A Practical Exploration of CDIO Engineering Education Model in Teaching of Fundamental of Software Technology

Rongli Tang

School of Business, Xi'an International University, Xi'an, China

happytrl@163.com

Abstract. This study introduces CDIO education concepts into the teaching of the "Fundamentals of Software Technology" curriculum for non-computer majors, starting from several aspects such as course conceive and design, course implementation cases, and course operation experience, discusses in detail the three major components of the course, namely, Data Structures, Operating System, Software Engineering, have achieved initial success at present, finally summarizes the CDIO teaching practice.

Keywords: Fundamental of Software Technology, CDIO, Teaching practice, Non-computer major.

1. Introduction

The "Fundamentals of Software Technology" is divided into three major parts: data structure, operating system, and software engineering. The core course of this major is relatively abstract, with a high degree of difficulty, making it difficult for students to master. This course requires students to have a foundation in advanced mathematics, such as the time and spatial complexity of algorithms. It also requires students to have a good programming foundation, especially pointers, functions, recursion, structures, classes, and objects. After studying in previous years, students were able to apply simple data types and the three basic program structures in structured design with ease. However, their understanding of construction types (arrays, structures, commons, enumerations) and pointer types remained superficial and could not be flexibly applied. This part was precisely throughout this course. Therefore, the above problems have resulted in many students not being proficient in practical applications, and the teaching effect is not ideal.

CDIO is an advanced modern engineering education concept founded by the Massachusetts Institute of Technology, which has been widely applied worldwide. As a core course for non-computer majors, this course requires students to have a certain foundation in advanced mathematical theory and programming skills. How-ever, our department's students do not have a solid grasp of the leading courses, and are even more afraid of technical courses. Therefore, we propose a reform approach to the teaching mode of "Fundamentals of Software Technology" based on the CDIO concept.

2. Scientometric Analysis

The data used to analysis in our research is downloaded from WoS, and the search strategy followed is below:

(1) Themes = ("Teaching Reform" AND "CDIO");

(2) Database = Science Citation Index Expanded (SCI-EXPANDED), Social Sci-ences Citation Index (SSCI), Conference Proceedings Citation Index-Science (CPCI-S), Conference Proceedings Citation Index-Social Science & Humanities (CPCI-SSH), Emerging Sources Citation Index (ESCI);

(3) Timespan = "1990-2023";

(4) Document types = "Article" or "Editorial Material" or "Proceedings Paper" or "Review";

(5) Literature type = "English";

303 publications are retrieved, and finally 225 publications are valid data, which were downloaded on August 31st, 2023.

Table 1. WoS Categories Distribution of Teaching Reform and CDIO

Num	WoS Categories	Numbers of Category	Proportion
1	Education Educational Research	98	44.954
2	Social Sciences Interdisciplinary	68	31.193
3	Computer Science Theory Methods	39	17.890
4	Education Scientific Disciplines	35	16.055
5	Management	28	12.844
6	Computer Science Interdisciplinary Ap-plications	21	9.633
7	Computer Science Information Systems	18	8.257
8	Engineering Multidisciplinary	11	5.046

From the perspective of WoS categories distribution, Table 1 shows that “teach-ing reform and CDIO” is a multidisciplinary research field, which mainly involving Education Educational Research (44.95%), Social Sciences (31.19%) and Computer Science (17.89% of the total).

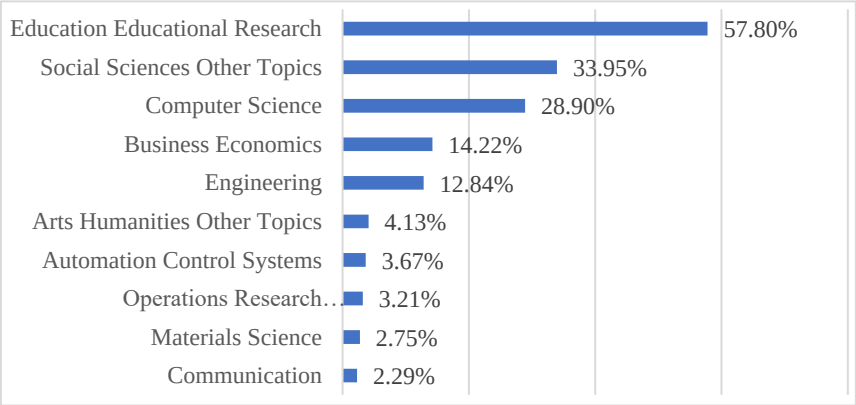

Figure 1. WoS Categories Distribution of Teaching Reform and CDIO

With regard to WoS research direction, as shown in Fig. 1, the visualization re-sults are consistent with the previous Table 1.

