clooca : Web based tool for Domain Specific Modeling

Shuhei Hiya*, Kenji Hisazumi, Akira Fukuda, and Tsuneo Nakanishi

Kyushu University 744 Motooka Nishi-ku, Fukuoka 819-0395, Japan
{hiya,nel,fukuda,tun}@f.ait.kyushu-u.ac.jp
http://portal.isee.kyushu-u.ac.jp/

Abstract. Clooca is a development environment which allows us to build up Domain-Specific Modeling Languages, or DSMLs, and its code generators. DSMLs are widely used to improve productivity and quality of developing software to raise the level of abstraction and to generate a fully functional software codes. Nowadays some tools to develop software are provided as web services. We can use the tool without installing efforts and professional knowledge of software development. Even domain specialists who do not have enough knowledge to develop software can use it. Therefore we tackle with developing SaaS type DSML Tools to bring both benefits of DSML and SaaS type development tools.We use clooca in education at first, because this tool solve the some problems in that field. The link of youtube is http://www.youtube.com/watch?v=VS5VNBOYA_o.

Keywords: Domain-Specific-Modeling, DSML, SaaS, Code Generator

1 Introduction

Clooca is a development environment which allows us to build up Domain-Specific Modeling Languages, or DSMLs, and its code generators. Clooca is provided as a Service, similar to Gmail. We just open a web browser supporting HTML5 and JavaScript, input URL of clooca, and just use it. There are no installing and setting up efforts. DSMLs are widely used to improve productivity and quality of developing software[1][2][3]. It raises the level of abstraction rather than describing programming languages. It allow a domain specialists, who do not have enough knowledge to develop software with programming language, to develop software describing these abstract models. It can generate fully functional codes from these models. We use clooca in education about Software Modeling.

To provide a development tool as a web service brings many benefits in education. In the class of Software Modeling, many students use modeling tool. The time of installation is also large for that, nevertheless teachers can use clooca without installing efforts for many students and professional knowledge of

* Graduate School of Information Science and Electrical Engineering of Kyushu University.

software development. In addition, teachers and students can share and analyze their model in real-time on clooca.

DSML tool has two benefits in education field. First, It raises the level of abstraction rather than describing programming languages. So teachers can define a modeling language each level of the student. Second, Because it can generate fully functional codes from these models, students can execute the model soon after draw, and they can check how the model would work.

The rest of the paper is organized as follows. We discuss architecture of the clooca and its implementations in Section 2. Section 3 demonstrates the clooca in real usage, and we conclude this paper in Section 4.

2 Implementation of clooca

JSON(JavaScript Object Notation)[4] based Model representation We identify requirements for the scheme of DSML definition for a SaaS type DSML tool as follows. The model representation of the scheme should be easy to serialize and deserialize. It is necessary to serialize and transfer the model between a web browser and a server, since clooca is a web browser based tool. The serialized form should be used on many platforms used by constructing web services (e.g. JavaScript, Python, PHP, and Ruby). We are able to serialize and deserialize the model without compiling or generating any stubs. The system should be able to deal with these models dynamically. The scheme allows us to define DSMLs easily. It is desirable that the representation of both of the model is not complicated. Not only the model should be defined by tools, but also should be defined by hand. In other words, the representation form of the model should be human readable. Figure 1 shows a DSML model. An extract of JSON expression of the DSML model is shown as Figure 2.

Architecture Figure 3 shows the architecture of clooca. Clooca consists of a server and a client. The clooca server runs on web servers such as Apache. The clooca server has functions to load and store models and meta-models according to request from clooca client. It is a kind of database wrapper. We employ Javascript to implement the server. The clooca client runs on web browsers which support HTML5 and JavaScript such as Google Chrome. It consists of the workbench and the model editor. The workbench is a tool to define DSLs and to develop code generators. The model editor is a tool to edit concrete DSLs.

Workbench The workbench is a tool to define DSMLs and to develop code generators. Figure 4 shows a screenshot of the workbench of clooca. It is quite complicated to define a DSML to describe a JSON description as you see. We can define DSMLs using GUI in the workbench rather than describing JSON description mentioned in the previous section. The workbench allows us to write the definition with table forms of diagrams, objects, relationships, and properties. We can develop templates to generate codes in the workbench. Next Subsection shows how to describe these templates.

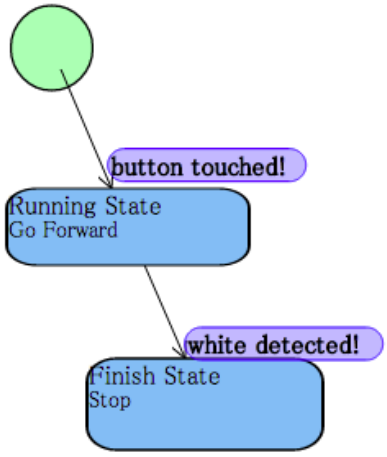

Fig. 1. An example of model.

```
{
    name : "Controller",
    states : {
        s1 : {
            srcTransitions : {
                t1 : {
                    target : "states.s2",
                    event : "button touched"
                }
            }
        },
        s2 : {
            name : "Running State",
            action : "Go Forward",
            srcTransitions : {
                t2 : {
                    target : "states.s3",
                    event : "white detected"
                }
            }
        },
        s3 : {
            name : "Finish State",
            action : "Stop",
        }
    }
}
```

Fig. 2. JSON expression.

