

(*tokA1* Team Description Paper)  
*tokA1* : A RoboCup 2D Simulation Team  
Developed Using a Novice Programming  
Environment

Yasuki IIZUKA<sup>1</sup> and Shun ENARI<sup>2</sup>

<sup>1</sup> Tokai University

<sup>2</sup> Toshiba Solutions Corporation

**Abstract.** The RoboCup Soccer 2D Simulation is the oldest RoboCup league, and has been variably used as a test bed for multi-agent systems [1]. Programming beginners cannot participate in RoboCup because the programs used for RoboCup Soccer 2D are very large and complicated. We want students who are interested in RoboCup Soccer 2D to participate more in games. We built the development environment yapEd that can develop a soccer agent program easily, in order to increase students' game participation. Tokai University's *tokA1* is the RoboCup 2D Simulation team developed using this development environment.

## 1 Introduction

RoboCup Soccer 2D Simulation requires advanced agent programming. The soccer agents have to play by grasping a situation through communication with a server and cooperating with a teammate. Beginners cannot use the programs easily because the soccer agent's programs consist of more than tens of thousands of lines. Akiyama has distributed the program source code of a strong team as the open source program Agent2D (HELIOS base) [2]. This Agent2D can be used as a base in agent development. However, the source code of Agent2D written by C/C++ is huge, and it is difficult to understand.

Nishino built the virtual soccer robot kit "OZED system" [3]. The aim of OZED was the programming education of junior high school students or high school students. There are two levels in the programming of OZED. The first is the stage where a user performs only an intuitive position setup using GUI. The next is the stage where a user programs an agent's action. However, stepping up to the second stage from the first stage is difficult for the user.

In this study, we designed a simple programming environment. By using this, programming beginners can develop a soccer agent. This environment features gradual set-up according to the user's programming skills. *tokA1* is the team developed using this environment.

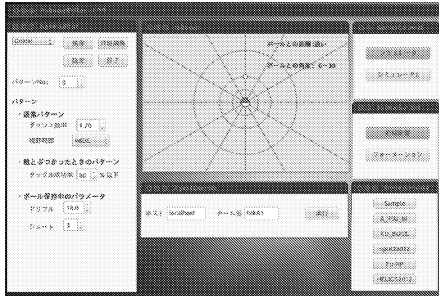


Fig. 1. Visual EDITOR

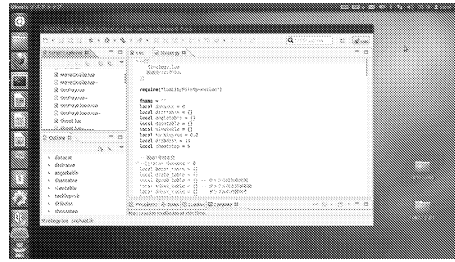


Fig. 2. LibSC/Lua

## 2 Script Language based Development Environment

### 2.1 Script Language

In this study, the script language Lua[4] was used for agent development. Lua is processed by an interpreter. It features dynamic typing of variables, and requires no compiling time. These features improve the productivity of the software. Lua is a language that is excellent in processing speed compared with other script languages, and it is easy to learn. It is also easy to link with C/C++ programs. Lua has been adopted in the development tools for the Robocup 4 legged league [6], or has been adopted in the robot development tools[5]. We embedded Lua in Agent2D[2] aiming at the simplification of agent development.

### 2.2 Development Environment

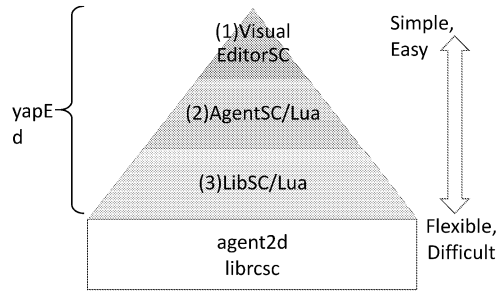
In this study, we developed yapEd as the soccer agent development environment. Programming flexibility will diminish if agent development is simplified. As a result, only similar soccer teams will be created from this development environment, which is not interesting for beginners. This is the dilemma of a simple development environment. For programming beginners, a balance between flexibility and programming simplicity is required. In order to resolve this issue, we prepared the following three program development tools.

