A study of a tool platform for developing formal specification

Takahiro Seino
t-seino@jaist.ac.jp
School of Information Science
Japan Advanced Institute of Science and Technology

March 10, 2005.

1 Overview

Since computer systems are pervasive and have a major impact on society, such systems must be built safely and reliably. One of the existing approaches to this issue is to formally model (the designs of) such systems and formally verify that they have desired properties. Computer systems are often modeled as transition systems. If a computer system can be modeled as a finite transition system, model checking techniques may be the most useful. Otherwise, theorem proving techniques should be used.

In our approach, a computer system is modeled as an OTS (Observational Transition System), a kind of transition system, which is described in CafeOBJ. It is then verified that the systems have desired properties by writing proof scores in CafeOBJ and having the CafeOBJ system execute (or rewrite) the proof scores. We have demonstrated the efficiency of our approach by performing several case studies. However, proof scores have been entirely written by hand, which means that human errors might have occurred. To minimize human errors, writing proof scores should be mechanically supported.

We think that the difficulty to develop tools treating CafeOBJ specifications causes lack of such tools around CafeOBJ. Since CafeOBJ's users can speficy that their own syntax of terms such as `_+_` or `if_then_else_fi` (An underscore `_` indicates the place where an argument is put), we cannot parse specifications with parser generators such as Yacc. To solve this problem, we propose an XML scheme called CafeOBJ/XML as a tool platform and present a tool called Buffet which converts the original specification to the XML version of it. The main advantage of this idea is that we can use common XML parsers to parse specifications, and most programming languages have such parser. This advantage makes easy to develop such tools.

In this paper, we also presents three applications to demonstrate that the proposed scheme is a suitable for such platform. One of these applications can generate proof scores automatically, it is guaranteed that generated proof scores cover all cases, excluding human errors.

2 System developments

We have completed implementing and testing the following software in this fiscal year.

2.1 CafeOBJ/XML

CafeOBJ/XML[2][6][7] is a scheme to describe specifications and proof scores of CafeOBJ. This scheme is given by means of DTD (Document Type Definition). XML documents according to CafeOBJ/XML are produced by Buffet and Gateau. We have demonstrated that CafeOBJ/XML is a useful tool platform with developing below tools.

2.2 Buffet

Buffet[2] is a toolkit for converting specifications written in CafeOBJ to CafeOBJ/XML and reducing terms with CafeOBJ's rewriting engine. Buffet is implemented as an HTTP server which can be used by multiple clients.

2.3 Gateau

Gateau[2] is a proof assistant for verifying whether a state machine have invariant properties or not. Given predicates used to split cases and lemmas, Gateau automatically generates proof scores and checks the proof scores with Buffet. Although the success of a proof depends on given predicates for case analysis and lemmas, it is guaranteed that generated proof scores cover all cases, excluding human errors.

We have done three case studies[2][4] that Gateau can been applied to a simple mutual exclusion protocol, NSLPK and Otway-Rees authentication protocol.

2.4 Proof Score Presenter

Proof Score Presenter (PSP)[2] is a pretty printer for proof scores written in CafeOBJ/XML. Given an XML document of a proof score and the results of reducing the proof passages in the proof score, PSP generates an HTML document like Figure 1. When an HTML document generated by PSP is first displayed on a web browser, proof passages for which results are not `true` and their results are shown, and other proof passages (namely those for which the proof has succeeded) are hidden. Proof passages are hierarchically shown according to the predicates used to split cases and each proof passage is clickable, allowing the proof passage to appear and disappear.

2.5 Executable code generator

We have implemented a code generator[6][7] which takes a specification of a state machine and generates an executable code for JAVA. CafeOBJ/XML allows this tool to parse information of state transitions easily. A rule set for generating executable code is verified with CafeOBJ.

