T. Hirashima et al. (Eds.) (2011). Proceedings of the 19th International Conference on Computers in Education. Chiang Mai, Thailand: Asia-Pacific Society for Computers in Education

Algorithm Learning Environment for Linked List with Programmers' Perspective

Yasuhiro NOGUCHI [a*], Satoru KOGURE [a], Tatsuhiro KONISHI [a] & Yukihiro ITOH [b]
[a]*Faculty of Informatics, Shizuoka University, Japan*
[b]*Shizuoka University, Japan*
*noguchi@inf.shizuoka.ac.jp

Abstract: We have constructed a learning environment in which algorithm/programming learners can operate objects on a linked list model in its domain world. The main advantage of our system is that our model reflects an expert programmers' perspective: some concepts of programming, and some implicit restrictions caused by those algorithms are written for computers not for people. Such a perspective is important for understanding algorithms; however, it is not common that this perspective is taught to learners explicitly. In this paper, we proposed such a model for a linked list as an example of dynamic data structure. An evaluation of the system and model with the programmers' perspective showed a marginally positive effect for understanding algorithms.

Keywords: Programming/Algorithm learning, Interactive learning environment, Programmers' perspective

Introduction

In recent years, there is an increasing importance of education for algorithms as a solution for not only computer engineers, but also the general population [1]. However, in the traditional educational method for programming/algorithms, learners must study programming languages and algorithms at the same time. It is difficult to develop learners' algorithm thinking skills, because their programming language skills are insufficient [2]. To resolve such problems, some learning environments have been proposed in which learners use an algorithm description language (e.g. flow-chart, PAD and so on) instead of a programming language for building and executing an algorithm. We think that learners who can write algorithms using an algorithm description language already have sufficient skills to build and execute algorithms without using such learning environments.

It has been reported that a hands-on learning method is effective for beginners [3]. Introducing this method to programming education, hand held educational materials were developed. We believed that tracing algorithm execution processes using a hands-on learning method is effective for beginners, and it is important that learners can be given rich feedback which shows if operations externalized by the learners are correct or not, and why not. In our previous research, we constructed a learning environment which supports some type of sorting algorithms for an array model in its domain world. In this learning environment, learners can operate objects with specific values on a data structure model to trace an algorithm, and the learning environment can give learners feedback which shows if learners' operations are correct or not [4]. In this paper, we extend our previous system to support some algorithms for a linked list as a dynamic data structure.

1. Linked List Model with Programmers' Perspective

To understand an algorithm, learners need to trace it with some rules of programming and some restrictions. In other words, learners cannot trace an algorithm execution process correctly without rules and restrictions because an algorithm is written for computers not for people. For instance, a program cannot directly access a linked object (LO) in the linked list as non-programmers want it to. The structure of a linked list in Fig. 1 allows learners to directly access the LO whose value is 6 via reference chains between LOs from pointer "P" in only one step, if learners know where is the LO. However in reality, the program should firstly get a reference to the LO before accessing the LO; the program needs to execute a process in which the reference of pointer "P" is shifted to the next LO until reaching the LO whose value is 6. This is because that data and processes in the algorithm must be abstracted to make computers execute it. We believe that expert programmers already have this perspective and they can explain an algorithm execution process correctly in a typical model of a linked list as in Fig. 1.

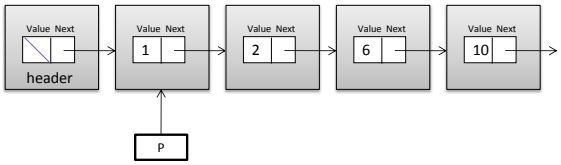

Fig. 1 Typical model of linked list

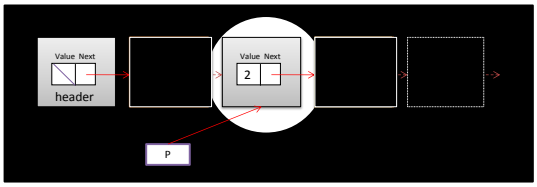

Fig. 2 Model of linked list with highlighting function

However beginners cannot easily explain this in the same way. They often operate objects which cannot be used at a specific step, or do operations which don't need to be done. For instance, learners operate a LO which is not referenced. Then, learners often cannot recognize such mistakes, then they make adjustments to achieve their desired outcome by closed processes which computers cannot do. Therefore, we believe that it is important for beginners to learn such programmers' perspective for understanding algorithms. Although it is important for beginners to learn such a perspective for understanding algorithms, it is not common to have such perspective taught explicitly. In many cases, learners are expected to implicitly get such perspective themselves by tracing an algorithm and reading its sample code. However, it is difficult for beginners. We believe that a learning environment should support the following in order for learners to experience such a perspective.

