Promotion of Formal Approaches in Japanese Software Industry and a Best Practice of FeliCa's Case (Extended Abstract)

Keijiro Araki[1(✉)] and Taro Kurita[2]

[1] Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
araki@ait.kyushu-u.ac.jp
[2] Sony Corporation, 2-10-1 Osaki, Shinagawa-ku, Tokyo 141-8610, Japan

Abstract. We have been making much effort to promote formal methods in Japan, especially Japanese IT companies. This paper describes our activities in Japan for almost twenty years, and shows typical reactions from such Japanese companies for application of formal methods. We mention about the obstacles they think to adopting formal methods in their real software development projects. On the other hand we also present a case of FeliCa Networks, Inc. as a best practice of applying formal methods in Japan. We discuss the lessons learned from our efforts of promoting formal methods and the FeliCa's case. Finally, we briefly introduce our research project to support software developers in adopting formal approaches to real projects.

Keywords: Formal methods · Rigorous specification · Practice · Development process · FeliCa IC chip · VDM · VDMPad · ViennaTalk

1 Introduction

We promote formal methods in Japan, and reported the status of formal methods in Japan. [1,2] Many Japanese companies are interested in formal methods in development of software systems. Some companies introduce and apply formal approaches successfully in their own development processes. However, there are not so many companies apply formal methods in their real development projects.

In this paper, we report our activities to promote formal methods in Japan, and briefly introduce our current research project.

2 Promotion of Formal Methods in Japan

2.1 Seminars and Publications

We have a variety of activities to promote formal mthods under the collaboration with several kinds of organization such as SIG-FM of SEA (Software Engineers Association of Japan), IPA (Information-technology Promotion Agency,

K. Ogata et al. (Eds.): ICFEM 2016, LNCS 10009, pp. 18–25, 2016.
DOI: 10.1007/978-3-319-47846-3_2

Japan)/SEC (Software Engineering Center/Software Reliability Enhancement Center), local public communities, private companies, universities, and so on. For example, we have had a series of seminars at Hiroshima, Kumamoto, Sapporo, Nagoya, Osaka, Tokyo, Okinawa, Ho Chi Minh City, Fukuoka, Nagano, Morioka, etc. We also made teaching materials for those seminars, and make them open on the IPA/SEC site. We published several reports on formal methods. Especially the report on successful cases of formal approaches to software development [6] focuses on eleven cases and presents many suggestions for applying formal methods in real systems.

2.2 Reactions from Japanese Companies

Many company people attend our seminars and become interested in formal methods. Especially the successful cases in Japan are quite attractive to them, and they want to know much in details about those cases. They also require to know more cases which may fit for their own projects. We survey such cases, and then they require much more. However, most of the companies will not adopt formal methods in their own projects.

The following is seven typical reactions from the Japanese companies:

- Real successful cases are interesting and attractive.
- Formal methods exist in the perfect formal world which seems unrelated to their everyday activities.
- Formal methods require highly trained mathematicians.
- Formal methods look difficult for them to apply by themselves.
- They need complete samples for their own problems.
- It is difficult to convince the top management of the benefits of formal methods.
- The cost-performance and effectiveness of formal methods are unknown.

Many of the above seem just excuse not to adopt formal methods.

3 A Best Practice of Formal Approach in Japan

Yes, we have applications of formal methods in Japan. Some of them adopt formal approaches to development of their final products. Some have continuous effort to learn formal methods with their own training courses and trials to apply formal methods to their projects.

We learn much about the practice of formal methods through the exchanges and collaborations with them. Especially, A best practice of formal approach to software development in Japan is the FeliCa's case to develop smart IC chips. [7,8] Fig. 1 shows the development process of FeliCa IC chip firmware with the formal specification in VDM++.

They only describe the specification in VDM++, and did not perform any formal proof nor code generation. The rigorous system description in VDM++ has good effects on the whole development process, and then they realized extremely high reliable products. No bug has appeared among more than 300 million chips which are used in a very wide variety of applications.