#### (1) Visual EDITOR

Visual EDITOR is a parameter adjustment tool of a GUI base for agent development (Fig.1 .) The decision-making rule according to the situation of the game can be specified using this tool. Visual EDITOR also has the function to launch programs, such as a Soccer Server and a Soccer Monitor. By using this, team development and a test are also possible for the user who is unfamiliar with programming language.

#### (2) AgtSC/Lua

AgtSC/Lua is the sample program of a soccer agent. It is written using the script language Lua and can freely adjust an agent's action or a parameter.



**Fig. 3.** Agent development model

**Table 1.** Code Comparison

	Lua	C/C++	Code ratio
Agent	21 lines	82 lines	3.9
Library (move module)	52 lines	99 lines	1.9

### (3) LibSC/Lua

LibSC/Lua is a library program to which the agent’s details of the action are written (Fig. 2). LibSC/Lua is the Lua program that uses Agent2D with LibRCSC as the base. This LibSC/Lua program is larger than AgtSC/Lua, and therefore, it has high flexibility. The possibility of various agent developments can be expected.

These three development tools were unified as yapEd (Fig. 3). The programming model has three stages, which can be selected according to the user’s programming skills.

## 3 Evaluation of the Development Environment

We evaluated yapEd by having a user utilize it. These users were freshmen at a university. We handed the users the development environment and a description. The users answered a description-type questionnaire after using yapEd. The questionnaire results revealed a high evaluation of yapEd. The general opinion was that the agent description of Lua is simple, and that agent development will become easier by using Lua. When the agent was actually developed, the program was 82 lines in C/C++, but it became 21 lines in Lua (Tbl.1). An example program is shown in Fig. 4. The details of the questionnaire result regarding the selection of agent development languages are as follows: “want to use Lua” – 27%, “would like to use Lua, but also interested in C/C++” – 64%, “would like to use C/C++, but also interested in Lua” – 9%, “want to use C/C++” – 0%.

```

require("robocup")
function mainLoop()
    if(isKick()) then
        Kick()
    else
        Move()
    end
end
function Move()
    basic_move()
end
function Kick()
    if(not basic_kick()) then
        offensive_kick()
    end
end
end

```

**Fig. 4.** Example of an agent program written by Lua

To use this system, the user needs to master Lua in order to develop an agent. One opinion was that the agent's customization was difficult if one did not learn Lua. Ultimately, using a new programming language lowers the motivation of program development for beginners.

## 4 Conclusion

In this study, we developed yapEd to facilitate agent development. The yapEd features a tool that can be selected according to the user's skill. We expect the users of RoboCup to increase by the use of yapEd.

*tokA1*, which is the team developed using yapEd, won the RoboCup Japan Open 2013.

## References

1. Noda, I. and Matsubara, H. : Soccer Server and Researches on Multi-agent Systems, Proceedings of IROS-96 Workshop on Robocup, pp.1-7, (1996)
2. Akiyama, H. : Agent2D <http://rctools.sourceforge.jp/pukiwiki/> (2010)
3. Junji NISHINO, Takenori KUBO, Hiroki SHIMODA, Tomoharu NAKASHIMA : Educational Soccer Simulation system OZED, SCIS&ISIS2006 @ Tokyo, Japan, pp.63-68(2006)
4. R. Ierusalimsky, L. H. de Figueiredo, and W. Celes.: Lua: an extensible extension language, Software: Practice & Experience Vol.26 No.6 (1996) pp.635-652.
5. Tim Niemller, Alexander Ferrein, Gerhard Lakemeyer : "A Lua-based Behavior Engine for Controlling the Humanoid Robot Nao", RoboCup 2009, LNAI5949, pp.240-251, (2010)
6. Hayato Kobayashi, Akira Ishino, and Ayumi Shinohara. "A Framework for Advanced Robot Programming in the RoboCup Domain", In Proceedings of the 9th International Conference on Intelligent Autonomous Systems (IAS-9), pp.660-667. IOS Press, March 2006.