The prior relevant studies can be summarized in the following aspects:

In terms of discipline construction and talent cultivation, Xueli Wang et al. (2022) integrated the CDIO engineering education model into the computer teaching, conducted questionnaire survey between the traditional educa-tion model and the reformed education model, found that the CDIO engineering ed-ucation model has a significant advantage. Gongwen Xu et al. (2021) introduced CDIO and OBE concept, adopted this educational model in the construction of E-Commerce majors to cultivate students' practical and innovative abilities.

In terms of curriculum reform and practice, Xu Guan et al. (2023) have taken Artificial Intelligence (AI) course of Liaoning In-stitute of Science and Engineering as an example, following the concept of CDIO engineering education and five learning methods (problem-based learning, pro-ject-based learning, case-based learning, do-it-yourself learning, and experience-based learning). Dong Yuxia (2022) has launched a teaching reform of CDIO engineering education model for Java course teaching, cultivating innovative and applied talents with good professional skills and professional literacy. Cai Yang (2020) enables students to learn Java programming course in an active, practical and comprehensive way, CDIO engineering education teaching effect of the model was verified by data statistics, the survey questionnaire results show that students who use the new method perform better. Haiyi Jin (2019) used a variety of course assessment meth-ods based on OBE-CDIO teaching design of Network Planning and Design course, took a chapter of WLAN design as an example, focused on engineering practices and related design points.

Other teaching reform methods, Cesare Aloisi (2023) believes that using artificial intelligence and machine learning to assist in high-risk assessment of exams is effective, including grading, quality assurance, and paper generation. Using gamified courses as the treatment group and non-gamified courses as the control group, Raquel Blanco et al. (2023) investigated whether gamified

usage techniques help improve the participation and performance of software testing students, found that gamified courses have a positive impact on students' participation and academic performance.

In the fields of management and business, George Rosier (2022) promoted online survey of case method course on the Case Centre at Cranfield, UK, the participants were scholars of business schools in Aus-tralia and Singapore, these findings of case method teaching have beneficial effect for business education practice. Aron Truss et al. (2023) contributed to blended learning pedagogy in business school circumstances, adopted semi-structured inter-view methods before and after COVID-19, suggested implications for business and management education. Xiaotao Song (2022) contributed to online-offline hybrid teaching reform model for software testing course. Leonidas Efthymiou (2021) pro-posed a transferable model for current technology and online pedagogy learning in business and management, the sample emerged from management students from 154 countries, concluded that student feedback at the end of each course is even more critical in online education.

This study introduces the CDIO education concept into the teaching of the "Soft-ware Technology Fundamentals" curriculum for non-computer majors. Starting from several aspects such as course concept and design, course implementation cases, and course operation experience, the author discusses in detail the three major compo-nents of the course, namely, data structure, operating system, and software engi-neering.

3. Conceive and Design

3.1. Data Structure

As the core part of this course, data structure has both abstract theoretical segments and practical segments emphasizing algorithm programming. The logical structure, physical (storage) structure and operational operation of commonly used data structures are taken as the main teaching contents, this study integrates the training of CDIO thinking ability into the course teaching, uses questions to stimulate students' learning enthusiasm, and designs each segment well, let the students feel that the knowledge is useful, interesting and effective.

Table 2. Some of the Questions to Import the Data Structure

Main Content	Questions Raised	Corresponding Knowledge Point
Algorithm	What is a good algorithm? How to measure the efficiency of algorithms as high or low?	The time complexity and spatial complexity of the algorithm.
Linear List	How to establish a student grade transcript and how to operate it?	Storage and basic operations of linear tables.
Stack	Nested calls to functions.	Basic operation of stack.
Queue	How to design a simulation plan for queuing to buy tickets? Input output buffer?	Basic operations of queues.
Tree and Binary Tree	How to establish and search for a family tree? What is the list of textbooks we use? Mining the relationship between full binary trees and binaries?	Basic concepts of trees and binary trees, traversal of binary tree.
Graph	What is the matter of the "Seven Bridges of Gothenburg"? Graph theory model and algorithm for course scheduling in universities?	Basic concepts of graphs. Minimum spanning tree, directed acyclic graph and their applications.

The new teaching mode integrates the CDIO concept into the data structure teaching, adopts the questioning teaching method, introduces the designed questions into the teaching, and allows students to actively think about the solution of the problem by asking questions. Based on the expected

learning results, reverse design of teaching content and use of questions to stimulate students' learning enthusiasm are shown in the following table.

