Formal Methods in Japan
— Current State, Problems and Challenge —

Keijiro Araki[1] and Han-Myung Chang[2]

[1] Deaprtment of Computer Science and Communication Engineering
Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581 JAPAN
araki@csce.kyushu-u.ac.jp

[2] Department of Information and Telecommunication Engineering
Nanzan University
27 Seirei, Seto, Aichi 489-0863 JAPAN
chang@it.nanzan-u.ac.jp

Abstract. Formal methods are not so popular in Japan. However, there exist several activities in promoting and applying formal methods recently. We describe such activities on formal methods in Japan as practical application to system development and education/training. We also descuss obstacles in the way of promoting formal methods.

1 Introduction

Formal methods are regarded as promising approaches to systematic and efficient development of high quality software systems [6]. The well-known technical essay on formal methods by A. Hall [9] discussed many misunderstandings about formal methods and their challenges. Bowen and Hinchey discussed yet other myths of formal methods [4].

Although much experience and research results had been accumulated especially in European countries, little efforts had been dedicated to formal methods in Japan [1]. We called it the Pre-Myth of formal methods in Japan. Yet several research groups have been working with formal methods even in Japan. Some work in theoretical foundations, some in development of methods and tools, some in practical applications. They have been advocating the formal methods in Japan and intended to apply them to real system developments.

In this paper, we briefly report some case studies of applying formal methods to real system development projects. We also discuss effectiveness of formal methods for system development in practice as well as problems or obstacles for the industrial poeple especially for the managers to accept formal methods [2].

In section 2, we briefly report some case studies in applying formal methods to real systems in Japan. In section 3, we discuss benefits from applying formal methods to real system development and problems in promoting them in Japan through those case studies. We give our concluding remarks in section 4.

2 Some Case Studies in Japan

2.1 Electric Feeder System

An electric power company supplies elecric power to end-customers such as homes and companies. We call the network from a substation of the elecric power company to a set of end-customers an electric feeder network. Figure 1 shows an simple example of electric feeder network.

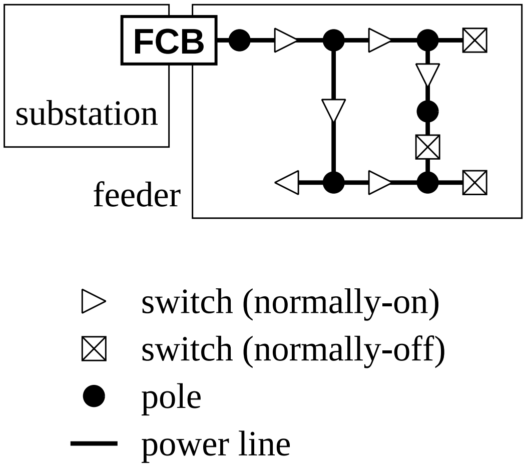

Fig. 1. Electric Feeder Network

FCB (Feeder Circuit Breaker) is the main switch of a feeder. There are two kinds of switches, i.e., normally-on switch and normally-off switch. At normal states, these switches remain on or off according to their kind. They will, however, be turned on or off in cases of accident.

The electric power company monitors the status of the network and must control the switches when an accident occurs with the network. For example, if the lightning makes a damage to the network and a line becomes short-circuited, then the FCB cuts off the electric current to the feeder. The company must detect the damaged part along the network, isolate the short-circuited section and then recharge the electric power to the network except the isolated section. Since the further section beyond the damaged need to be recharged with the electric power, the network must be reconfigured by opening another path from a source to the end-customers.

We have applied formal methods to such a system for monitoring and controlling the electric feeder network. It is natural to use graph as a mathematical tool in modelling the feeder network. We had many meetings with the domain experts from a manufacturing company and tried to make an abstract model for the electric feeder network system.

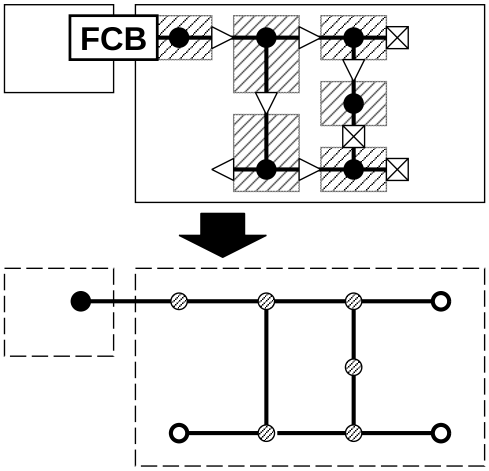

Fig. 2. Abstract Modelling (Dual Graph)

Figure 2 shows the way how to abstract a feeder network into a mathematical model. It might be a straightforward modelling to map a switch to a node in an abstract model and map a line to an edge.

