Study on the Teaching Reform for Theory and Application of Algorithm Analysis based on OBE

Gong Mei, Zhang Haiqing
School of Software Engineering
Chengdu University of Information Technology
Chengdu, China

Abstract—Students majoring in software engineering usually have solid theoretical knowledge of computer science. However, when facing practical problems, their comprehensive ability of innovative thinking, analysis and problem solving still needs to be improved. According to the concept of OBE (Outcome Based Education), the teaching reform for the theory and application of algorithm analysis course in software engineering major is studied in this paper. The teaching quality of this course is optimized by formulating teaching objectives and contents reversely, designing diversified teaching process, constructing process-based multi-link assessment and evaluation mode, so as to improve students' learning initiative, cultivate students' innovative thinking and ability of solving practical problems.

Keywords—OBE; theory and application of algorithm analysis; reform; initiative

I. INTRODUCTION

Algorithm-related courses play an important role in computer major. These courses are listed as basic courses of computer science in many universities of China. Foreign universities such as Stanford university [1], Carnegie Mellon university [2] and MIT [3] also offer relevant algorithms courses [4]. The goal of algorithm-based courses is to train students' thinking and cultivate their ability of solving problems [1-7]. It is difficult and challenging to study this kind of course, so how to improve teaching of these courses has aroused many scholars' interest [4-7]. For the course of algorithm design and analysis, He Kejing et al. [4] proposed a comprehensive exploration of practical teaching reform. Li Shaojing et al. [5] studied the teaching method of it. Lei Peng et al. [6] and Guo Huifang et al. [7] respectively explored the teaching mode about it.

The concept of OBE (Outcome Based Education) [8] is to guide the formulation, implementation and evaluation of teaching activities via students' expected learning Outcomes [8-9]. This concept has been used in the certification of engineering education [9-10]. Algorithm courses are very practical and difficult to learn, and require students have mastered the knowledge of mathematics, data structure, and programming skills. Applying OBE model to reform the whole teaching process of algorithm courses is of great educational value, because it could cultivate students' personalized learning, innovative thinking and creative ability.

II. THE CURRENT STATE OF TEACHING

Students majoring in computer and software engineering in our school generally have good theoretical basis, but their practical ability still needs to be further improved. Many courses are equipped with corresponding experimental classes, and some engineering training classes are also added, but there still exist some problems such as limited class hours, inflexible teaching design and examination mode, more validating experiments than designing and integrated experiments, so students' interest in these courses is not high. The learning of algorithm courses requires students have mastered good data structure, programming ability of C, or C++, or Java, or Python, and relevant mathematical knowledge. The contents of these courses are closely related to practical problems, so it could arouse students' interest. However, learning such courses is difficult and challenging, therefore, how to maintain their learning interest and cultivate their ability of exploration and active learning are worth further exploration.

Some problems in traditional teaching:

(1) There are many teaching contents, relatively few experimental or practical hours, and insufficient time for students to actively think and practice.

(2) Practice contents are not rich, for example, verification topics are mostly, the design and synthesis experiments are relatively easy or less, hence, it cannot sufficiently arouse students' studying interest and raise their innovation ability.

(3) The assessment method is simple. The scores of many courses are mainly based on the final exam, and the experiment content is not high. Hence, it is not enough to fully assess students' personal ability.

III. TEACHING REFORM OF THEORY AND APPLICATION OF ALGORITHM ANALYSIS COURSE BASED ON OBE

A. Changes in Teaching Centers

Theory and application of algorithm analysis is a basic compulsory course for the software engineering major of our school. It integrates theory and practice, but it has strong practicality. The teaching of this course is mainly dominated by students and supplemented by lectures. Teachers play a guiding role, and the teaching activities are shown in Fig. 1.
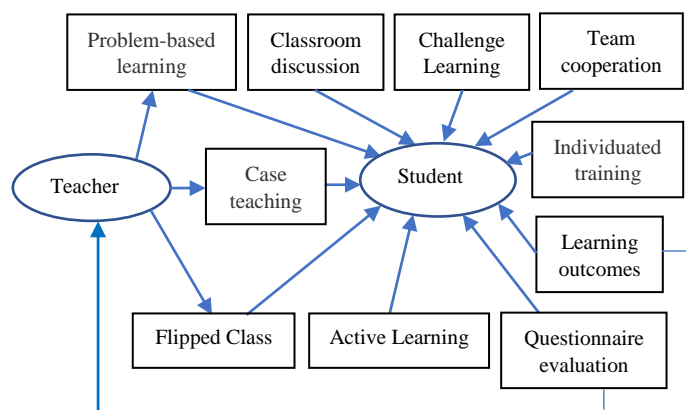
Fig. 1.   Teaching activities

Teachers guide students to learn by case teaching and expanding questions, while activities such as classroom discussion, challenge learning, questionnaire evaluation etc are all dominated by students. The learning output of students in turn influences the optimization of teaching contents by teachers.

