Original Research Article

Reconstruction and practice of the PBL teaching model for operating systems under digital transformation

Long Zhao
Department of Computer Science, Inner Mongolia Honder College of Arts and Sciences, Hohhot, Inner Mongolia, 010070, China

Abstract: This study focuses on structural contradictions in traditional Operating Systems laboratory teaching, such as excessive emphasis on verification, insufficient industry alignment, and a single evaluation system, and conducts a systematic teaching reform in the context of digital transformation. By integrating PBL (Problem-Based Learning) with the 5E inquiry-based teaching model, it constructs a novel teaching loop of 'real-life scenario introduction — Tiered and stratified experiments — Data-driven intelligent assessment.' Empirical data show that this model effectively stimulates the learning motivation of students upgrading from junior college to undergraduate level, significantly enhances their system programming skills and ability to solve complex engineering problems, and provides a replicable practical paradigm for digital teaching reform in science and engineering courses at local application-oriented universities.
Keywords: digital transformation; operating system; PBL teaching model; 5E teaching method; reconstruction of experimental teaching

1. Introduction

Driven by the construction of emerging engineering education and the digital education strategy, operating systems, as a core course in computer science, have long suffered from issues such as abstract concepts and difficulty in cultivating higher-order abilities[1]. Particularly for students from associate-to-bachelor programs, traditional "lecture + demonstration" experimental teaching suffers from a severe "emphasis on verification, neglect of design" path dependency, making it difficult to adapt to students' diverse foundations and lack of engineering experience, leading to a misalignment between teaching objectives and industry demands. Existing reforms often remain superficial, lacking a profound analysis of internal contradictions in teaching elements[2]. Although the PBL model is widely applied in engineering education reforms due to its effectiveness in stimulating learning initiative[3], traditional implementation often faces challenges such as loose processes and delayed evaluation. To address this, this study reconstructs from four dimensions—Content, environment, mechanism, and evaluation—To build a "PBL teaching model for operating systems under digital transformation." By introducing life-oriented scenarios, layered task adaptation, and data-driven evaluation, it achieves a transformation from "knowledge transmission" to "capability generation," providing a reform paradigm for application-oriented universities.

2. Theoretical logic and integration mechanism of teaching model reconstruction

Given the strong abstraction and high practicality of operating system courses, this study constructs a theoretical framework that integrates PBL and the 5E teaching method, and clarifies the empowering role of digital transformation within it.

2.1. Dual-drive coupling of PBL and the 5E instructional model

This study constructed a "dual-drive coupling" mechanism that deeply integrates PBL with the 5E teaching model, using the PBL "problem chain" as the engine to anchor objectives and the 5E "inquiry loop" as the path to ensure progressive teaching, thereby reshaping the structure of operating systems experimental teaching. In the Engage stage, life-based scenarios are used to trigger cognitive conflict, transforming perceptual experiences into motivation to explore core concepts such as process scheduling. In the Explore stage, relying on virtualization and collaboration platforms, verification is replaced with "learning by doing," enabling students to bridge the gap from theory to solutions through trial-and-error and debugging. In the Explain stage, AI tools are employed to visualize system operations, linking experimental phenomena with kernel principles to achieve knowledge internalization. In the Elaborate stage, layered tasks are used to promote the migration of abilities from basic operations to complex engineering problems. In the Evaluate stage, the platform's full-cycle data is used to construct multi-dimensional profiles, shifting focus from results to process value-add, thus implementing "assessment to promote learning."

2.2. Empowerment mechanism of digital transformation

2.2.1. Cognitive concretization: From abstract symbols to visual imagery

Concepts such as operating system kernel mode, interrupt vector table, and memory fragmentation, due to their high level of abstraction and invisibility, have long constituted major cognitive obstacles for students upgrading from junior college to university. Digital technology here acts as a 'perspective lens,' reconstructing the invisible system call process through the introduction of Tencent digital humans and virtual simulation animation in three-dimensional visualization. For example, when explaining 'process state transitions,' dynamic graphics are used to intuitively display the transfer paths of the PCB (Process Control Block) between ready, running, and blocked queues. This visual presentation significantly reduces students' cognitive load, achieving a cognitive leap from 'obscure code on the blackboard' to 'dynamic real scenes before their eyes,' effectively bridging the gap between theoretical principles and engineering practice.

2.2.2. Collaborative connectivity: From isolated experiments to ubiquitous coordination

Traditional computer lab experiments are limited by physical space and fixed terminals, leading to noticeable delays and loss in information transmission between teachers and students. This study leverages collaborative platforms such as Feishu and DeepSeek to construct a ubiquitous "teacher–student–machine" triadic interactive space. This not only addresses the pain points of complex Linux environment configuration and dependence on specific hardware, but also achieves real-time accumulation of experiment logs and asynchronous collaboration. Students can continue their classroom code progress after class, using AI assistants for syntax correction and idea inspiration; teachers can break through the walls of the computer lab to track group discussions in real time. Digital media thus dismantles the boundaries of traditional laboratories, providing a stable and efficient technological foundation for in-depth discussions under the PBL model.

