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

Programs and Algorithm Learning Environment by Visualizing Relations among Program Codes, Operations and World Model

Satoru Kogure[a*], Makoto Okamoto[b], Yasuhiro Noguchi[a], Tatuhiro 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: We propose a method to construct an educational system that supports learners to relate their understanding on concrete operations to both generalized algorithm and program codes. The system has functions visualizing relations among program codes, a sequence of operations achieving the purpose of the target algorithm, and status of the domain world. We implement the system and perform preliminary experimental evaluation. The result suggests that our system has some educational effects as we expect.

Keywords: Programming education, Algorithm learning, Interactive learning environment

Introduction

There is a traditional learning method for novice learners of algorithm such that a learner is given some teaching materials (such as playing cards) as a metaphor of data or variables, then he/she operates the materials to reproduce the target algorithm he/she should learn. The learners try to generalize the procedure performed by them to understand algorithm not in concrete level, but in abstract level. In the previous research [1], we have developed an educational system for algorithm based on a learning method which is a kind of extension of the traditional method. The system visualizes the domain world of the target algorithm (ex. Numbered cards arranged horizontally are displayed for sorting problems.) in GUI. It has a user interface in which a learner can operate the domain world by direct manipulation. The learner tries to reproduce the target algorithm using the interface. The system is better than traditional educational materials because it can evaluate validity of learner's operations by comparing them with a standard sequence of operations that it generates from source code representing the target algorithm.

However, both our previous system and traditional educational materials can only support learners to design a sequence of operations on concrete data. They don't have effective functions to support learners generalize the operations into abstract level. Therefore, many learners tend to reach an impasse that they can't understand the relation between operations on concrete data and generalized algorithm or program codes.

In this paper, we propose a method to construct an educational system that supports learners to relate their understanding on concrete operations to both generalized algorithm and program codes. Our basic approach is as follows: The system has some functions visualizing relations among program codes, a sequence of operations achieving the purpose of the target algorithm, and status of the domain world. Learners can execute any steps of the program code and the operations as they like, and observe how the execution causes changes in the domain world in order to find a general meaning of the executed program or operations. For supporting such learning, the system has some more functions: Our GUI makes it easier for learners to compare the statuses before and after the execution of operations or statements in the program. Another function is an editor for externalization of meaning of operations that the learners find. We implement the system and perform preliminary experimental evaluation. The result suggests that our system has some educational effects as we expect.

1. Learning Support Policy

Learning programming and algorithms in the scene, there are three important worlds. They are sequence of statements (program world), model of data structure on source code (domain world) and sequence of operations on specific data (operation world). To learn algorithm and program, we believe that learners should grasp the relationship among these three worlds. However, there are two major barriers. First, it is difficult for learners to grasp the relationship between each world because the program has a generalized representation but domain world and operation sequence have a concrete representation. Second, it is difficult for learners to grasp the correspondence between each world.

We classified learners according to the degree of understanding.

(L1) A learner does not quite understand the algorithm.

(L2) Although a learner can almost understand the algorithm, he/she cannot reproduce the algorithm behavior for exact data on the domain world.

(L3) A learner is able to reproduce the algorithm behavior for exact data on the domain world, but he/she does not understand the relationship between the domain world and the program world or between the operation world and the program world.

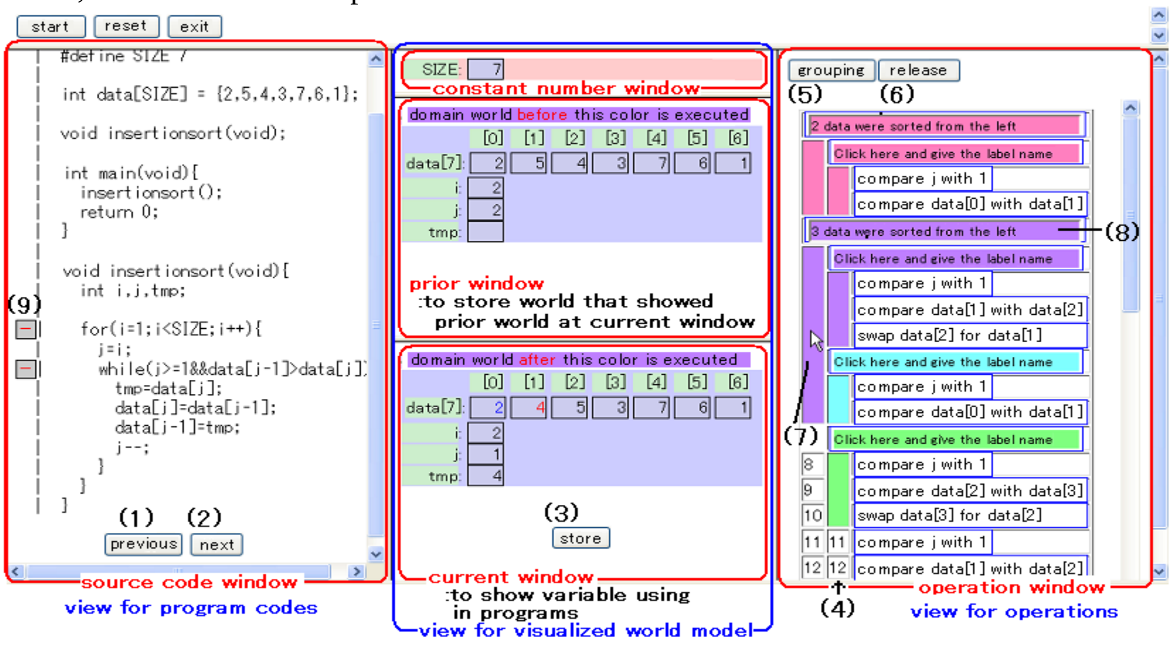
(L4) A learner understands the relationship among the three worlds, and he/she is able to understand the relationship between operation world and program world.