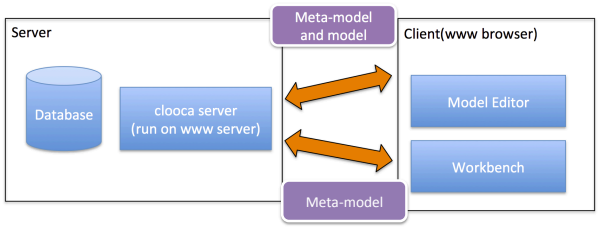

Fig. 3. Architecture of clooca.

Model Compiler The model compiler framework facilitates to develop model compilers which gets a DSL description and generates source codes according to a template and the DSL description. The model compiler runs on the server side. The model compiler framework is developed with ejs[5]. els is a template library written in Javascript.

Figure 6 shows an example of the code template to generate a name list of objects of " action " defined in Figure 2. We can access all of diagrams, objects, relations, and properties in the DSML model to describe an object reference code of Javascript in the code template. All of them are bound to "_" as a top-level variable. For example, if we want to access all of objects of "action", we can get a reference to the states to describe "_.states". In the code template we can use Javascript expressions with special markers. A line begun by " <%" marker indicates that the line is a control structure." <%expression% >" means that it will be substitute a concrete string generated to evaluate Javascript expression.

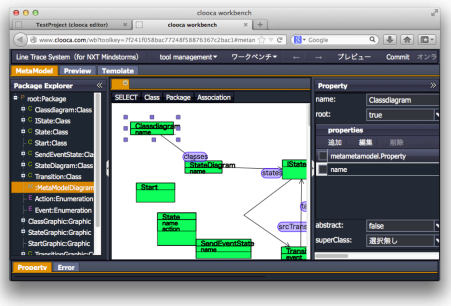

Fig. 4. A screenshot of the Workbench.

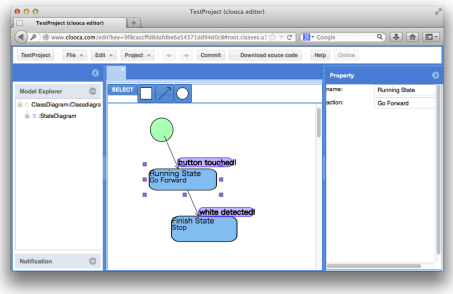

Fig. 5. A screenshot of the Editor.

```
<%for(var key in _.states) {%>
 <%=_.states[key].action%>
<% } %>
```

Fig. 6. An example of code template.

Model Editor The editor is a tool to draw models defined by the workbench and to generate codes according to code templates and the model. Figure 5 shows a screenshot of the Editor of clooca.

3 Evaluation

As a test, we use clooca to define diagrams and code templates for some applications.

3.1 A class of Software Modeling[7]

We use clooca for class of Software Modeling. In this class,studens design class diagram and state machine diagram for the line trace robot and run the robot. Teacher Define DSML for line trace robot on clooca, and students use DSML. Teachers are always worried about building up environments of a lab where it employs special tools. We can success to use in these labs without any install efforts even though a number of students attending these classes is about 80.

3.2 For a Project Based Learning and a Classroom

We use clooca as a UML editor for Project Based Learning or PBL in our university. In the PBL, students form four teams in which there are four or five students. Task of the PBL is to enhance a web-based bookstore system based on existing codes according to their original requirements. Teachers define class

and activity diagram of UML to use the workbench of clooca in advance of starting the PBL. The students design the bookstore system using clooca with UML meta-model. In the PBL, they did not employ a scheme of code generation. We also employ clooca to lecture the concept of model-driven development and domain-specific modeling language. A teacher defines a simple class diagram and a state-machine diagram of UML and develops code generators which generate JavaScript codes running on web browser according to the input model. We can execute the generated code on browsers. A number of students is about 20. We can success to use in this class without any install efforts.

4 Conclusion

Clooca is a Software as a Service type development environment which allows us to build up Domain-Specific Modeling Languages, or DSMLs, and its code generators [6] . We can use clooca without any installing effort. We proposed a novel method to define DSMLs which is suitable for web based services including clooca. We describe the architecture of the clooca and its implementations which employs the method to define DSMLs. We showed some cases to prove usefulness of clooca. These cases figure out the benefits of the SaaS type DSML Tools.

Our future work is to evaluate quantitatively with effort according to the deployment of clooca, for example we should compare the time it takes to set up and installation with the conventional tool.

References

1. Steven Kelly and Juha-Pekka Tolvanen. Domain-Specific Modeling: Enabling Full Code Generation. Wiley-IEEE Computer Society Press, 2008.
2. S. Cook, G. Jones, S. Kent, and A. C. Wills. Domain Specic Development with Visual Studio DSL Tools (Microsoft .Net Development). Addison-Wesley Longman, 2007.
3. Graphical Modeling Project. http://www.eclipse.org/modeling/gmp/ [Online; accessed 1-July-2013].
4. Introduction JSON, `http://www.json.org/` [Online; accessed 1-July-2013].
5. ejs Javascript Template, `http://embeddedjs.com/` [Online; accessed 15-July-2013].
6. clooca `http://www.clooca.com`
7. Kento Masumoto, Mizue Kayama, Shinpei Ogata, Kazunori Itoh, Masami Hashimoto, Makoto Otani. A Basic Study of Teaching Methods for Model-Based Design with Error Analyses of Beginners. ESS2012. pp101-109, 2012.