In the above feeder network in the figure 2, each section is shaded, which consists of a set of directly connected lines. The electric current is carried from a charged section to the next sections through the switches turned on. We regard that sections are connected each other via switches. A section is mapped to a (shaded) node, and a switch is mapped to an edge in the graph model as shown at the bottom in the figure 2. It is a dual graph by converting the edges and nodes each other. We summarize this case study with the electric feeder system as in the table 1.

Table 1. Electric Feeder System Specification Summary

Period	about 6 months
Meetings	twice a month
Domain Experts	1 - 2
Specifiers (incl. Reviewers)	4 - 6
Languages	Z, VDM-SL, Java
System Model	graph
Products	model, formal specification, generic graph analysis tool [14], glossary linked with formal definitions

2.2 Automated Storage System

As another case study, we modelled and specified an automated storage system which stores a variety of goods and provides a required amount of a specifyed good. Such storage systems are used in many different ways at many places. Some are used to store and provide parts in manufacturing factories. Some are used to store and deliver consumer products.

Developers of such automated storage systems must satisfy a wide variety of clients' requirements which are to be changed frequently. The goods which clients deal with would be changed, the policies for dealing goods would be changed, the sizes of the storage areas would be changed, the conveying routes and vehicles would be changed, and so on.

We discussed with the domain experts and tried to make an abstract model for automated storage systems. As a very intuitive model, it might be a data structure to store thing such as stack or queue. A general policy for dealing the stored goods is First-In-First-Out for each kind of goods. So we regarded that a set of queues would be a basic data structure for storage systems. There is a case, however, that some goods remain in the container for a certain number of a request. The remaining goods must be stored back again in the storage area and must be delivered first at the next request.

Therefore we adopt the data structure deque (double ended queue) for each kind of goods. We specified a set of deques as a basic model for automated storage systems and specified primitive operations for it. We summarize this case study with the automated storage system as in the table 2.

Table 2. Automated Storage System Specification Summary

Period	about 6 months
Meetings	twice a month
Domain Experts	1 - 2
Specifiers (incl. Reviewers)	4 - 6
Languages	Z, VDM-SL, ML
System Model	multiple deque
Products	model, formal specification

2.3 University Accounting System

The third case is an accounting system at a univeristy administration office. They have ordered the accounting system developmnent to a software company. They spent a long time to define their own requirement before they started the competitive bidding for the system development. After the bidding, the contract software company joined to specify and design the system.

They have been using formal methods and object-oriented analysis and design methods at the upper stages of the system development. A part of the VDM specification is shown in the figure 3.

```
--
-- 単位会計
--

  単位会計 = 単位 * (単純単位会計 | 複合単位会計);
  単純単位会計 = 部署会計;
  複合単位会計 = set of 単位会計 inv s == s <> {};

  部署会計 = 部署 * (単純部署会計 | 複合部署会計);
  単純部署会計 = 目的会計;
  複合部署会計 = set of 部署会計 inv s == s <> {};

  目的会計 = 目的 * (単純目的会計 | 複合目的会計);
  単純目的会計 = 科目会計;
  複合目的会計 = set of 目的会計;

  科目会計 = map 目的科目 to (金額 * set of 属性);

  目的 = token;
  事業 = 目的;
  目的科目 = token;
  属性 = token;
  金額 = nat;
```

Fig. 3. University Accounting System Model (A Part)

They first clarified the notions and defined the words concerning the university accounting affairs rigorously. They shared the common understandings based on these notions and words among different sorts of members; a domain expert (an accounting officers at the university office), a formal specifier, an expert of onject-oriented analysis and design, dvelopers from the software company. They use the Japanese words in VDM specifications. We summarize this case study as the tables 3 and table 4.