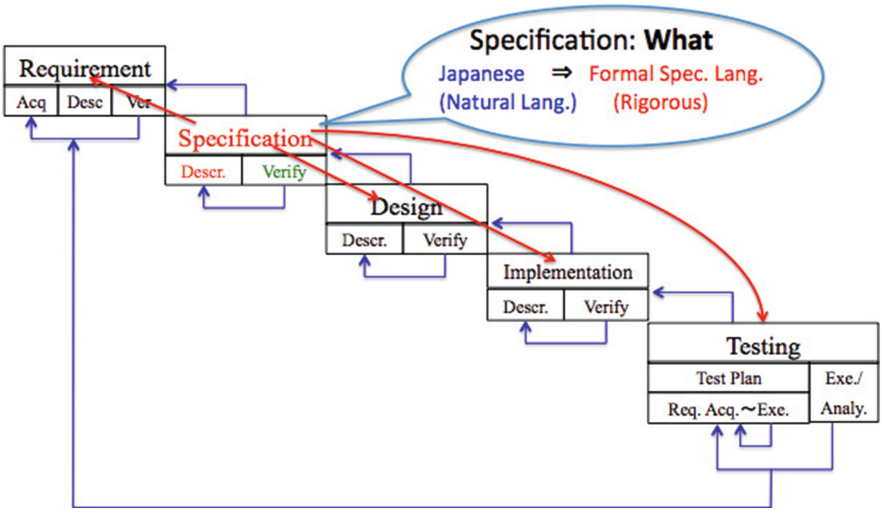

Fig. 1. VDM specification in the development process at FeliCa.

4 Formal Methods for Working Engineers

We aim to propose formal approaches applicable and effective in the real software development projects. We focus on the way how to incorporate formal development methodologies into the development process of each specific project in an IT company. We propose a development process for a specific purpose as a reference model. [9,10] We recommend the developers to customize our reference process model for their own project.

Figure 2 shows the overview of our current research project to propose formal approaches effective over the whole software life cycle. There exist many theories, methodologies, tools in formal methods. We believe each of them is applicable and effective for a specific purpose at some phase(s) in the software life cycle.

We have developed several tools to support construct rigorous descriptions for the target systems. Here we briefly introduce the following three tools:

- VDMPad: Interactive VDM-SL Tool
- ViennaTalk: IDE for Lightweight Formal Approach
- JOD: Dictional Tool Bridging Informal documents and Formal Documents

4.1 VDMPad

VDMPad is an interactive VDM-SL tool. [11] Fig. 3 shows a screen shot of VDMPad. A user writes a VDM-SL description in the top window. VDMPad performs the syntax and type checking for the VDM-SL description. If the state invariants are included in the description, VDMPad also check the invariants. In the second window, the user can write an expression to be evaluated by the VDMPad. And

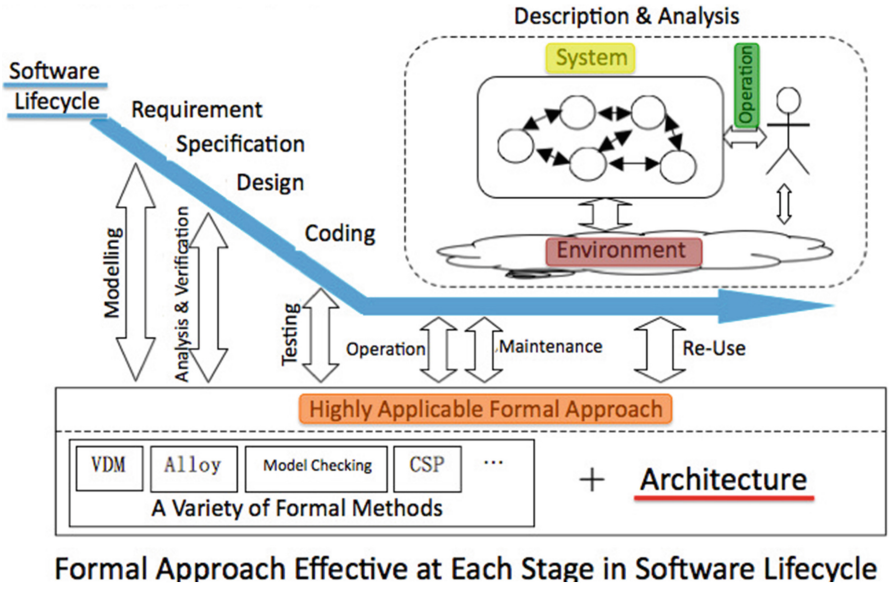

Fig. 2. Overview of our research project.

then the result of the evaluation appears in the third window. If any error is detected, the message is output beneath the bottom line.

VDMPad is very useful to write VDM-SL description with a simple and interactive user interface. For beginners of VDM specification, VDMPad is easy to use and understand how to write VDM-SL specifications. It is also a nice tool for matured specifiers to describe components in VDM-SL and validate them with testing at the specification level.

