Adapting Guidance and Externalization Support Features to Program and Algorithm Learning Support Environment

Satoru KOGURE [a*], Makoto OKAMOTO[b], Yasuhiro NOGUCHI[a], Tatsuhiro KONISHI[a], Yukihiro ITOH[c]
[a]*Faculty of Informatics, Shizuoka University, Japan*
[b]*Graduate School of Informatics, Shizuoka University, Japan*
[c]*Shizuoka University, Japan*
*kogure@inf.shizuoka.ac.jp

Abstract: In this paper, we proposed novel learning support environment for understanding programs and algorithm by emphasizing three worlds: programs world, domain world and operations world. To lead learners deeply understand programs and algorithm, we suggest "guidance feature" and "externalization support feature" in learning environment. Hereby, we guess that learners could bridge their gap between programs world and operations world by these features.

Keywords: learning support environment, program and algorithm learning, externalization

Introduction

In order to learn programs and algorithms on elementary programming education, the learners perform three typical learning procedures: understanding algorithm, understanding behavior of sample program, and writing code [1][2][3][4][5][6].

In our study, we focus on "understanding algorithm" and "understanding behavior of sample program". Then, we propose three worlds which learners should consider with learning programs and algorithm. These worlds are "programs world" (including abstract statements in program code), "target model world" (including concrete representation of the target problem) and "operations world" (including concrete operation sequences for the exact input data).

We believe that it is necessary for learners to understand the relationship between three worlds in order to understand a program and algorithm implemented in the program. There is our previous study[6] that emphasizes the relationship between three worlds. In the research, we categorized three programming levels and construct the programming learning support environment which has some suitable functions for each level's learners. Then we evaluated the environment in code reading. Based on the policy of this research, we construct extended programming learning support environment which has two novel features: **"guidance feature"** (to lead learners to perform suitable process at any time) and **"externalization support feature"** (to lead learners to abstract concrete operation sequences to a description of general procedure).

1. Learning Support Policy

It is relatively easy for programmers to understand the relationship between the domain world and operations world because the both of them are composed by concrete data However, it is difficult for learners to understand a difference between programs world and other two worlds. In program world, concrete data is abstracted, and operations with concrete data are structuralized.

We focus learners that are able to reproduce the operation sequences for concrete data (L1). We design our system to lead learners to be able to generalize operation sequences (L2), and to be able to understand the relation of programs world and two other world (L3) from (L1) level. We present programming learning support environment for learners that reached L1 level but does not reach L2 or L3 levels. Our proposed support environment has two novel features, "guidance feature" and "externalization support feature". "Guidance feature" is function that leads learners to perform a suitable process on the learning environment according to learner's understanding state. In order to realize "guidance feature", our proposed learning environment support to indicate guidance to learners based on learning scenario. "Externalization support feature" shows learners some candidates of the description externalizing some operation sequences, and leads learners to judge a correctness of the description by choosing correct one from some candidates. By these two features, learners deeply understand the relationship between program codes and operation sequences.