2.4 Other cases

There are some other cases applying formal methods to practical system development in Japan. The following is a partial list of them:

Table 3. University Accounting System: Domain Analysis Sammary

Period	2 months
Meetings	6 times
Domain Experts	1
OO Analyser	1
Formal Specifiers	1
Languages	Z, UML
Products	requirement specification (usecase diagram, class diagram, Z specification)

Table 4. University Accounting System: Requirement Analysis Summary

Period	3 months
Meetings	one a week
Domain Experts	several
OO Analyser	1
Formal Specifiers	1
Domain Analyzer	2
System Engineer	1
System Designers	2
Languages	VDM-SL, UML
Products	usecase diagram, sequence diagram, class diagram, VDM Specification

- Railway Signal Control System [10]
  Organization : Railway Technical Research Institute
  Language/Method : VDM
- Window System [13]
  Organization : Unisys Japan
  Language/Method : Z
- Stock Trading System [11]
  Organization : JFITS
  Language/Method : VDM++
- Vending Machine
  Organization : Nanzan Univ., Kyushu Univ.
  Language/Method : Z, VDM, UML

3 Discussions

In this section, we discuss the effectivenewss of formal methods through those case studies as well as the problems and obstacles in promoting and applying formal methods at industrial communities in Japan. This discussion would show the current state of formal methods in Japan, too.

3.1 Effectiveness of Formal Methods

In first two cases, the modelling and specification team consisted of one domain expert from the company and four or five specifiers from the academic side. One chief specifier specified working descriptions and the others reviewed them and elaborated the model and formal specification. Each time a new version of model and formal specification are made by the specification group, they had a meeting together with the domain expert and specifiers. The specifiers presented their model and specification to the domain expert in both formally and informally with natural languages and illustrations. The domain expert understood the presented models and specifications, and examined whether the specification is confirmed with respect to his real doamin knowledge. And then the specifiers got further knowledge about the domain. They improved and enhanced their models and specifications. In about a half year, they obtained a stable and elaborated model and formal specification for each of the two cases.

They have specified only a small part of the whole system in each case. But they have treated a most important core of the system and primitive operations. Further extensions, enhancements and modifications would be performed based on those core specifications.

At a first few meetings, they did not have fruitful discussions because the specifiers had no domain knowledge nor the domain expert could not understand formal languages. After several meetings they began to have a common sense in domain modelling and formal specifications. And then they had very efficient meetings with straight forward discussions based on regorous description and common understandings.

New comers could easily join the projects with formal specifications. In our cases, undergraduate students joined our projects and obtained proper understandings very quickly. They enhanced and elaborated the existing specifications and they quickly made prototypes and visualized them with Java or a functional programming language, too.

It is not so difficult even for practioners to read formal specifications. They could read them after introductory lectures courses and a little training. It is, however, difficult to write a formal specification. The syntax is simple and easy to learn, but a proper recognition of the given problem and making abstract models are essencial. Specifiers need experience and insights in system analysis and modelling.

We also produced a well structured document set for the electric feeder network system. It consists of both formal specification and informal description including glossary and specification of system elements. Formal and informal parts correspond each other with links on web pages. We intend to elaborate this document set and propose a framework for well organized and maintained design document sets. They will serve as reference models in those domains and support unambiguous discussions and common understandings among the poeple concerned in a particular system development. We have also clarified concepts in the domains; both basic concepts and specific detailed concepts.

In the third case, formal approaches to analyze, model and specify the accounting system work very well. Especially, abstract models and rigorously defined words play improtant roles in inter-cultural communication among a variety of members.

At the domain analysis phase, formal methods serve as a communication vehicle between the formal specifier and the OO analyzer. They used both formal description in Z and informal UML diagrams. Formal methods supported system abstraction and finding common structures in the acoounting system.

At the requirement analysis phase, formal methods serve again as an effective communication vehicle among a wider variety of members including the system developers. Meeting logs contain formal description about the concensus obtained at the meetings and supported efficient discussions at the successing meetings. Formal description in VDM served as the specification of the UML diagrams and assigned meanings to the diagrams partially. This combination of the formal description and the UML diagrams helped the system developers from the software company to understand the intention of the client.

3.2 Problems in Japan

Formal methods are still lesser known in Japan. Most of software engineers are unfamiliar with them. They hardly have a chance to learn formal methods. Only a small number of researchers and engineers study formal methods and try to apply them to their own problems.

If working engineers in Japan want to apply formal methods to their practical system development, their managers would not allow them to adopt formal methods unless development costs are reduced. To convince such managers, we must show the cost effectiveness of formal methods with an enough number of practical case studies. We also need a way of cost estimation with formal methods based on real background data.

Recently, serious problems have happened in the Japanese software industry. For examples, the banking system of a mega bank formed by combining three major banks in Japan has made serious system troubles. A large number of cellular phones with system bugs were recalled, and the manufacturing companies paid for the huge loss.

