K-007

REFINING PROGRAMMING COURSE PLAN WITH LEARNER BEHAVIOR DATA

Dinh Thi Dong Phuong　Fumiko Harada　Hiromitsu Shimakawa

1. INTRODUCTION

The introductory programming course itself consists of variety of knowledge that the learners take the course are required to acquire. The knowledge is usually arranged in a successive manner, too. If a student cannot master an issue well in a certain time, she may encounter difficulties later. Once the difficulty gets more and harder, it may cause the student depressed and drop out of the class [1].

The paper proposes a method to accomplish a successful programming course design. The key point is the refinement of a course plan (CP) over practical courses with the student opinions reflecting their real programming experiences and genuine behavior data presenting their true learning progress.

To achieve the CP, we get opinions of previous learners of the course by contextual inquiry method. We establish services for education service categories: objective, teaching, assessment and motivation taking the student opinions important. We integrate these services into the CP considering the level of understanding of the learners together with the advantages and disadvantages of every service. We apply the CP into actual programming courses and take log of student learning progress. We evaluate the CP to improve it. Applying the method over two programming courses in Ritsumeikan University, with nearly 500 students in each course, we have accomplished a CP that 80.05% of about 500 students with full score in the mid-term test.

2.WEAKNESSES OF PROGRAMMING COURSE

In order to master the knowledge in the introductory programming course, the learners need to keep spending time and making effort throughout the course. Since the knowledge is successive, if they have gaps in their understanding and the gaps are not filled in a timely fashion, the students would encounter successive difficulties that cause them depressed. Moreover, the level of understanding of a student may change depending on the knowledge category. She may master an issue well while she cannot understand another.

For the time being, there is no methodology to obtain a successful programming course that motivates all learners throughout the course. Educators often determine one primary strategy such as lab practice [2], game programming [3], pair programming [4] and so on [5] to them. The strategy motivates one set of the learners only in a specific period of the course. Many learners who cannot adapt to the course design may not be motivated to keep taking efforts on the learning. The more the course proceeds, the harder it is for them to deal with the course. Once they cannot overcome the problems, they would drop out of the course.

3. REFINING COURSE PLAN WITH GENUINE BEHAVIOR DATA

To achieve a course that motivates all learners to make efforts on the learning, we should design the course so that it can recommend proper strategies to a level of understanding. A course plan is a general drawing describing how the course will be carried out. The plan is an integration of many education services. Each service is a means to contribute to the learning. In order to accomplish a good CP, we refine the CP over actual programming courses with genuine data. Figure 1 represents five steps to achieve the CP.

First, we use contextual inquiry [7] on students who have finished a programming course to get information reflecting the learners themselves as well as the course that they have just taken. The information tells us their learning styles, what are their difficulties and the way they solve them. It also reflects their interests and desires in programming field. Moreover, the information is somewhat real assessment of the course. When we design the CP, the information plays a significant role as if customer requirements in software development. We will use this information in establishing education services.

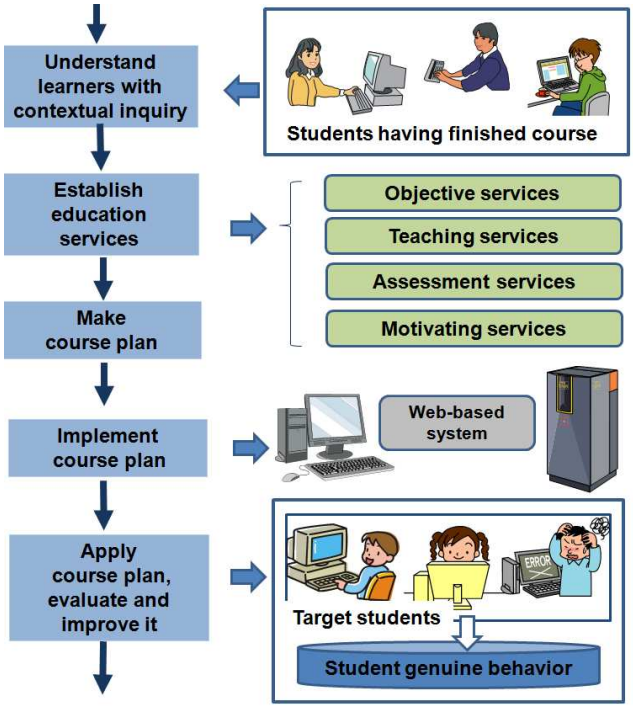

Figure 1. Overview of achieving a course plan

Second, we establish education services. They are classified into four categories: objective, teaching, assessment and motivating. Objective category includes services presenting the objectives of the course. Usually, the objectives of the whole course are organized into the objectives of each chapter in a successive fashion. For introductory programming course, the objectives are often represented as assignments in every week with the increasing requirements during a semester lasting 15 weeks. The assignments of a week represent knowledge of a chapter that we expect learners to acquire within the week.

Teaching category consists of services transferring the objectives to the learners. The objectives of introductory programming course are the student programming ability, which consists of mainly skills to make computer programs by applying many kinds of skills and knowledge. Besides the lesson class in which a teacher mainly demonstrates coding with a specific programming language, other classes are indispensible. One example is lessons to explain knowledge of computer concepts.

Another is practice classes where teachers and teaching assistants, or supervisors in short, transfer skills to make a successful program. It also emphasizes that the practice classes is the place that brings about opportunities to practical programming experience for students. In the places, the students themselves synthesize the mastered knowledge to develop their own skills. In other words, the practice classes are the real place where the learning and skill training take place. Therefore, to make the learning and skill training reach to high quality, we should prepare a team of supervisors sufficient in quantity and quality. They will take charge of giving guidance to individuals when they question or when they remain in difficulties.