The VDMPad service is available on the VDMPad Server [14].

4.2 ViennaTalk

ViennaTalk is an IDE (Integrated Development Environment) for lightweight formal approach which is a SmallTalk library to handle VDM-SL specifications. [12,15] Major components of ViennaTalk is as follows:

- VDMBrowser - A VDM-SL browser inspired by Smalltalk's class browsers and inspectors
- VDMPad - A lightweight web IDE for VDM-SL with animation and diagram presentation of data
- VDMC - A Smalltalk wrapper of VDM-SL animation
- Lively Walk-Through - A UI prototyping environment to animate UI prototype by VDM-SL specification
- Webly Walk-Through - A Web API server to publish VDM-SL specifications of web APIs.

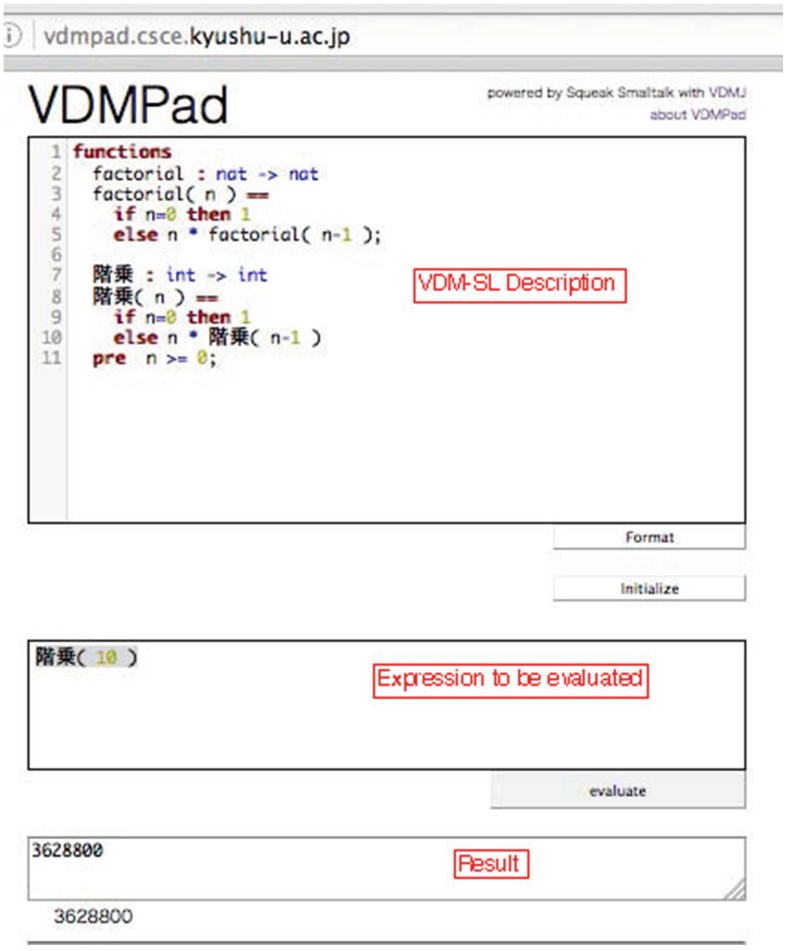

Fig. 3. VDMPad: interactive VDM-SL tool.

- ViennaEngine - Animation engine wrapper
  - ViennaVDMJ - Animation engine by local VDMJ process
  - ViennaServer - Web server to publish animation engines
  - ViennaClient - Client module of ViennaServer/VDMPad
  - ViennaBankEngine - Aggregated animation engine
- ViennaTalk-Types - Smalltalk classes for VDM types
- ViennaTalk-Values - Smalltalk objects for VDM values
- ViennaTalk-Parsers - VDM parsers, Smalltalk code generators and VDM source formatter.

