Design and Implementation of a Simulator for the Operational Control of Creative 3D Assembly*

Sanguk Noh
School of Computer Science and Information Engineering
The Catholic University of Korea
43 Jibong-ro, Wonmi-gu, Bucheon-si, Gyeonggi-do 420-743, Republic of Korea
E-mail: sunoh@catholic.ac.kr

Abstract - *This paper presents the design and development of a simulator for the operational control of creative 3D assembly. We model the creative 3D assembly as a moving agent, which generates a sequence of actions according to the scripts previously defined at any given domain. To guide or control the operation of the creative 3D assembly, we formulate logical rules represented as if-then rules into knowledge base, and construct various knowledge bases for a series of 3D assemblies in different settings. We design and implement our simulator including programming blocks, which tests the operational control of 3D assemblies in simulated settings.*

Keywords: Object-oriented modeling and simulation, Creative 3D assembly, Definition and interpretation of behaviors, Operational control using reactive rules

1 Introduction

In this paper, we design and implement a simulator for the operational control of creative 3D assembly. The simulator shows the creative 3D assembly's behavior that maps any given percept sequence to an action. We model the creative 3D assembly as a moving agent, which generates a sequence of actions according to the scripts previously defined at any given domain. To guide or control the operation of the creative 3D assembly, we formulate logical rules represented as if-then rules into knowledge base [1], and construct various knowledge bases for a series of 3D assemblies in different settings. We design and implement our simulator including programming blocks, which tests the operational control of 3D assemblies in simulated settings.

The paper is organized as follows. In the following section, for the operational control of creative 3D assembly, we describe two kinds of programming blocks, which consist of behavior blocks and control blocks. The behavior blocks define a set of available actions at a given domain, and the control blocks decide what actions are chosen, when they are performed, and how many times they are executed. In the concluding section, we summarize the preliminary implementation of our simulator and discuss further research issues.

2 Defining behaviors using programming blocks

Our work is built on efforts by several other research groups who focus on making educational programming language, such as SCRATCH, which is developed by the Lifelong Kindergarten Group at the MIT [2, 3]. We also develop "drag-and-drop" programming blocks for the operational control of creative 3D assembly in simulated settings.

To define the behaviors of the 3D assemblies and test their operational control, our approach uses a purely reactive reasoning procedure [4, 5, 6, 7]. The goal of reactive systems is to directly respond to the condition with a predefined action. These systems could react very quickly to environmental conditions. We formulate logical rules represented as if-then rules into knowledge base, and construct various knowledge bases for a series of 3D assemblies in different settings.

We provide programming blocks with users to experience definition of behaviors for creative 3D assemblies in a virtual environment. For the operational control of creative 3D assembly, we design and implement two kinds of programming blocks, which comprise behavior blocks and control blocks. The behavior blocks define a set of available actions at a given domain, and the control blocks decide what actions are chosen, when they are performed, and how many times they are executed. Three panels are shown in Fig. 1. The leftmost panel presents a set of behaviors, for example, move forward, turn some degrees in a counterclockwise and in a clockwise direction, grab, and so on. The mid panel shows control programming blocks, for example, condition block and loop block. The rightmost panel illustrates the combination of action blocks and control blocks, and their sequence as a result. The "drag-and-drop" programming blocks, as depicted in Fig. 1, generate the scripts for the

* This work has been supported by the ICT R&D program of MSIP/IITP, Republic of Korea, [2014-044-063-003, "Development of the graphic engine and framework of 3D contents for the creative hands-on science education"].

operational control of creative 3D assembly in a simulated environment.

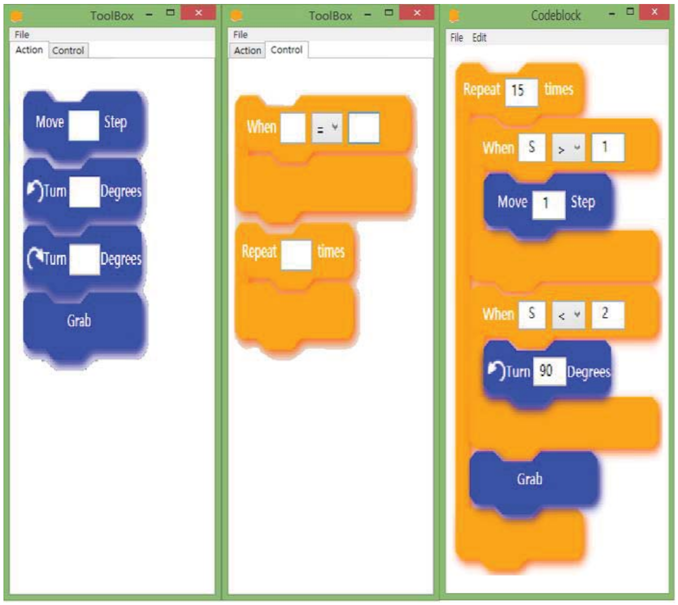

Fig. 1. Three panels representing action programming blocks, control programming blocks, and their scripts as a resulting sequence.