3.2. Operating System

At first, interact with students regarding the operating system of mobile phones, and then step into the operating system of computers, which is the second part of "Fundamentals of Software Technology". As an important branch of computer related majors, this section involves a wide range of aspects, abstract processes, complex content, and strong theoretical significance, making it difficult to truly understand and master, therefore, this part is the "most difficult to teach" for teachers and the "most difficult to learn" for students. This study integrates the cultivation of CDIO thinking ability into the teaching process of this part, arouses students' enthusiasm for learning, and changes the situation that students are "unable to learn" and "unwilling to learn", has good teaching effect.

The import settings for some operating system issues are as follows:

No matter how much memory is installed on 32-bit system, it can only use 4GB. Is that true?

No way... So I bought 16GB memory module for nothing?

Interact with students and encourage them to calculate 2^{32}=?

4. Implement

4.1. Classic P/V Operations, Status of Processes (Operating System)

Adopting situational teaching methods, designing learning contexts is used to improve teaching effectiveness, this teaching method has a certain degree of operability, highlighting students' learning by doing and doing while learning. In "learning by doing", emphasis is placed on hands-on learning, requiring students to understand and comprehend the process of doing, and then acquire corresponding skills. In "doing by learning", students need to first master some targeted skills systematic theoretical knowledge.

The state of the "process" is the key and difficult part of the operating system. We integrate abstract, obscure theoretical knowledge into pre-created story scenarios to avoid teacher centered and lecture based cramming, which is also one of the key points that trigger CDIO.

The story scenario presents the fundamental knowledge intuitively and vividly in our vision, making students realize that abstract operating system knowledge supports specific life points. Next, we will discuss and sort out the knowledge points behind the story, this teaching mode is more effective than simple textual explanations, and it also makes students aware of the significance branch of "Software Technology Fundamentals", namely, operating system, the seemingly dull theory is actually full of fun, which has led to more interaction in the classroom. Besides, it has also stimulated students' interest and enthusiasm for learning.

4.2. Small Design Oriented (Data Structure)

If there are six competition events, it is stipulated that each contestant can participate in up to three events, with five registering to participate in the competition. Require students to apply knowledge of data structure to design a schedule for track and field competitions, so as to complete the competition in the shortest possible time.

During the implementation process, guide students to set different competition items as vertex sets in graph theory algorithms, and edge sets as "connecting an edge between items that cannot be competed at the same time", has made this non-numerical arithmetic issue easy to be solved.

5. Operate (Software Engineering)

Taking the third part of "Fundamentals of Software Technology" as an example, namely, software engineering, this practical segment corresponds to the Conceive practice (C) and Design practice (D) in the CDIO engineering education model. Conceptual practice (C) requires students to familiarize

themselves with CDIO, software engineering concepts, and the entire software project development process by consulting relevant materials in the experimental guide book, in addition, Design practice (D) requires students to complete requirement analysis and system design for software systems.

Through this practical session, it is expected that students will be proficient in programming languages and database design. We have chosen a development framework based on MFC, using SQL Server 2019 for the database, MFC interacts with data, and we have chosen ODBC (Students are also required to try DAO development methods):

(1) Database Design (SQL Server 2019).

(2) Creating an ODBC Data Source Link (ODBC).

(3) MFC Development (Visual Studio 2019 development environment).

For the hands-on practice section, we explain the experimental purpose, content, and related knowledge points in advance, students can prepare the hands-on code in advance, based on this, we can have a targeted approach in class.

For some students with poor understanding and weak foundation, the teacher provides reference programs and code, and the core statements are presented in the form of filling in the blanks, so that computer practice will not become mere formality, and each student will also master relevant knowledge to varying degrees for comprehensive application. By contrast, some optional questions are also assigned and additional points are set, providing opportunities for students with strong hands-on and comprehension abilities to integrate. For the latter, the teacher guided students to explore similar search algorithms for the data structure part (Binary Search), how to define intervals? For example, if we give an interval of 0-10, we now request to search for 11. Obviously, 11 is not within the given interval. If we execute the Binary Search algorithm step by step, it will waste memory and re-sources, therefore, we will implement it in two steps:

Step 1: Determine whether the number to be queried is within this interval, and if it is not, we will directly provide feedback to the operator, otherwise, we will search.

Step 2: If the number to be searched is within the given interval, perform a Binary Search.

In this CDIO teaching practice exploration, the students gave me feedback one by one on the difficulties they encountered, at last, a total of 12 students packaged their designed algorithms and software running results, then sent them via email. The achievement of CDIO engineering education is that students can actively analyze and solve problems, and apply what they have learned.