Accordingly, many people in software industry in Japan begin to require controlled ways to construct reliable systems. Some of them become aware of formal methods and hope to solve the problems by applying them. Most managers in Japanese companies look for ready-to-use methods to solve their problems quickly. We, however, do not think that formal methods satisfy such quick requests. We must make enough efforts to learn how to utilize formal methods as effective tools to clarify and solve problems in system development.

We use the Japanese language in Japan. Therefore, the language problems are very serious in Japan. We need to use the Japanese language at every stage in system development including system modelling, analysis and specification. This means that we must provide at least two kinds of support to promote formal methods in Japan. Firstly, we need Japanese materials on formal methods such as textbooks and system documentation. Secondly, we must establish a practical way to express and discuss our ideas rigorously even in Japanese.

There have been published a number of English textbooks and case study reports on formal methods. However, there are few books on formal methods written in Japanese. A new introductory textbook on formal specification with VDM-SL will be published in the coming November [3]. This would be a first textbook on formal methods writen in Japanese. We are also translating Fitzgerald and Larsen's book on modelling systems [8] into Japanese. These two books will help us to learn formal methods much easier in Japan.

We must accumulate practice and experience with formal methods in real system development. We also need some effective guidelines for utilize formal methods in real practical system development. Bowen and Hinchey presented Ten Commandments of Formal Methods as such guidelines [5]. We would like to propose guidelines appropriate for Japanese companies.

We have a small but active community named SIG-FM (Special Interest Group on Formal Methods) in the Software Engineers Association of Japan (SEA) [12]. SIG-FM has been playing very imprtant roles in advocating formal methods and support technology transfer and sharing knowledge, experience, know-how, etc. It has a series of forums on formal methods from a practical viewpoint and is bridging between theoretical people and industrial people. Its members have been preparing tutorial textbooks on formal method in Japanese. They are also planning large-scale projects to apply formal methods to real problems and try to share their practice and experience.

4 Concluding Remarks

We have described some of case studies of applying formal methods to practical system development and the current state of formal methods in Japan. These case studies really convince us of practical effectiveness of formal methods. We intend to expand their applicability to a wide variety of application domains. We need more experience and efforts with applying formal methods to real practical problems.

References

1. Araki, K.: Are Formal Methods Relevant? — How to Explode the Seven Myths in Japan —, Proc. APSEC'95, pp.514-515, 1995.
2. Araki, K., Chang, H.-M. and Tanaka, T.: Case Studies of Formal Approaches to Domain Modelling and Specification, Proc. ISFST-2000, pp.213-218, 2000.
3. Araki, K. and Chang, H.-M.: Program Specification, Ohm-SHA Pub. Co., 2002. (to be published in Japanese)
4. Bowen, J. P. and Hinchey, M. G.: Seven More Myths of Formal Methods, IEEE Software, Vol. 12, No. 4, pp.34-41, 1995.
5. Bowen, J. P. and Hinchey, M. G.: Ten Commandments of Formal Methods, IEEE Computer, Vol. 28, No. 4, pp.56-63, 1995.
6. Bowen, J. P. and Hinchey, M. G.: High-Integrity System Specification and Design, Springer-Verlag, 1999.
7. Endo, T., Araki, K. and Chang, H.-M.: SYstematic Generation of Statechart from VDM-SL in Multi-Aspect Formal Methods for System Development, Proc. ISFST-2001, pp.9-14, 2001.
8. Fitzgerald, J. and Larsen, P. G.: Modelling Systems — Practical Tools and Techniques in Software Development —, Cambridge University Press, 1998. (to be translated into Japanese)
9. Hall, J. A.: Seven Myths of Formal Methods, IEEE Software, Vol. 7, No. 5, pp.11-19, 1990.
10. Ogino, T. and Hirao, Y.: Safety Technology and Formal Methods in Railway Signalling, Proc. Foundation of Software Engineering'99, KINDAI-KAGAKU-SHA, pp.2-6, 1999. (in Japanese)
11. Sahara, S.: Case Study of Application of Formal Specification to Stock Trading Systems, Proc. Software Symposium, Software Engineers Association of Japan, pp.27-34, 2001. (in Japanese)
12. SEA SIG-FM:
http://shinsahara.com/www.sigfm/
13. Someya, M., Kinoshita, H. and Tabata, F.: A Trial of Software Development through Formal Specification, Proc. Foundation of Software Engineering '99, KINDAI-KAGAKU-SHA, pp.8-12, 1999. (in Japanese)
14. Tanaka, T., Watanabe, Y., Araki, K. and Chang, H.-M.: A General Graph Analysis Tool with Attributed Path Query, Proc. ISFST-2001, pp.240-245, 2001.