At brain storming and reviewing stages in a system development process, Informal descriptions like hand-drawn diagrams are often used. Those informal

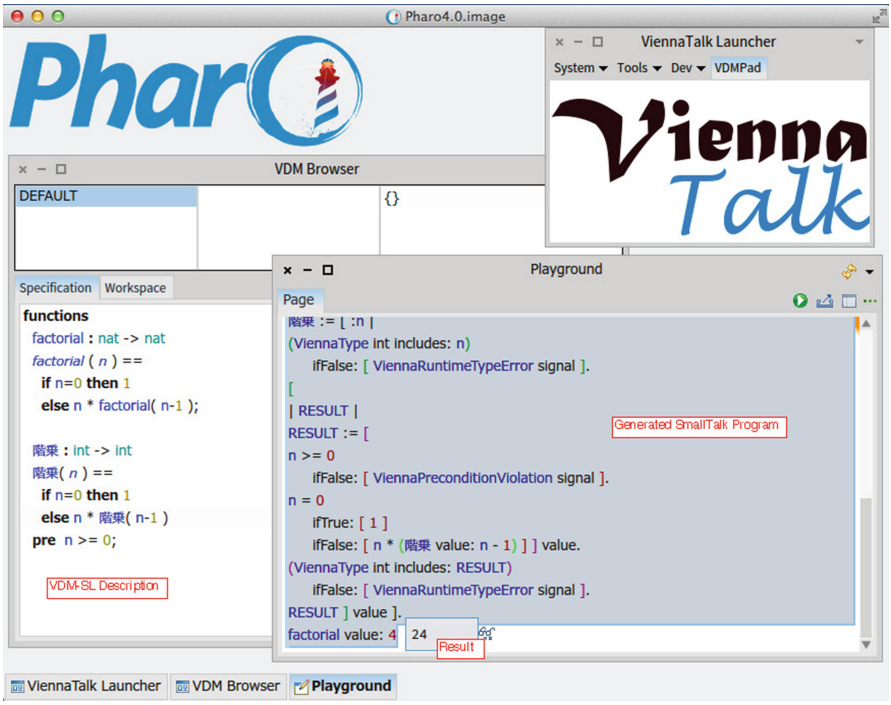

Fig. 4. ViennaTalk: IDE for lightweight formal approach.

description can be justified with VDM-SL easily on ViennaTalk. Lively Walk-Through and Webly Walk-Through support such work, and help the developers to clarify and share the ideas. The interactive animation of VDMPad also helps the validation of the ideas.

It can also generate a SmallTalk program from a VDM-SL description. The SmallTalk program is useful to test and validate the VDM specification with much wider data space. It also serves as a prototype of the specified system.

Figure 4 shows a screen shot of ViennaTalk. The left window shows a VDM-SL specification. The right window show the generated SmallTalk program for the VDM-SL specification. The small window with the value 24 is the result of the invocation of the SmallTalk program "factorial" with the parameter 4.

4.3 JOD Tool

The third tool is JOD, Dictionary Tool to Support Rigorous System Description. [13] It accepts informal documents written in a natural language, and manipulates the documents and provides useful information to clarify the system structures and functions. It bridges between informal documents and rigorous descriptions written in VDM-SL and VDL++.

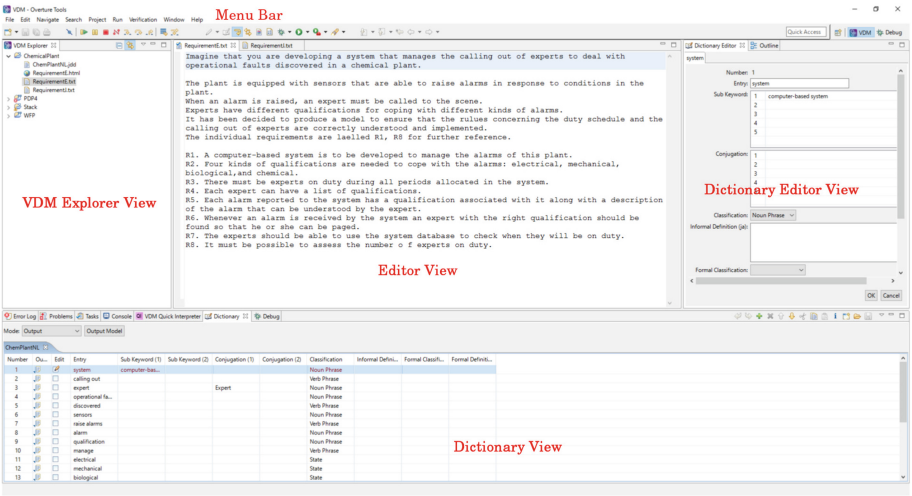

Fig. 5. Dictionary tool to support rigorous system description.

Figure 5 shows a screen shot of JOD. This tool is open to use and embedded in the Overture Tool.

