The Method to Visualize the Domain-oriented-explanation of Program’s Behaviors

Fumihiko Anma, Taketoshi Ando, Ryoji Itoh, Tatsuhiro Konishi, Yukihiro Itoh
Faculty of Information, Shizuoka University.
E-mail: anma@cs.inf.shizuoka.ac.jp

Abstract
In this paper, we discuss the method of visualization of the explanations generated by our educational system. Our system gives domain-oriented-explanations of programs. Our system can help learners to understand model programs and to find bugs in their programs by themselves. It generates explanations of the program’s behaviors. Our previous system can output verbal explanations. However, it is not easy for novice learners to get a concrete image of behavior of the program. Therefore, we are trying to add a facility of generating explanations by visualizing the state of the world.

1 Introduction
The purpose of our research is to construct an educational system that helps novice-programming learners by explaining domain-oriented-functions of programs. Our previous system outputs explanations using vocabularies on the world where the problem is present. We call such a world ‘the domain world’ of the program. However, some learners cannot get a concrete image of behavior of the program from verbal explanations. Visual explanations are more effective to understand algorithms. Many algorithm animation systems have been developed [3]. In most of such systems that generate animations including concrete objects, teachers have to embody commands for drawing objects to the programs. On the other hand, our system can generate a visual explanation without such commands. In this paper, we discuss the method to generate visual explanations.

2 Our previous work
2.1 Domain world models
Our system has a domain world model and simulates a target program on the model, to understand behavior of the program. By using the result of understanding, it generates an explanation of the program’s behaviors. However, it is difficult to simulate behavior of all types of program on the only one domain world model. Then we examine programming exercises and classify them into 15 types. We have to design individual domain world model for each type of exercises [1][2]. In previous work, we constructed the ‘greater and lesser world model’ for the exercises solved by paying attention to numerical order such as an exercise of sorting and the ‘two dimensional space world model’ that is used for some numerical analyzing methods such as ‘Newton method’ or ‘Simpson method’.

2.2 Generation of a verbal explanation
Our system generates verbal explanations of target programs written in PASCAL language. In order to make a familiar explanation, our system likens values in an array to cards or balls. Figure 1 shows an example of verbal explanations generated by our system. It illustrates the verbal explanation of behavior of a sorting program on the ‘greater and lesser world’.

Figure 1: an example of the system’s outputs

3 Visualization of explanations of programs
3.1 Drawing procedure
When the system outputs visual explanations, they are desirable to correspond to verbal explanations. Therefore, our system generates one picture per each statement. Each picture shows the state after execution of each statement. In accordance with the order of execution of the program, the system shows each picture with a verbal explanation. In each picture, entities constructing a world model are drawn. The variations of entities are different by the domain world. For example, the ‘greater and lesser world’ can be constructed from various sizes of balls. We prepare procedures that draw entities for each domain world. We call the procedures ‘drawing procedure’.

3.2 Generation of drawing data
Each picture represents the state of the domain world. In order to draw each picture, it is necessary that the system know information on the world after execution of the statement, such as number of entities, positions of them, size of them and so on. We call the information the ‘drawing data’. We define the initial drawing data by each exercise. The drawing data can be generated by modifying the previous drawing data according to execution of a statement. The modification is done in the following way: First, the system identifies the operation corresponding to the statement. Next, it identifies the
entities operated by the statement. Then it modifies the drawing data by a procedure defined for each operation. As long as the system draws pictures of the 'greater and lesser world', it can identify an entity and an operation from the syntactic pattern of the statement. For example, the pattern “variable_1 := value_1” means that the operation “movement” is applied to “value_1” in order to move the value to the position “variable_1” regardless of context. Therefore, we define procedures to update a drawing data for every syntactic pattern of statement, i.e. for assignment statement, read statement, for statement, and if statement.

3.3 Example of the visualization system of the 'greater and lesser world'

For example, when a learner inputs the program shown in Figure 2, our system outputs verbal explanations (a), (b) and (c). The Figure 3 shows the output of the prototype system that visualizes the behavior of the program in the 'greater and lesser world'.

![Figure 2: a verbal explanation](Image)

(a) x:=a[2]; (b) a[2]:=a[1]; (c) a[1]:=x;
Figure 3: an explanation of sorting

4 Extension of the visualization

4.1 The problem of extension

In order to handle general domain worlds, we have to resolve a problem concerning to simulation. In the general world such as 'two dimensional world', it is usual that the same syntactic pattern of sentences have different roles in a general domain world. For example, in the 'two dimensional space', an assignment statement may have a role to define the intersection point of two lines, or a role to define the middle point of two points. Accordingly it is difficult to recognize the role of the statement by syntactic pattern of the sentence. As a result the problem of ambiguity of the role of the values in the statements is caused.

4.2 The drawing procedure and the drawing data in the world of the 'two dimensional space'

When we think extension of the visualization method, we can apply the following framework.

1 We define the drawing procedure of each entity.
2 We define the updating functions for each data.
3 The system draws the initial state of the world.
4 The system calculates the drawing data after executing a certain statement.
5 The system draws the following states by executing drawing procedure with the new drawing data.

In the 'greater and lesser world', the system can update drawing data by using the syntactic pattern of the sentence in the step 4. Instead, the system updates drawing data by referring the domain world model. In the simulation process, all entities in the domain world are discovered and each attribute of them is clarified. Moreover, each variable manipulated in each statement is connected with corresponding attribute of a certain entity. So, the system can update drawing data in the following way:

1) The system identifies a variable the value of which is changed by executing the targeted statement.
2) It identifies the entity operated by the statement by checking the connection between the variable and an attribute of the entity.
3) It arranges the drawing data for the entity, and adds the data to previous drawing data.

4.3 An Example of the visualization system of the 'two dimensional space' world

Figure 4 shows the output of the behavior of the Newton Method program.

![Figure 4: an explanation of the Newton Method](Image)

(a) x := x1; (b) x := ((x^3-2)/3x^2)+x1;

5 Conclusion

In this paper, we discuss the method to visualize the behavior of the program. The method are applied to the domain world of the 'greater and lesser' and the 'two dimensional space'. We have to extend the system that can output the visualization of the buggy program in future work.

References