Fig. 2 illustrates a simulated testbed in which agents in the form of 3D assemblies can show their behaviors given a specific environment. The agent can move forward, turn left and right by 90 degrees, grab the ball in a 10×10 cell of rooms, as depicted in Fig. 2. If the agent keeps moving forward and bumps into a wall, then the agent does not move. Users can test the operational control of creative 3D assemblies along the various dimensions given a simulated environment, before they are made in real shape using 3D printers.

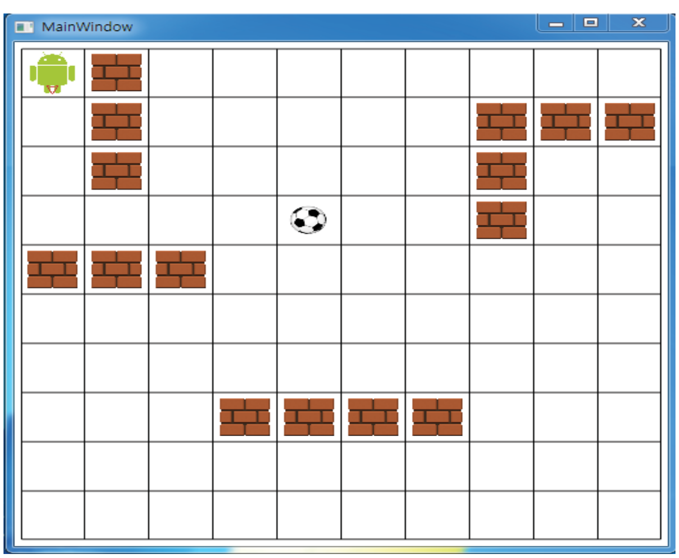

Fig. 2. Our simulator including a moving agent, an array of walls, and a ball.

3 Conclusions

In a virtual reality environment, we model the creative 3D assembly as a moving agent, which generates a sequence of actions according to the scripts previously defined at any given domain. To guide or control the operation of a specific 3D assembly, we formulated logical rules represented as if-then rules into knowledge base. The prototype of our simulator was designed and implemented, and the behaviors of 3D assembly defined by "drag-and-drop" programming blocks were also tested by in a simple grid environment. We are working on the expansion of the set of behaviors, and the capabilities of our simulator.

Based upon the preliminary implementation of our system, we will develop and construct various knowledge bases for 3D assemblies, and apply them to the assemblies built with 3D bricks, which are made for creative science education. Further, we will extend our inference system for the fuzzy operational control of creative 3D assemblies. Given vague input variables, our system using fuzzy logic [8, 9] could be used to control fuzzy operations, for instance, steering, accelerating, and braking, which usually could happen in operational control settings. Our inference system to support the assembly of 3D bricks and the operational control of creative 3D assembles, as well, then can be equipped with both crispy rules and fuzzy rules to be practical in real-world domains.

4 References

[1] Hao, C., *Research on Knowledge Model for Ontology-Based Knowledge Base,* In Proceedings of International Conference on Business Computing and Global Informatization, pp. 397-399, 2011.
[2] MIT News, *Kids coding in the cloud*, May 2013.
[3] SCRATCH, http://scratch.mit.edu/, 2015.
[4] Agre, P.E. and Chapman, D., *Pengi: An implementation of a theory of activity*, In Proceedings of the National Conference on Artificial Intelligence, Seattle, Washington, pp. 268-272, 1987.
[5] Fox, J. and Krause, P., *Symbolic decision theory and autonomous systems*, In Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence, UCLA, California, pp. 103-110, July 1991.
[6] Abraham, A., *Intelligent Systems: Architectures and Perspectives,* Recent Advances in Intelligent Paradigms and Applications, Studies in Fuzziness and Soft Computing, Vol. 113, Springer Verlag, Germany, pp. 1–35, 2003.
[7] Giarratano, J. and Riley, G., *Expert Systems: Principles and Programming*, Fourth Edition, Course Technology, October 15, 2004.
[8] Cordon, O., Gomide, F., Herrera, F., Hoffmann, F. and Magdalena, L., *Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends*, Fuzzy Sets and Systems, 141(1):5-31, Elsevier, 2004.
[9] Zadeh, L.A., *Toward extended fuzzy logic—A first step,* Fuzzy Sets and Systems, 160(21):3175-3181, Elsevier, 2009.