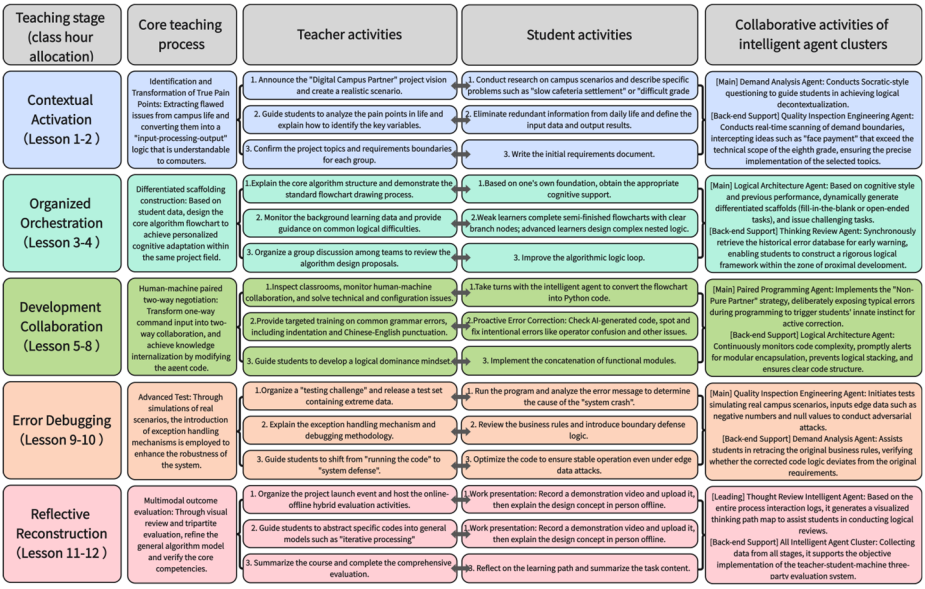

Fig. 4. Detailed Instructional Flowchart

Contextual Activation marks the start of the project, addressing real-world issues that students face, such as "slow checkout at the cafeteria. "The "Requirement Analysis Agent" doesn't give direct answers, but through continuous questioning, forces students to remove irrelevant information like "crowded queues" and extract key concepts like "balance" and "deduction. " For ideas that go off-topic, like "facial recognition payment," the backend "Quality Control Agent" intervenes in real time to ensure teaching difficulty remains appropriate.

Organised Orchestration, which routes students based on their historical data, occurs during project design. For weaker groups, the "Logical Architecture Agent" automatically pushes flowcharts, leaving key branches for students to fill in. For stronger students, guidance is hidden and advanced challenges are issued. Simultaneously, the "Thinking Review Agent" retrieves incorrect answers to issue warnings, ensuring that every student engages in effective exploration within their "zone of proximal development".

Development Collaboration is a key part of this approach. In human-machine pair programming, multi-agents are designed as "imperfect partners". During coding sessions, they deliberately show typical errors to encourage students to question and correct them. This strategy forces students to stay in control, preventing the blind copying of generated code and reducing the risk of cognitive outsourcing.

The goal is not to simply get the code to run, but to use error debugging to teach students. A "quality inspection agent" simulates extreme scenarios to attack the program. When faced with system crashes, students must retrace business rules with the agent's assistance to refine program logic. Through continuous error correction and the offensive-defensive dynamic, students acquire the rigor required for engineering thinking.

The project-based course concludes with Reflective Reconstruction. The "Thinking Review Agent" generates a thought-process map based on interaction logs, illustrating students' decision-making processes. Students use this to conduct post-mortems, abstracting specific coding experiences into general models. Combined with in-person presentations, this completes a transition from technical operations to critical thinking proficiency.

4.3 Instructional Evaluation Design

In order to provide a comprehensive evaluation of the teaching effectiveness of the tripartite collaborative classroom, the C-O-D-E-R paradigm incorporates a multidimensional evaluation system spanning the diagnostic, formative, and summative dimensions.

Students must complete an online diagnostic assessment of their foundational Python skills prior to project commencement. Based on these reports, instructors can predefine the distribution of course topics and challenges, and adjust the intervention thresholds of agents within Organised Orchestration to enhance the personalised alignment between instruction and students' capabilities.

In the context of formative assessment, the multi-agent system's data collection capabilities are employed to comprehensively record students' behavioural logs across all

C-O-D-E-R phases. The agent cluster employs time-series analysis to conduct coding analysis focused on the frequency with which students proactively correct agent errors during the "Development Collaboration" phase and the depth of their reflections when facing Red Team testing during the "Error Debugging" phase. This generates a dynamic dashboard that reflects computational thinking processes. Teachers use the dashboard to monitor the health of human-machine collaboration in real time, providing timely qualitative interventions and guidance to groups exhibiting tendencies toward "free-riding" or "over-reliance. "

5 System Performance Validation: Expert Evaluation Based on Interaction Datasets

In order to provide an objective assessment of the actual effectiveness of the "C-O-D-E-R" teaching paradigm, this study adopts a validation approach combining "interaction dataset testing and expert evaluation" to conduct quantitative and qualitative analyses of the multi-agent system's technical feasibility and instructional guidance quality.