Based on OBE and three requirements of graduation project for our algorithm analysis course, the teaching model of this coursed is shown as in Fig. 2.
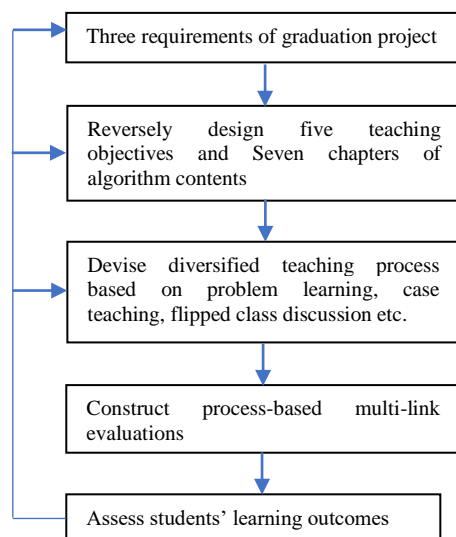

Fig. 2.   Teaching model via OBE

From Fig. 2, based on corresponding requirements of graduation project, five teaching objectives and Seven chapters of algorithm contents are firstly designed reversely, then teaching process and process-based multi-link evaluations are devised to assess students' learning outcomes, and using the learning outcomes to reversely adjust the preceding process.

B. Reversely Design Reasonable Teaching Objectives and Contents

According to students' learning outcomes, the goal of this algorithm analysis course is to help students flexibly integrate and apply data structure, mathematical knowledge and programming ability, train students' ability of analyzing algorithm complexity, understand and master the basic principle and related application of algorithm analysis, promote students' active learning, cultivate students' innovative algorithm thought, develop students' ability to analyze complex problems and design efficient algorithms, enhance student's ability of modeling and solving practical problems. These goals can support three corresponding graduation design requirements of our school, namely, the ability to build mathematical models and solve them for specific objects; able to correctly express complex software engineering problems based on computational thinking and engineering principles, able to reason, deduce and pose questions.

This course is mainly to computer practice and supplemented by teaching. The main content includes algorithm complexity analysis, the application of 6 important algorithms, such as brute force method, subtraction method etc., in practical problems, the contents of the combination of some models, such as ACM competition or online assessment, for strengthening students' ability of solving practical problems. For example, teachers can analyze algorithm complexity through case teaching of fake coin, shortest path, backpack and other problems, and require students to analyze the optimal, worst and average time complexity of algorithms when practicing. The goals are achieved through routine process assessment. Each evaluation link corresponds to a certain weight. For instance, as for the content of the greedy algorithm, through case teaching for the problems of graph coloring, job scheduling and activity arrangement, and then apply problem oriented, task driven and flipped the classroom discussion to facilitate students to design high performance algorithm and implement the codes for practical application problems such as backpack and travel business etc. Besides, the problems of bridge crossing and rumors spreading are assigned to broaden students' horizons and stimulate students to do challenging learning. When evaluating these practical contents, the pseudo-code writing and code implementation based on greedy strategy account for 0.5 weight of the experimental report respectively.

C. Devise Diversified Teaching Process

The classroom teaching of our algorithm analysis course is mainly students-centered, problems-oriented and tasks-driven to develop diversified teaching links. Based on the concept of OBE, the classroom teaching of this course is planned to be combined with case teaching, problems-oriented learning, tasks-expanding discussion and flipped classroom. For example, for the identification of fake coin, teachers will assign homework to students in advance to collect about the issue, circulation and regulations of currency through online or offline platforms. In class, firstly, some students will be asked to discuss and share how to identification various banknotes and the circulation history and process of currency. Then a problem of how to identify a fake coin among n coins with the

same shape will be raised by the teacher, and in the light of the weighing principles of scales, students will be guided to solve the problem with dichotomy. Subsequently, the students will be asked to analyze and discuss the performance of the dichotomy, to think and further discuss in groups whether there is any other method to solve this problem? If yes, how to modify the existing algorithm? When n is different, how to make sure that other algorithms are correct? After implementing each algorithm, students are required to compare the performance of each algorithm. Finally, designing and implementing algorithms about the identification of different banknotes is mainly assigned to train and improve those students with strong learning ability. By means of team study and discussion, excellent students could drive students with weak learning ability to carry out active learning.

By combining the interpretation of specific case, the inspiration of a series of relevant issues, flipped class discussion, individual initiative, team cooperation and challenging learning, the whole class is mainly students-centered and teacher-guided. Only in this way can stimulate students' learning interest, training their thinking, cultivate personalized learning and the innovative ability of solving complex problems. Besides, students are also required to participate in some training and testing of other relevant online platforms, and encouraged to attend ACM competitions, so as to improve students' ability of future employment or entrepreneurship.