2.2.3. Precise evaluation: From intuitive experience to data profiling

By moving away from the traditional subjective approach of grading solely based on lab reports, the digital platform serves as an objective and impartial 'recorder' and 'analyst.' The system automatically captures behavioral data such as the number of times students compile code, types of errors encountered, debugging duration, and API usage patterns, forming a full-cycle learning trajectory. This data-driven evaluation model provides solid evidence for process-based assessment. Through mining and analyzing this data, teachers can accurately identify students' weaknesses in algorithmic logic or systems programming, thus shifting from generalized, one-size-fits-all feedback to personalized interventions tailored to each student, ensuring that no one falls behind on the track of digital transformation.

3. Reconstruction of the operating system PBL teaching model based on digital transformation

Based on clarifying the coupling logic between PBL and the 5E model as well as the mechanism of digital empowerment, the teaching team focuses on the practical implementation level of the model. From four dimensions—Teaching content, structural design, environment setup, and evaluation system—They systematically reconstruct the operating systems experimental teaching model, aiming to build a complete teaching closed loop that is adapted to the learning conditions of junior college to undergraduate transfer students and aligned with the requirements of emerging engineering education.

3.1. Layered content reconstruction driven by everyday life scenarios

3.1.1. Scene mapping: Transition from life semantics to system semantics

Traditional operating system experiments often directly present concrete API functions, causing students to "see the trees but not the forest." This study adopts a reverse design approach, mapping abstract system principles to everyday life scenarios familiar to students. For example, "process synchronization and mutual exclusion" is mapped to "window scheduling when queuing for meals in the cafeteria," and "deadlock" is mapped to "traffic congestion at an intersection." This semantic transformation greatly lowers the cognitive threshold, enabling students to grasp the engineering problems they are about to solve at a logical level before encountering complex code.

3.1.2. Third-level progression: Building a differentiated experimental task chain

To avoid a one-size-fits-all approach that leads to gifted students not being sufficiently challenged and weaker students falling behind, the restructured experimental content adopts a three-level progressive framework: Basic Level — Advanced Level — Challenge Level. At the Basic Level (verification type), the focus is on command operations and API calls, such as 'Process Creation Experiment in a Linux Environment', aiming to strengthen fundamental operational skills. At the Advanced Level (design type), practical scenarios are integrated to pose questions, such as 'Cafeteria Queue Simulation Based on the Producer-Consumer Model', requiring students to master semaphore mechanisms and multi-thread programming. At the Challenge Level (comprehensive type), the design aligns with industry needs, such as 'Implementation of Lightweight Container Scheduling Strategies Based on Docker', encouraging students to perform algorithm optimization and performance tuning. This tiered design gives students the freedom to choose, enabling them to achieve spiraling growth in capability within their zone of proximal development.

3.2. "One Core, Dual Drive, Triple Closed Loop" model architecture design

3.2.1. Pre-class self-study loop: Knowledge warm-up and diagnosis

Relying on micro-lesson resources and an online testing platform, the teacher releases the 'Cafeteria Window Scheduling Optimization Problem' (as shown in **Figure 2**) before class. Students complete independent learning of the prerequisite knowledge by watching an animated demonstration and submit an online questionnaire. The system automatically grades and generates a learning diagnosis report, providing data support for precise teaching during class, thereby achieving a shift from 'teach first, learn later' to 'learn first, teach later'.

3.2.2. In-class practical training loop: In-depth exploration and collaboration

This is the core link of the model. The teacher uses animation to demonstrate critical section conflicts and raises the core question of "how to avoid deadlock"; students work in groups to set up a virtual environment using VMware and write multi-threaded programs. During this process, the teacher uses DeepSeek to assist students in debugging and provide personalized guidance. Through the iterative cycle of "scenario introduction — Group discussion — Code implementation — Debugging and troubleshooting," knowledge is internalized and skills are consolidated.

3.2.3. Post-class extension loop: Ability transfer and innovation

Leveraging discipline competitions and open-source community resources, students are encouraged to transform their in-class experimental results into competition entries or open-source projects. This closed loop breaks the boundaries between in-class and extracurricular activities, enabling the transfer of skills from 'course assignments' to 'engineering practice', and fostering a virtuous cycle of mutual enhancement in teaching and learning.

3.3. Integrated reconstruction of a multimodal digital support environment

3.3.1. Seamless integration between virtual simulation and cloud platform

To address the pain points of complex operating system lab environment configuration and the risk of damaging the underlying system, VMware virtualization technology and the Tencent Cloud lab platform are fully introduced. Students can deploy a standard Linux lab environment with a single click via a browser, eliminating the complicated process of dual-system installation and driver configuration. This not only greatly reduces ineffective time loss outside of teaching but also enables the possibility of remote labs and ubiquitous learning.

3.3.2. Deep integration of intelligent assistants and collaboration tools

Deeply embed AI assistants such as Feishu DeepSeek into the teaching process. During the code debugging phase, AI can analyze syntax errors in real time and provide modification suggestions; during the discussion phase, AI can serve as a 'Socratic interlocutor,' guiding students to discover logical flaws through counter-questions. Meanwhile, Feishu's online document feature enables collaborative editing and version control of lab reports, ensuring efficiency and transparency in team collaboration.