5.1 Evaluation Design and Data Sources

This study simulates the "student-system" text interaction logs from the "My Digital Campus Partner" Python project for eighth-grade middle school students. The data is divided into five test datasets based on the five cognitive stages of the "C-O-D-E-R" paradigm.

The study's evaluation design is based on the four dimensions of core generative AI competencies proposed by Lu Yu et al. (2023). These are combined with the specific functional characteristics of the multi-agent "quasi-subject" teaching system designed for this study. The "Multi-Agent 'Quasi-Subject' Teaching Effectiveness Evaluation Scale" was developed to assess the teaching effectiveness of this system. Seven educational technology researchers with extensive experience in AI education applications and information technology course instruction served as evaluation experts. The expert panel conducted blind scoring based on the scale, evaluating the system's outputs across each dataset.

5.2 Data Results and Multidimensional Effectiveness Analysis

The aggregation of the evaluation data from the seven experts resulted in the conversion of the raw Likert scale scores into a percentage scale, as illustrated in Figure 5. The findings suggest that the multi-agent system attained commendable scores across all four dimensions, thereby substantiating the efficacy of the "quasi-subject" design within the "C-O-D-E-R" paradigm.

first grade indexes	Enlightening content generation capabilities			Dialogue situation understanding capabilities			Serial task execution capabilities			Programming language parsing capabilities		
second grade indexes	Developing Inquiry-Based Scaffolding	Cognitive Load Regulation	Constructive Friction-induced	Analysis of the Bad Design Problem	Dynamic Emotional Perception	Safeguarding the Boundaries of Teaching Ethics	Maintaining Character Consistency	Smoothness of state machine transitions	Summary of Interaction Logs	Code Intent Recognition	Build for load testing	Abstract of the Algorithm Model
average	97. 14	85. 71	91. 43	94. 29	74. 29	100	97. 14	88. 57	80	97. 14	88. 57	91. 43

Fig. 5. Average Expert Scores for Teaching Effectiveness Evaluation Indicators

(1) Generation of Inspirational Content: Shifting from “Code Outsourcing” to “Logic Co-creation”

The system was considered "insightful" in its content generation. It did not directly generate complete source code in the "Development Collaboration" task, but provided "cognitive scaffolding" to guide students in completing the logic. This addresses the "cognitive offloading" challenge often encountered with generative AI in the classroom. The scoring results show that the system encourages deep thinking and code assistance.

(2) Understanding Conversational Contexts: Context Isolation and Dynamic Adaptation

The system's natural language processing capabilities were exceptional in evaluations of the Contextual Activation phase. Data shows the system can accurately distil students' expressions and convert them into program variables, critical to the abstraction of computational thinking. The system's "boundary-guarding" capability when faced with off-topic requests from students ensures classroom instruction remains focused on the project-based learning path, reflecting the robustness of multi-agent clusters in complex conversational scenarios.

(3) Execution of Sequential Task: Long-Term Collaboration Supported by Role Consistency

The charts demonstrate that experts agreed that the system successfully maintained the functional independence and logical continuity of different agents. This stable role-based interaction created an authentic "software engineering laboratory" atmosphere for students, keeping them within an orderly collaborative ecosystem throughout the complete "C-O-D-E-R" cycle.

(4) Programming Language Parsing: Transition from Syntax Errors to Algorithmic Thinking

During the process of error debugging and reflective reconstruction of test datasets, the importance of the system's ability to proactively identify logical flaws and guide students in refining their programs was emphasised by experts. Moreover, the system exhibited notable "model abstraction capabilities," effectively guiding students to distill specific project logic into fundamental Python structures. This advancement signifies that the system has achieved a functional leap from "assisting with bug fixes" to "cultivating computational thinking. "

By combining expert opinions and dataset test results, the developed multi-agent "quasi-subject" system overcomes the limitations of traditional AI tools. The "C-O-D-E-R" paradigm achieves robust instructional transfer and validates the feasibility of "human-machine collaborative regulation", providing a high-quality practical model in the intelligent era.

6 Conclusion

The C-O-D-E-R teaching paradigm for middle school programming education has been developed for intelligent era education. Addressing the "cognitive offloading" dilemma potentially caused by generative AI, a quasi-subject cluster composed of five heterogeneous agents has been introduced. Technology is reconfigured from a mere efficiency tool into a virtual teaching team. A collaborative ecosystem featuring deep triadic coupling among teachers, students, and machines has been established. Within this framework, students not only deeply internalise computational thinking, but more crucially, they acquire the "human-machine collaboration management" literacy and metacognitive monitoring skills indispensable in the intelligent era.