6. Conclusion

We cultivate computer application talents, in terms of the teaching content of this curriculum, teachers should not only comprehensively consider the key and difficult points of each part, students' acceptance ability, and the connection of subsequent courses, but also design more scenarios, small projects, and cases that are closely related to students' learning, living environment, and the forefront of society to associate with various learning objectives, therefore, summarize the teaching practice of CDIO as follows.

(1) Digging Questions from Classics. In terms of teaching settings, the starting point should be to activate students' enthusiasm and learning interest, using situational cases as carriers, combining theoretical overview with algorithm design to solve practical problems, allowing students to walk into the curriculum with questions, gradually grasp the teaching content, and ultimately apply what they have learned. Identify research projects related to teaching content, bring more and up-dated practical application materials and achievements back to the classroom, through the display and application of these materials and achievements, enhance students' practical abilities to solve practical problems and carry out software development.

(2) Assessment from Multi-perspectives. First, in static search (sequential search, binary search, block search), a related question is given, and students who design additional algorithms have corresponding added value points. Second, in dynamic search, the algorithm implementation of binary sorting tree (mandatory) requires each student to design, and the difficulty coefficient of this

experimental practice class is relatively high. Third, the sorting algorithm in the data structure section re-quires students to focus on designing the implementation of bubble sorting algorithm.

(3) Multi-channel Platform. By utilizing multimedia teaching tools, computer simulation technology, and practical programming software, profound and abstract concepts and principles are presented in vivid images and flash animations, demon-strating step-by-step program command lines of algorithms, etc., allowing students to have a more intuitive understanding, familiarity, and mastery of the correspond-ing content. For instance, for several typical operating systems, students can study the development of Windows, UNIX, and Linux operating systems voluntarily and accept them through classroom communication. In addition, for the key and difficult parts such as data structure and algorithms, a combination of flash animation and blackboard writing is adopted to master the algorithm program, and the stability, usage scenarios, time complexity, spatial complexity, etc. of several sorting algorithms are compared in the form of charts. After class, it is assisted by task driven to consolidate and strengthen these contents, and multiple teaching methods are used in a cross and comprehensive manner, which is beneficial for improving students' ability to comprehend the teaching contents.

Acknowledgements

This research was supported by Shaanxi Province "14th Five-Year Plan" Education Science Planning Project (SGH22Y1872).

References

[1] Wang, X., Cui, L., & Wang, S. (2022). Research on the Application of CDIO Engi-neering Education Model Under the Background of Big Data. In 2022 International Conference on Computer Applications Technology (CCAT) 75-79.

[2] Xu, G., Shi, L., Xiao, P., & Zhang, Z. (2021). Exploration of E-Commerce Talents Training Mode Based on CDIO-OBE Concept. European Journal of Education and Pedagogy, navigational challenges 2 (4), 20-24.

[3] Guan, X., Yan, H., Wang, Z., Gao, P., & Ding, B. (2023). Research on Teaching Reform of Artificial Intelligence Course Based on CDIO. In SHS Web of Conferences (Vol. 152, p. 03005). EDP Sciences.

[4] Yuxia, D. (2022). Java Course Teaching Reform and Research Based on CDIO Teaching Mode. International Journal of Management and Education in Human Development, 2 (02), 388-391.

[5] Yang, C. (2020). Research on Java Programming Course Based on CDIO and Iterative Engineering Teaching Pattern. Recent Advances in Computer Science and Communications, (Formerly: Recent Patents on Computer Science), 13 (3), 519-530.

[6] Jin, H. (2019). A Case Study of OBE-CDIO Implementation in Network Planning and Design Course. In 2019 14th International Conference on Computer Science & Education (ICCSE) 331-334.

[7] Aloisi, C. (2023). The Future of Standardised Assessment: Validity and Trust in Algo-rithms for Assessment and Scoring. European Journal of Education, 58 (1), 98-110.

[8] Blanco, R., Trinidad, M., Suárez-Cabal, M. J., Calderón, A., Ruiz, M., & Tuya, J. (2023). Can Gamification Help in Software Testing Education? Findings from an Empirical Study. Journal of Systems and Software, 200, 1-14.

[9] Rosier, G. (2022). The case method evaluated in terms of higher education research: A pilot study. The International Journal of Management Education, 20 (3), 100660.

[10] Truss, A., & Anderson, V. (2023). The Navigational Challenges of a Blended Learning Approach to Teaching in Business and Management. The International Journal of Management Education, 21 (1), 100733.

[11] Efthymiou, L., & Zarifis, A. (2021). Modeling students' voice for enhanced quality in online management education. The International Journal of Management Education, 19 (2), 100464.