2. System Outline

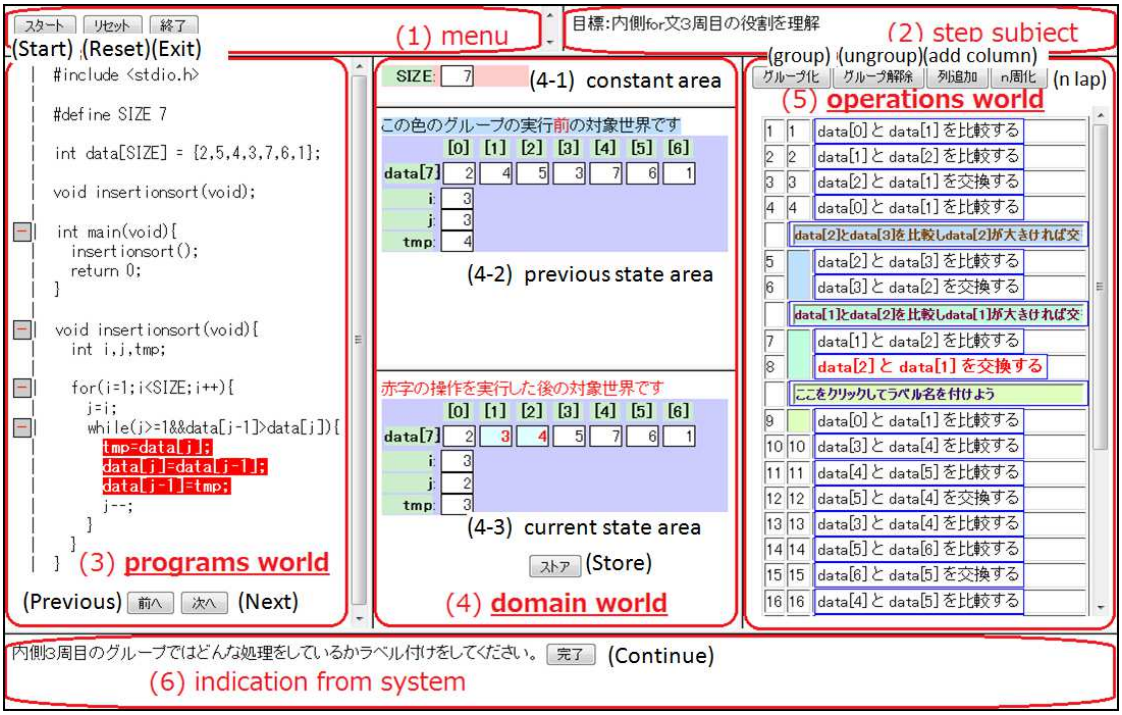

Figure 1. Interface overview.

Figure 1 shows the interface of our proposed learning support environment. Figure 1 (3), (4) and (5) show three worlds that we guess very important concept for programming and algorithm learning. Figure 1 (1) shows a menu of support environment, Figure 1 (2) shows a step subject at a learning process, and Figure 1 (6) shows the indication the system give to learners for suitable learning process according to learner's understanding state.

The system has nine functions for supporting learners. In a previous study[6], we implemented six functions: executing step by step (Figure 1 (3)), packing any block (Figure 1 (3)), storing each state of target world (Figure 1 (4)), making a group of operation sequences (Figure 1 (5)), writing a label made group of operation sequences (Figure 1 (5)), and displaying behavior of grouped operation sequences (Figure 1 (4-2) and (4-3)).

In this study, we proposed new three functions to realize "guidance features" and "externalization support feature". New three functions are **function of indicating guidance (Figure 1 (6)), function of judging correctness of written label, and function of externalizing n laps.**

In pre-experimental results [6], we inscribed that learner wants suitable guidance for learners understanding level in learning programs and algorithm to use learning support environment. The system can lead learners to learn along a prepared scenario by a teacher. The system can give each learner appropriate instructions for each learner's understanding state according to the scenario (**function of indicating guidance**). In order to judges a correctness of learners labeled description, the system can lead learners select the most similar alternative in content to description that learners label in operations world (**function of judging correctness of written label**). In order to bridge a gap between program world (abstract world) and operations world (concrete world), the system lead learners to increase abstract level of operations world using **function of externalizing n laps**. Due to limitations of space, a detailed discussion of these functions is not possible here.

3. Conclusion

We proposed novel learning support environment for understanding program and algorithm. We emphasized the three worlds: programs world, domain world and operations world for understanding program and algorithm. So, we implemented nine functions. Using this support environment, learners will grow understanding and become L3 understanding level. In the near future, we bring implementation of our proposed learning support environment to completion and evaluate it.

Acknowledgements

This study was supported by Japanese Grant-in-Aid for Scientific Research (B) 24300282.

References

[1] Butz, C.J., Hua, S., & Maguire, R.B., (2006). A web-based bayesian intelligent tutoring system for computer programming, *Journal of Web Intelligence and Agent Systems*, 4(1), 77-97.

[2] Fossati, D., Eugenio, B. D., Brown, C., & Ohlsson, S. (2008). Learning linked lists: experiments with the iList system, *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, 80-89.

[3] Gabor, T., (2009). Algorithm visualization in programming education, . *Journal of Applied Multimedia*. 4(3), 68-80.

[4] Nakahara, T., Konishi, T., Kogure, S., Noguchi, Y., & Itoh, Y. (2009). Learning environment for algorithm and programming where learners operate objects in a domain world using GUI. *Proceedings of the 17th International Conference on Computers in Education*, 59-66.

[5] 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.

[6] Satoru, K., Makoto, O., Yasuhiro, N., Tatuhiro, K., & Yukihiro, I. (2011). Programs and algorithm learning environment by visualizing relations among program codes, operations and world model, *Proceedings of 19th International Conference on Computers in Education*, 302-306.