Assessment category describes services to assess, or examine the learner understanding. Assessment is to inform the learners how well their learning proceeds. It helps them adjust and improve the learning to a better way. The more the assessment are in frequency, quality and timeliness, the better they contribute to the learning [6]. Various knowledge and skill are needed for students to be able to make a program. The knowledge involves includes the understanding basic computer concepts such as a path, a file name, an operating system, and a memory. It includes the knowledge about programming language syntax and using library functions as well as data types. It contains experiences of programming techniques such as editing, compiling and debugging. It sometimes means skills on how to design a program and make an easy-to-read one. Therefore, portfolio assessment [9] that examines the understanding of each knowledge point would be significant. In parallel, we should give confirmation of proper behavior of the source codes to them. Performance assessment [9] is also necessary.

Motivating category covers all services that encourage and motivate learners to make effort on the learning. The information representing the student interests and desires we have known through the contextual inquiry method plays important roles in guiding us to consider motivating services. Since motivation to learn programming varies with learners, if we can classify the learners into groups based on their motivating factors, we can direct services to each specific group when we design the CP [8].

Along with defining services needed for each category, we investigate their advantages and disadvantages. The information is significant reference to integrate the services into a CP.

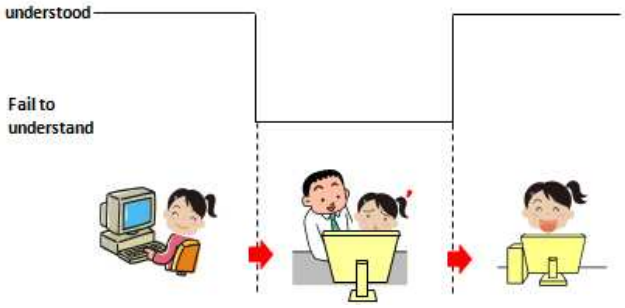

Figure 2. The CP recommends different services to a student based on her level of understanding.

Third, we make the CP by selecting the service of the four categories to integrate into a unified structure properly. When a learner cannot understand knowledge of a chapter, the CP should provide appropriate services to help her fill in the understanding gaps in time, considering the advantages and disadvantages of each service. When she has mastered the knowledge of a chapter, the CP should give her services to encourage her try more challenging assignments. Figure 2 illustrates two levels of understanding of content of a chapter. When she understood the content, she is encouraged to work alone. When she fails to understand it, interactive check, the service at that a supervisor comes to her to give face-to-face instructions, is recommended.

The next step is implementing the CP into a web-based system to acquire data that precisely represent the learning progress of learners throughout the course. The system records actions with time, place, and references to relating materials to solve an assignment. The logs are for analyzing effects of the CP to the learners. The system also facilitates transferring feedbacks of assessment quickly to the learners. It supports the learning in any places at any times to encourage different learning styles of students, especially practicing programming at home, too.

The last step is applying and evaluating the CP. The CP that is the integration of selected services should bring benefits to all involving parties: learners, teacher and TAs. During the execution of the CP, we address the real advantages, disadvantages of the services. We discover the articulating capability of a service with others. This real information makes services practical. It enables us to improve later versions of the CP. After the course finishes, we examine the user genuine log data to know how the CP works under many aspects. For example, we measure the time which every student has spent on the learning in every week to understand how much the learners keep taking effort on the learning. Another is evaluation of improvement on programming ability by inspecting score of the assignment. Besides the evaluation regarding to the students, we can calculate accurately the teacher cost to prepare the services. We can figure out the cost of TAs providing learning services and assessment services.

With practical evaluation of the CP, we can enhance it over courses to assure the course quality.

4. CONCLUSION

The paper proposed a method to accomplish a high quality programming course design to motivate all of learners and assure the course objectives. The application on two programming courses with nearly 1000 students that it is a successful approach.

REFERENCES

[1] Nghi Truong, Paul Roe and Peter Bancroft, 2005, Automated Feedback for “Fill in the Gap” Programming Exercises, January 2005, *ACE '05: Proceedings of the 7th Australasian conference on Computing education* - Volume 42 , pages 117-126.

[2] Kai Stapel, Daniel Lübke, Eric Knauss, 2008, Best Practices in eXtreme Programming Course Design, *ICSE’08*, May 10–18, 2008, Leipzig, Germany.

[3] Michael Kölling, Poul Henriksen, 2005, Game Programming in Introductory Courses With Direct State Manipulation, *ITiCSE’05*, June 27–29, 2005, Monte de Caparica, Portugal.

[4] Laurie Williams, Kai Yang, Eric Wiebe, Miriam Ferzli, Carol Miller, Pair Programming in an Introductory Computer Science Course: Initial Results and Recommendations, *Proceedings of the AAAI Workshop on Approximation and Abstraction of Computational Theories*.

[5] Leo F. Denton, Dawn McKinney, and Michael V. Doran, 2005, A Melding of Educational Strategies to Enhance the Introductory Programming Course, October 19 – 22, 2005, Indianapolis, *35th ASEE/IEEE Frontiers in Education Conference*, Paper F4G.

[6] Anne Venables and Liz Haywood, 2003, Programming students NEED instant feedback!, *ACE '03: Proceedings of the fifth Australasian conference on Computing education*, Volume 20, January 2003.

[7] Hugh Beyer,Karen Holtzblatt,1998, *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann Publishers, Inc. San Francisco, California.

[8] Robert E Salvin, 2012, *Educational psychology: theory and practice,* Pearson.

[9] Catherine A. Palomba, Trudy W. Banta, *“Assessment Essentials: Planning, Implementing and Improving Assessment in Higher Education"*, Jossey-Bass, A Wiley Imprint, 989 Market Street, San Francisco, CA 94103-1741, 1999.