D. *Construct a Process-based Multi-link Evaluation Method*

The evaluation of many courses in our school are usually 20 ~ 30 percent of regular performance and 80 ~ 70 percent of the final written examination scores. The written test cannot fully and accurately measure students' learning ability, so our algorithm course focuses on students' experimental training and supplemented by teaching. The designed training contents are closely related to practical problems, and the corresponding evaluation is changed to the routine process-based assessment. This assessment mainly includes students' participation in class accounting for 10%, algorithm performance analysis, pseudo-code writing ability, code implementation and description of algorithm design strategy accounting for 20% respectively, and English description of algorithm accounting for 10%. Class participation takes into account students' activeness in discussion, initiative and accuracy in reporting learning content, enthusiasm in learning and attendance rate, etc. Algorithm performance analysis mainly examines the analysis of space-time complexity. Pseudo-code writing ability tests students' computer thinking and design ability. Code implementation tests whether students can accurately and effectively implement the corresponding algorithm with various programming languages. English description of algorithm assesses whether students can read the English expression and write English pseudo-code. The assessment objects are mainly individuals complemented by teamwork. There is no final written examination, the evaluation depends on all the routine process-based assessment.

E. *Carry out Various Assessment*

In this course, not only the teaching process should be tracked and evaluated, but also the formative assessment on students' learning ability should be made. Therefore, two course questionnaires are designed for students' evaluation of teaching, and are used in the middle and the end of the course respectively. Students' middle evaluation is used to collect their feedbacks on the teaching process, methods and attitudes, so that teachers could timely adjust the following teaching. Students' evaluation at the end of the course is mainly used for understanding students' mastery of the course content and suggestions on the course. Base on the completion of various training contents, relevant in-class tests, extracurricular expansion exercises and some other online tests, students are assessed whether they have achieved the expected learning objectives, and the teaching links and quality of this course are optimized according to the learning outcomes and students' evaluation.

IV. CONCLUSION

OBE-based teaching reform of theory and application of algorithm analysis course is preliminarily explored in this paper. By taking students as the center and specific problems and tasks as the drive, the teaching content and quality of this course are reversely designed and adjusted, so as to better stimulate students' learning initiative, train their algorithmic thinking and ability of solving specific problems, and lay a foundation for their better employment or entrepreneurship in the future. In this course, while teaching is complementary, but the teacher is a guide to help and stimulate students to study, and should optimize and continuously improve the course, therefore, teachers should constantly strengthen the learning of new ideas and knowledge, and are encouraged to exercise themselves and increase the actual project development experience in enterprises, so that they could dynamically adjust some teaching contents and link for more effective teaching.

REFERENCES

[1] Stanford University. CS 161-Design and Analysis of Algorithms. [2019-08-20]. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms /.

[2] Carnegie Mellon University: Algorithms and Advanced Data Structures [Fall 2019]. [2019-08-20]. https://enr-apps.as.cmu.edu/open/SOC/SOCServlet/search.

[3] Massachusetts Institute of Technology. 6.046/18.410 Design and Analysis of Algorithms. [2019-08-20]. https://stellar.mit.edu/S/course/6/sp16/6.046/index.html.

[4] He Kejing，Zhang Xingming, and Zheng Yunping, "A comprehensive exploration of practical teaching reform of algorithm design and analysis course," Computer Education, vol. 2, pp. 45-49, February 2017. (In Chinese)

[5] Li Shaojing, Xu Haiyang, and Lv Jianbo, "Research on the teaching method of algorithm analysis and design in the training of applied talents", SCIENCE & TECHNOLOGY INFORMATION, No. 4, April 2017. (In Chinese)

[6] Lei Peng, Zhang Xiaofeng, and Wang Jinglian, "Reform of Teaching Mode Based on Training Innovative Ability: Taking Design and Analysis of Algorithm Course as an Example," vol. 305, pp. 97-99, January 2019. (In Chinese)

[7]   Guo Huifang, and Liu hanbing, "Exploration of research-and-discussion teaching of algorithm design and analysis course," Computer Education, vol. 4: pp. 60-62, April 2018. (In Chinese)

[8]   Spady William G., "Outcome-Based Education: Critical Issues and Answers," Arlington Virginia: American Association of School Administrators, 1-212, 1994.

[9]   Cao Ying han, Yang Long, Ye Hongyong, Li Rulin, and Du Chaojun, "Practical Teaching Reform based on OBE Educational Idea," Jiangxi Chemical Industry, vol. 5, pp. 213-214, October 2018. (In Chinese)

[10] Shao Xiaoyan, Zheng Xiaodong, and Sun Jianhua, "Research on teaching reform of Python based on OBE concept," Journal of Henan Mechanical and Electrical Engineering College, vol. 26, pp. 66-68, July 2018. (In Chinese)