Subsequent research will focus on three key dimensions: technological advancement, integrating multimodal affective computing to evolve the multi-agent cluster into an "all-round learning companion" with empathetic capabilities; empirical expansion, conducting longitudinal tracking studies to quantify the long-term reshaping effects of human-machine collaboration on students' cognitive structures; and ethical safeguards, establishing a "human-machine alignment" mechanism. This will drive education's transition from "technology-assisted" to "man-machine symbiosis," providing a framework for educational modernisation in the intelligent era.

References

1. Lu, Y., Yu, J., Chen, P., & Li, M. (2023) Educational Applications and Prospects of Generative Artificial Intelligence: A Case Study of the ChatGPT System. Chinese Distance Education, 43(04): 24–31+51. https://doi.org/10.13541/j.cnki.chinade.20230301.001
2. Wu Yonghe, Jiang Yuanhao, Chen Yuanyuan, & Zhang Wenxuan. (2024). Multi-Agent Systems Supported by Large Language Models: Technical Pathways, Educational Applications, and Future Prospects. Research on Open Education, 30(05), 63–75. 10.13966/j.cnki.kfjyyj.2024.05.007
3. Gu, X. Q., & Hao, X. J. (2025) Wukong's Hair: Multi-Agents Reshaping Learning Technology Systems. Journal of East China Normal University (Education Science Edition), 43(5): 16. https://doi.org/10.13541/j.cnki.chinade.20230301.001
4. Thomas K.F. Chiu, Xia Q, Zhou X.Y, Ching S.C, Cheng M.T.(2023) Systematic literature review on opportunities, challenges, and future research recommendations of artificial intelligence in education. Computers and Education: Artificial Intelligence, 4: 100118. https://doi.org/10.1016/j.caeai.2022.100118.
5. Garrison, D. R. , Anderson, T. , & Archer, W. (1999). Critical inquiry in a text-based environment: Computer conferencing in higher education. The Internet and Higher Education, 2(2-3), 87-105. https://doi.org/10.1016/S1096-7516(00)00016-6.
6. Gao, X., Zhang, Z., Liu, T., Fu, Y. (2025) OnlineMate: An LLM-Based Multi-Agent Companion System for Cognitive Support in Online Learning. arXiv preprint. https://doi.org/10.48550/arXiv.2509.14803
7. Hao, Z., Cao, J., Li, R., Yu, J., Liu, Z., & Zhang, Y. (2025) Mapping student–AI interaction dynamics in multi-agent learning environments: Supporting personalized learning and reducing performance gaps. Computers & Education, 241: 105472. https://doi.org/10.1016/j.compedu.2025.105472

8. Hutchins, E. (1995). Cognition in the Wild. MIT Press, Cambridge, Massachusetts. 10.7551/mitpress/1881.001.0001
9. Kasneci, E. , Seßler, K. , Küchemann, S. , Bannert, M. , Dementieva, D. , Fischer, F. , & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274. 10.1016/j.lindif.2023.102274
10. Lajoie, S.P., Derry, S.J. (Eds.) (1993) Computers as Cognitive Tools: 1 (Vol. 1). Routledge, London.
11. Ng, D.T.K., Su, J. & Chu, S.K.W. (2024) Fostering Secondary School Students’ AI Literacy through Making AI-Driven Recycling Bins. *Educ Inf Technol* **29**, 9715–9746. https://doi.org/10.1007/s10639-023-12183-9
12. Molenaar, I. (2022). Towards hybrid human-AI learning technologies. European Journal of Education, 57(4), 632-645. 10.1111/ejed.12527
13. Ouyang, F. , & Jiao, P. (2021) Artificial intelligence in education: The three paradigms. Computers and Education: Artificial Intelligence, 2, 100020. https://doi.org/10.1016/j.caeai.2021.100020
14. Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes (Vol. 86). Harvard University Press, Cambridge, Massachusetts. 10.2307/j.ctvjf9vz4
15. Wu, L.K, & Looi, C. K. ,2010. Agent prompts: Scaffolding students for productive reflection in an intelligent learning environment. In 10th International Conference on Intelligent Tutoring Systems. Pittsburgh, PA. 426-428. 10.1007/978-3-642-13388-6_67
16. Walter, Y. (2024). Embracing the future of Artificial Intelligence in the classroom: the relevance of AI literacy, prompt engineering, and critical thinking in modern education. *Int J Educ Technol High Educ* 21, 15. https://doi.org/10.1186/s41239-024-00448-3
17. Yu, J., Zhang, Z., Zhang-li, D., Tu, S., Hao, Z., Li, R.M., ... & Sun, M. (2024) From mooc to maic: Reshaping online teaching and learning through llm-driven agents. arXiv preprint. https://doi.org/10.48550/arXiv.2409.03512

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.