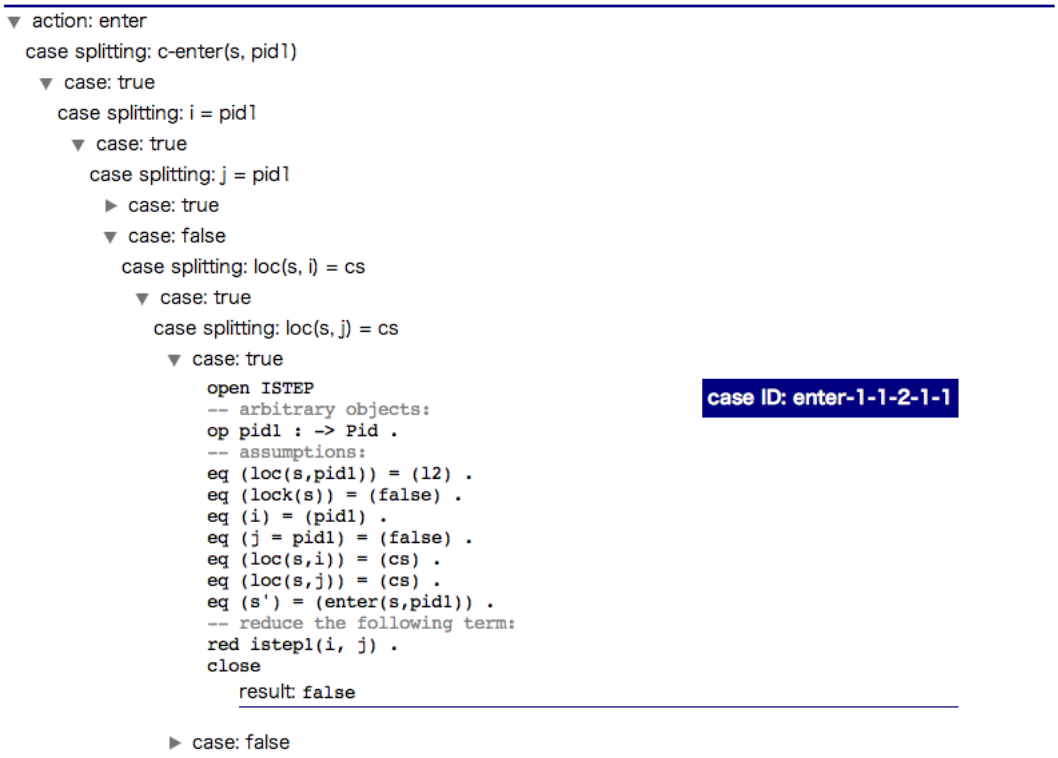

Fig.1 An output of Proof Score Presenter.

3 Future direction

Now, we plan the following research:

- Maude is a similar language to CafeOBJ and it has a good model checking tool. Although complex problem cannot be solved by model checking, it is useful to find counter examples. Therefore, developing a translator which translates a state machine written in CafeOBJ/XML to Maude is a good application of CafeOBJ/XML.
- We have learned from developing Gateau, generating proof scores can be manage more systematically. We plan that managing proof scores is implemented as an additional service of Buffet. This service makes easy developing proof assistants such as Gateau.
- All tools presented in this paper have character based interfaces. We plan to implement these tools on Eclipse which has a good GUI. Since Eclipse is very popular IDE (Integrated Development Environment), this plan makes a good opportunity to introduce our method to industrial society.

4 Publications

Refereed journal paper

[1] Takahiro Seino, Kazuhiro Ogata and Kokichi Futatsugi: Mechanically Supporting Case Analysis for Verification of Distributed Systems. *International Journal of Pervasive Computing and Communications.* (Submitted. A preliminary version appeared in Proceedings of CIT 2004).

Refereed Conference/Symposium/Workshop papers

[2] Takahiro Seino, Kazuhiro Ogata and Kokichi Futatsugi: A Toolkit for Generating and Displaying Proof Scores in the OTS/CafeOBJ Method. Proceedings of the 6th International Workshop on Rule-Based Programming (RULE '05). (To appear).

[3] Takahiro Seino, Kazuhiro Ogata and Kokichi Futatsugi: Supporting case analysis with algebraic specification languages. Proceedings of 4th International Conference on Computer and Information Technology (CIT 2004), pp. 1100-1107. IEEE CS Press, 2004.

[4] Takahiro Seino, Atsushi Kato, Kazuhiro Ogata and Kokichi Futatsugi: Verification of Otway-Rees authentication protocol with rewriting. Proceedings of the 11th Workshop on Foundation of Software Engineering (FOSE 2004), pp. 153-156. 2004. (In Japanese)

[5] Kazuhiro Ogata, Daigo Yamagishi, Takahiro Seino and Kokichi Futatsugi: Modeling and Verification of Hybrid Systems Based on Equations, Proceedings of the IFIP TC10 Working Conference on Distributed and Parallel Embedded Systems (DIPES 2004), Kluwer Academic Publishers, pp.43-52, 2004.

[6] Jittisak Senachak, Takahiro Seino, Kazuhiro Ogata and Kokichi Futatsugi: Provably Correct Translation from CafeOBJ into Java. Proceedings of The 17th International Conference on Software Engineering and Knowledge Engineering. (Submitted).

Unrefereed Conference/Symposium/Workshop paper

[7] Jittisak Senachak, Takahiro Seino, Kazuhiro Ogata and Kokichi Futatsugi: Provably Correct Translation from CafeOBJ into Java. Forum on Information Technology 2004 (FIT 2004).