3.4. Reconstruction of a data-driven diversified evaluation system

Abandoning the single final examination system, a data-driven evaluation framework with the triad of "process + ability + value" has been established, shifting the focus of assessment from "result certification" to "ability enhancement." This framework redefines the dimensions and weightings of evaluation. Process performance accounts for 40%, primarily assessing attendance rate, online test scores, and participation activity in group discussions, with data sourced from log extraction on the Learning Pass/Feishu platforms. Ability attainment also accounts for 40%, focusing on the standardization of experimental reports, code quality, and performance in defenses, using a combination of teacher scoring and peer evaluation. Innovation value accounts for 20%, emphasizing the degree of algorithm optimization and the capability to resolve unexpected faults, jointly determined by AI code analysis tools and teacher subjective assessment. This framework not only considers whether "learning has occurred" but also how much "progress has been made," fully reflecting the fairness and scientific nature of educational evaluation under digital transformation.

4. Teaching practice and effectiveness analysis

The construction of theoretical models ultimately needs to be validated through practice. To verify the feasibility and superiority of the "PBL Teaching Model for Operating Systems under Digital Transformation," this study selected two parallel classes from the 2024 cohort of Computer Science and Technology (top-up program) at our college for a one-semester teaching control experiment. The experimental class (Class A, 62 students) adopted the reconstructed "PBL+5E" dual-drive integration model, while the control class (Class B, 62 students) continued with the traditional "lecture+verification" model. The following analysis will be conducted from two dimensions: Specific implementation cases and quantitative evaluation data.

4.1. Quantitative analysis of teaching effectiveness

Quantitative evaluation of teaching effectiveness shows that the PBL model based on life-oriented scenarios has a significant positive impact on knowledge acquisition, skill improvement, and learning attitudes. Post-test questionnaire data indicate that the improvement rate of experimental class students' understanding of core operating system concepts reached 92.5%, significantly higher than the control class's 67.8%; the improvement rate of system programming skills was 94% (control class 58.2%), and the ability to solve complex technical problems improved by 83.7%, 31 percentage points higher than the control class. Final exam results further verified the model's advantages: The experimental class had an average score of 78.6, a pass rate of 93.5%, and an excellence rate of 27.4%, all significantly higher than the control class (65.3, 75.8%, and 11.3% respectively), and independent sample t-tests confirmed the differences were statistically significant ($p<0.05$). Additionally, learning attitude and participation data revealed that 83.6% of experimental class students recognized diversified assessment methods, and 79.5% affirmed group project work, significantly higher than the control class's 45.2% and 52.4%; behavioral monitoring data showed that the experimental class had higher classroom interaction frequency, better experimental task completion quality, and longer autonomous learning time than the control class, indicating that the digital PBL model effectively stimulated students' learning enthusiasm through gamified task design and instant feedback mechanisms.

4.2. Qualitative analysis of teaching effectiveness

Qualitative data, obtained through semi-structured interviews, further revealed the mechanisms of the teaching reform and students' learning experiences. The interviews showed that students generally gave high evaluations to the "life-based scenarios + tiered experiments" teaching design, believing it effectively lowered the learning threshold for abstract concepts. For example, one student mentioned understanding the process priority inversion problem by analogy with cafeteria window scheduling, which helped detach operating system knowledge from a purely theoretical framework and increased its concreteness and engagement. The auxiliary functions of the digital platform also received positive feedback, particularly the DeepSeek intelligent assessment system's instant error correction and related micro-lecture recommendations, which significantly improved learning efficiency; the stability of the VMware virtualization environment addressed the tedious configuration issues of traditional lab settings. In addition, through group collaboration and project-based practice, students perceived an improvement in their comprehensive abilities, including team collaboration awareness, problem analysis pathways, and the formation of systematic solutions. One student pointed out that the complete inquiry process—From observation of phenomena, to hypothesis formulation, to experimental verification—Not only reinforced learning outcomes in this course but also provided a transferable methodological framework for learning in other courses.

Fundings

Inner Mongolia Honder College of Arts and Sciences institutional re-search project "Application Research on Digital Experimental Teaching of 'Operating Systems' under the PBL Model" (HD2024034).

About the author

Long Zhao (1987.01–), male, Hohhot, Inner Mongolia, Master's candidate, Lecturer at Inner Mongolia Honder College of Arts and Sciences, majoring in computer education.

References

[1] Zhang Xianzhong. Exploration of the Application of Problem-Based Learning in the "Operating Systems" Course[J]. Journal of Changzhou Institute of Technology. 2018, 31(03): 88-91.

[2] Deng Song, Yang Lechan. Practice and Exploration of Agile Teaching in Operating System Courses[J]. Software Guide. 2022, 21(11): 177-181.

[3] Yu Qun, Wang Xiuli, Yang Xiaoxia, et al. Discussion on the Application of PBL Combined with TBL in the Teaching of "Principles of Operating Systems"[J]. Journal of Shandong Agricultural Engineering College. 2019, 36(03): 159-161.