Usual lectures of programming and algorithm try to improve the understanding level learners from (L1) to (L4) directly. We believe that these learning are more effective by doing in stages. In general, the teacher lectures using the textbook for changing comprehension of a learner from (L1) to (L2). Using visual simulator [1, 2, 3, 4] is effective for a learner in order to improve from (L1) or (L2) to (L3) of a learner's comprehension. The purpose of our study is to create a programs and algorithms learning environment that can improve understanding of learners from (L3) (or middle of (L2) and (L3)) to (L4).

We describe related researches. First, there are learning methods with non-electronic materials. Learners learn the algorithms by reading a textbook that includes the source code and explanation on behavior of the algorithms. Learners can understand the relationship among three worlds by simulating the change of them in his/her mind. This learning method has the disadvantage that it is hard to trace the behavior of each world and to reproduce the change of each world accurately. Second, there are learning methods with electronic materials. Debugger or tracer is an example. These tools can accurately reproduce the domain and operation world. However, learners cannot learn by comparing the domain worlds before/after executing operations. Study to promote understanding of programs and algorithms by a visual representation of the domain world is also active [1, 2, 3, 4]. These studies support to show learners the relationship among program, domain and operation world. However, these studies do not support to show learners the relation among program world, domain world and operation world of which operations are abstracted by learners.

2. System Overview

We designed the interface shown in Fig.1. The interface includes three views for program world, domain world and operation world.



```
#define SIZE 7

int data[SIZE] = {2,5,4,3,7,6,1};

void insertionsort(void);

int main(void){
  insertionsort();
  return 0;
}

void insertionsort(void){
  int i,j,tmp;

  for(i=1;i<SIZE;i++){
    j=i;
    while(j>=1&&data[j-1]>data[j]){
      tmp=data[j];
      data[j]=data[j-1];
      data[j-1]=tmp;
      j--;
    }
  }
}
```

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
data[7]:	2	5	4	3	7	6	1
i:	2						
j:	2						
tmp:							

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
data[7]:	2	4	5	3	7	6	1
i:	2						
j:	1						
tmp:	4						

```
2 data were sorted from the left
  Click here and give the label name
    compare j with 1
    compare data[0] with data[1]
3 data were sorted from the left
  Click here and give the label name
    compare j with 1
    compare data[1] with data[2]
    swap data[2] for data[1]
  Click here and give the label name
    compare j with 1
    compare data[0] with data[1]
  Click here and give the label name
8     compare j with 1
9     compare data[2] with data[3]
10    swap data[3] for data[2]
11 11 compare j with 1
12 12 compare data[1] with data[2]
```

Figure.1 Designed interface that includes three views.

To compare the domain worlds before/after executing operations, we separated the view of domain world into *current/prior windows* (for showing variable) and constant number window.

Our proposed system has three functions. Each function are as follows:

<u>(Function.1) functions that replicate statement execution history and show correspondence</u>

The repetition function is effective to display the relationship among three worlds together. If a learner pushes next button (Fig.1 (2)), the system shows next statement execution history at *current window*. Then the system shows emphatically the corresponding statement at program window and shows emphatically the corresponding operation at operation window. In the same way, a learner can use previous button (Fig.1 (1)). The comparison of *current window* with *prior window* is also beneficial to understand programs and algorithm. If a learner pushes store button (Fig.1 (3)), the system copies current world model at *current window* onto *prior window*. After pushing next button on several times, a learner can compare both worlds.

<u>(Function.2) functions that group a sequence of operations and label the sequence</u>

The function that abstracts operations is needed for algorithm understanding. To group operations, a learner pushes number in front of each operation (Fig.1 (4)) in order to dictate the region of the grouping operation. Then a learner pushes grouping button (Fig.1 (5)). A learner pushes grouped operations (Fig.1 (7)), so the system shows current world to *current window* after executing grouped operations and prior world to *prior window* before executing grouped operations. A learner improves understanding the behavior of grouped operations due to compare both worlds. The group of operations is labeled as the suitable name considered by a learner if he/she pushes the group label of operations (Fig.1 (8)).