5 Concluding Remarks

We have promoted formal methods in Japan, and provided seminars, tutorial lectures, teaching materials, reports and support tools. We always cite the well-known "Seven Myths of Formal Methods" [5] and "Ten Commandments of Formal Methods." [3] We find many suggestions and insights to introduce and adopt formal methods in real system development processes. In the seminars and lectures, we tell those suggestions and insights with our own experiences and thinkings.

We found one more attractive concept "Preformal" [4] and regard it as a good guiding principle in formal approaches to system development. It is not easy for the beginners to construct formal description in their system development project. They need to understand the essential properties of formal methods as well as their own project. They also need to realize what are their purpose to apply formal methods. And then they may decide which formal method and how to apply to their project.

We intend to propose the purpose-oriented preformal approaches to construct rigorous/formal system description. We are sure that our tools described above work well along the preformal approaches.

Finally, we state messages to the Japanese IT companies.

– Know thyself.
– Heaven helps those who help themselves.

These are very common teachings. We will continue to promote formal methods in Japan and support IT engineers and users keeping the above commandments and teachings in mind.

Acknowledgments. This work is partly supported by Grant-in-Aid for Scientific Research (S) 2422001.

References

1. Araki, K.: Are formal methods relevant?: how to explode the seven myths in Japan. In: Proceedings of the APSEC 1995, pp. 514–515 (1995)
2. Araki, K., Chang, H.-M.: Formal methods in Japan: current state, problems and challenges. In: Proceedings of the Third VDM Workshop, VDM 2002 (2002)
3. Bowen, J.P., Hinchey, M.G.: Ten commandments of formal methods. IEEE Comput. **28**(4), 56–63 (1995)
4. Gmehlich, R., Jones, C.: Experience of deployment in the automotive industry. In: Romanovsky, A., Thomas, M. (eds.) Industrial Deployment of System Engineering Methods, pp. 13–26. Springer, Heidelberg (2013)
5. Hall, A.: Seven myths of formal methods. IEEE Softw. **7**(5), 11–19 (1990)
6. IPA/SEC: Report on Successful Cases of Formal Approaches with Rigorous Specification, WG on Rigorous Specification, IPA/SEC, Tokyo (2013) (in Japanese). http://sec.ipa.go.jp/reports/20130125.html
7. Kurita, T., Nakatsugawa, Y.: The application of VDM to the industrial development of firmware for a smart card IC chip. Int. J. Softw. Inf. **3**(2–3), 343–355 (2009)
8. Kurita, T., Ishikawa, F., Araki, K.: Practices for formal models as documents: evolution of VDM application to "Mobile FeliCa" IC chip firmware. In: Bjørner, N., Boer, F. (eds.) FM 2015. LNCS, vol. 9109, pp. 593–596. Springer, Heidelberg (2015)
9. Kusakabe, S., Lin, H.-H., Omori, Y., Araki, K.: Developing core software requirements of energy management system forsmart campus with advanced software engineering. Int. J. New Comput. Archit. Appl. **4**(1), 48–55 (2014)
10. Kusakabe, S., Lin, H.-H., Omori, Y., Araki, K.: Visualizing centrality of process area networks in CMMI-DEV. In: Proceedings of International Conference on Software and Systems Process (ICSSP 2015), pp. 173–174 (2015)
11. Oda, T., Araki, K., Larsen, P.G.: VDMPad: a lightweight IDE for exploratory VDM-SL specification. In: Proceedings of the 2015 IEEE/ACM 3rd FME Workshop on Formal Methods in Software Engineering, pp. 33–39 (2015)
12. Oda, T., Araki, K., Larsen, P.G.: ViennaTalk and assertch: building lightweight formal methods environments on pharo 4. In: Proceedings of the International Workshop on Smalltalk Technologies (2016, to appear)
13. Omori, Y., Araki, K., Larsen, P.G.: JODTool on the Overture Tool to manage formal requirement dictionaries. In: Proceedings of the 13th Overture Workshop, Co-located with FM 2015, pp. 3–17 (2015)
14. VDMPad Server: http://vdmpad.csce.kyushu-u.ac.jp/
15. ViennaTalk: https://github.com/tomooda/ViennaTalk-doc

http://www.springer.com/978-3-319-47845-6

Formal Methods and Software Engineering
18th International Conference on Formal Engineering Methods, ICFEM 2016, Tokyo, Japan, November 14-18, 2016, Proceedings
Ogata, K.; Lawford, M.; Liu, S. (Eds.)
2016, XVII, 486 p. 144 illus., Softcover
ISBN: 978-3-319-47845-6