1. A domain world must present a model of a linked list to study. In the domain world, the GUI visually shows which resources learners can access: LOs, pointers, values, and so on.
2. Learners can operate models in a domain world via GUI. Then, learners can reproduce steps of an algorithm execution process for a linked list.

Fig. 2 shows our model for a linked list with the highlighting function supported by our proposed system. In this model, LOs, pointers and values which learners can access at a specific step are highlighted. In contrast, LOs, pointers, and values which learners should not access at each step are out of sight. Using this learning environment, learners can be consciously aware of the algorithm execution process for a linked list. Therefore, learners can focus on what/how they must do at each step.

Fig. 3 shows our educational system in which learners can create a new LO, compare each data between two LOs, change a reference of a pointer and release a LO on the linked list model. In our model, the highlighting function highlights only pointers, values, and LOs which are directly accessible by a pointer; the highlighting function shows referable resources by a pointer in distinction from highlighted resources (Fig. 4 (E)). Operations

which change reference to an un-highlighted resource and/or break the structure of the linked list lead to a popup feedback message (Fig. 4 (B)).

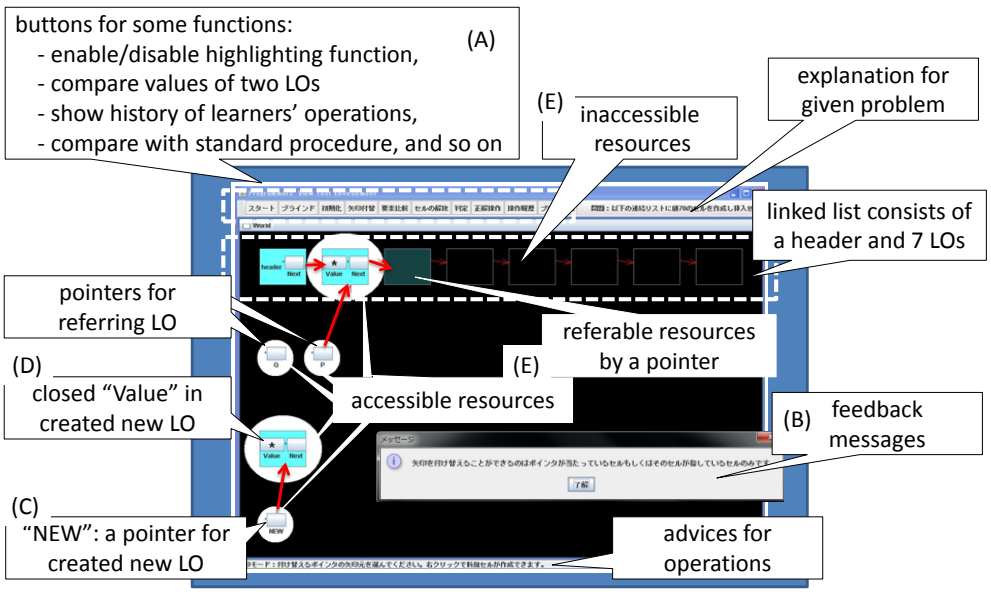

Fig. 3 Main window of our system

2. Conclusion

In this paper, we built an educational system based on our discussed requirements of models using a programmers' perspective for a linked list algorithm learning environment. Our preliminary evaluation with eleven college students showed these results.

1. In the learning environment, models in programmers' perspective may have a positive effect for algorithm understanding by making learners focus on whether they can access LOs, values and pointers, and which operations learners should do in each step.
2. It is difficult for beginners to understand source code by reading only a sample program. Source code reading with the learning environment in which learners operate models appears to be helpful for such learners.

Acknowledgements

This work was supported by Japanese Grant-in-Aid for Scientific Research (B) 20300267.

References

[1] Harada, E. (2000). *Informatics and Computer Science Education Non-Computer Science Undergraduate students: Objectives and Issues (Special Features Informatics and Computer Science Education for Non-Computer Science Undergraduate Students) [in Japanese]*. Journal of Information Processing Society of Japan, 41(3), 227-233. (in Japanese)

[2] Fuwa, Y., Kunimura, H., Kayama, M., Niimura, M. & Miyao, H. (2009). *Proposal and Practice of an Educational Method for Developing Algorithmic Thinking for Students in Computer Science*. JSiSE Research Report, 23(6), 34-41. (in Japanese)

[3] Holmes, G. & Smith, T.C. (1997). *Adding some spice to CS1 curricula*. ACM SIGCSE Bulletin, 29(1), 204-208.

[4] Noguchi, Y., Nakahara, T., Kogure, S., Konishi, T., & Itoh, Y. (2010). *Construction of a Learning Environment for Algorithm and Programming where Learners Operate Objects in a Domain World*. International Journal of Knowledge and Web Intelligence, 1(3/4), 273-288.