(Function.3) functions that wrap/unwrap a block

It is effective for understanding programs to reproduce program execution process by statement and block. If a learner pushes minus symbol on the left side of block ("for", "while" or "if") (Fig.1 (9)), the system hides the brackets and those contents block. The system reproduces execution process of wrapped block by block. And the system reproduces execution process of unwrapped block by statement.

We describe a learning procedure with our system. A teacher sets target algorithm and programs. A learner assigns initial values to variables. The system embeds initial values in target programs and generates html file for program, domain and operation worlds. A learner uses following two ways to learn programs and algorithm.

1. understanding relationship among program, domain and operation worlds by pushing next/previous button to change status of the program world

A learner observes what to do in a statement (using function 1) and/or a block (using function 1 and 3).

2. understanding relationship among program, domain and operation worlds by pushing operation or grouped operations label to change status of the operation world

On *current/prior window*, a learner observes a difference of domain world's status between freely selected operations of a region the learner considers as a group (using function 1). If a learner finds a certain meanings of the region, he/she groups the region. And the group is labeled based on the meanings (using function 2). A learner improves understanding programs and algorithms in order to use these two ways jointly.

3. Preliminary Experimental Evaluation

The aim of our experiment is to confirm whether our propose methods have effectiveness for learning programs and algorithms.

We collected four subjects who are university students and who have six months of experience on novice programming. As target algorithms, we select insertion sort and selection sort. The procedure of our experiment is as follows:

First, we perform a pre-exam. We divide four subjects into group A and B based on score of the pre-exam. The pre-exam is code reading of a two dimension array algorithm which includes nested loop. Second, we explain two target algorithms to subjects. And we show procedures of these algorithms in specific values using playing cards.

Third, group A learns selection sort algorithm using our system for fifteen minutes. Group B learns selection sort algorithm for fifteen minutes, using schoolbook which includes an array model and typical initial data. And we give group B work sheets. We perform post exam for selection sort. Forth, group A learns selection sort algorithm using schoolbook, and group B learn selection sort algorithm using our system. We perform post exam for insertion sort. Finally, we have a questionnaire for investigating the learning effect of our proposed system.

The post-exam includes four questions as follows:

(Q1) Describe a role of the external loop in the program code.
(Q2) Describe a role of the internal loop in the program code.
(Q3) Find statements from the program code corresponding to operations of swapping.
(Q4) Find values of some variables at a certain step of statement execution process.

We grade the answers on zero-to-ten scale. We prepared Q1, Q2 and Q3 to investigate the subjects' understanding level on the relation between algorithms and program codes. Q1 and Q2 are more difficult than Q3 because they request learners to understand meanings of loops. Then, we prepared Q4 to investigate the subjects' ability of code tracing. We predict that the differences between each group's score are ranked in descending order of Q1/Q2, Q3 and Q4. Table 2 shows the results of post-exam.

Table 2. Results of post-exam.

	insertion sort		selection sort	
	schoolbook	system	schoolbook	system
Q1	0.0	5.0	0.0	5.0
Q2	0.0	10.0	5.0	5.0
Q3	10.0	10.0	5.0	10.0
Q4	7.5	6.5	6.5	6.5

The results show our prediction is valid. According to this result, we think that our system has a certain educational effect which improves learners to expected learners' level. However the evidence is relatively weak because subjects are only 4 people.

In the result of questionnaire, subjects grade our system according to the following points on one-to-ten scale.
(Q5) Was the usability of each learning material (schoolbook or system) good?
(Q6) Did each of the learning material (schoolbook or system) improve the learning efficiency?

In Q5, the score of the schoolbook and the system are 5.5 and 7.75 respectively. In Q6, the score of the schoolbook and the system are 6.0 and 7.0. There are salient difference in Q5 and certain difference in Q6. In our subjective assessment, this result suggests that the learning using the system is more effective than the schoolbook.

In free description, we obtain feedback that a learner wishes to group a sequence of operations automatically. We think that automatically giving all correct answers negatively affects learning. However, we think that it is effective for programs and algorithms learning that the system gives a learner hints of grouping and judges whether grouping is correct.

4. Conclusion

We propose a method to construct an educational system that supports learners to relate their understanding on concrete operations to both generalized algorithm and program codes. We implement the system and perform preliminary experimental evaluation. The result suggests that our system has some educational effects as we expect.

We want to improve the system according to the result of questionnaire. In the future, we plan to advance the research on the support of coding.

Acknowledgements

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

References

[1] Malmi, L., Karavirta, V., Korhonen, A., Nikander, J., Seppala, O. & Silvasti, P. (2004). Visual Algorithm Simulation Exercise System with Automatic Assessment: TRAKLA2 *Informatics in Education, 3*(2), 267-288.

[2] 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. *Proc. of the 17th International Conference on Computers in Education*, 59-66.

[3] 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', *Int. J. Knowledge and Web Intelligence, 1*(3/4), 273-288.

[4] Fossati, D., Eugenio, B. D., Brown, C., & Ohlsson, S. (2008). Learning Linked Lists: Experiments with the iList System *Proc. of the 9th International Conference on Intelligent Tutoring Systems,* 80-89.