Recent Industrial Applications of VDM in Japan

Peter Gorm Larsen
Engineering College of Aarhus
Dalgas Avenue 2, DK-8000 Aarhus C, Denmark
pgl@iha.dk

John Fitzgerald
School of Computing Science
Newcastle University, UK
John.Fitzgerald@ncl.ac.uk

Abstract

This paper describes the industrial use of the Vienna Development Method (VDM and VDM++) technology in Japan since the acquisition of VDMTools by CSK Systems in 2003. This acquisition followed a very successful application of VDM++ in the development of two subsystems of the TradeOne back office system for securities trading. Subsequently, FeliCa Networks has also successfully applied VDM++ in the development of a new generation IC chip for use as an electronic purse which can be embedded in a cellular telephone. This paper provides a short overview of some of the most important industrial applications of VDM in Japan in recent years. It also reports about the main lessons learned, particularly regarding quality and cost. Finally, the future prospects for this kind of technology in Japan are considered.

Keywords: Vienna Development Method, VDM, formal methods, industrial applications

1. INTRODUCTION

Most industrial applications of formal methods to date have taken place in Europe [1, 2, 3, 4] and US [5, 6]. This paper demonstrates that Japan has been moving forward strongly in this area in recent years. Many of the results have so far been published in Japanese, typically at the Japanese Software Symposium, so this paper aims to provide an English-language update on the recent achievements. Having worked with Japanese engineers for more than a decade makes us believe that we will see a substantial increase of the industrial take-up of formal methods in the coming years in Japan.

The work reported here exploits the Vienna Development Method (VDM) [7, 8, 9]. This is one of the longest established model-oriented formal methods, having originated in the IBM laboratories in Vienna at the beginning of the 1970s. VDM supports modelling and analysis at various levels of abstraction, using a combination of implicit and explicit definitions of functionality. VDM has a strong record of industrial application [3, 10, 11, 12, 13], in many cases by practitioners who are not specialists in the underlying formalism or logic. Experience from these projects suggests that the effort expended on formal modeling and analysis can be recovered in reduced cost of rework arising from design errors [14, 15]. The projects discussed in this paper support that result.

In the middle of the 1990s, cooperation began between the Railway Technical Research Institute (RTRI) [16, 17] and IFAD, the Danish company that had spent over a decade developing VDM technology to industrial standards [18]. This initial work used VDM-SL [8]. Correctness of railway systems has been a major goal for RTRI for a long time. Natsuki Terada, one of their engineers, spent two full years at IFAD in Denmark in order to take advantage of the proof support then being developed for VDM in the EU-supported PROSPER project [19].

CSK Systems (at that time Japan Future Information Technology & Systems Co., Ltd. (JFITS), a part of the CSK Group) decided to purchase the VDMTools technology from IFAD. JFITS had been using the

technology successfully on two of the subsystems of the TradeOne system, and saw the potential for further exploitation.

This paper provides a very brief review of the use of VDM++ in the TradeOne project (Section 2) and in a more recent and larger project carried out by FeliCa Networks (Section 3). In each case, we give a brief presentation of the application and the metrics that have been gathered. The paper ends with a few concluding remarks and predictions about future expectations.

2. THE TRADEONE PROJECT

TradeOne is a *back-office solution* aiming to lower the general operating costs in trading securities. A detailed description of the use of VDM++ in TradeOne has been provided elsewhere ([20], Chapter 11). Here we will give a flavour of the development and discuss some of the metrics from the VDM++ development.

2.1. A High-level Project Description

The users of the TradeOne system are securities companies and brokerage houses trading in securities. A *security* is a certificate attesting credit or the ownership of stocks, options, bonds, etc. An *option* is a contract that entitles its owner to either buy or sell a security or an index at a certain price before a certain date. The *dates* at which securities should be exercised, cancelled or agreed are critical to successful business in this domain. In trading companies, significant resources are therefore devoted to keeping track of the dates for securities. To remain competitive in a worldwide trade market, it is necessary to optimise the trader's business model and deal with the complexity of older traditional systems and TradeOne does that.

TradeOne's functionality covers several areas, shown informally in Figure 1. Two of these have been developed using the object-oriented extension of VDM (VDM++): the tax exemption subsystem and the option subsystem. The former automates the handling of Japanese tax regulations, previously a manual and error-prone task. The latter is responsible for handling the business process related to trading options. This kind of automated support is necessary to accommodate business process change, for example, to reduce the securities transaction settlement date.

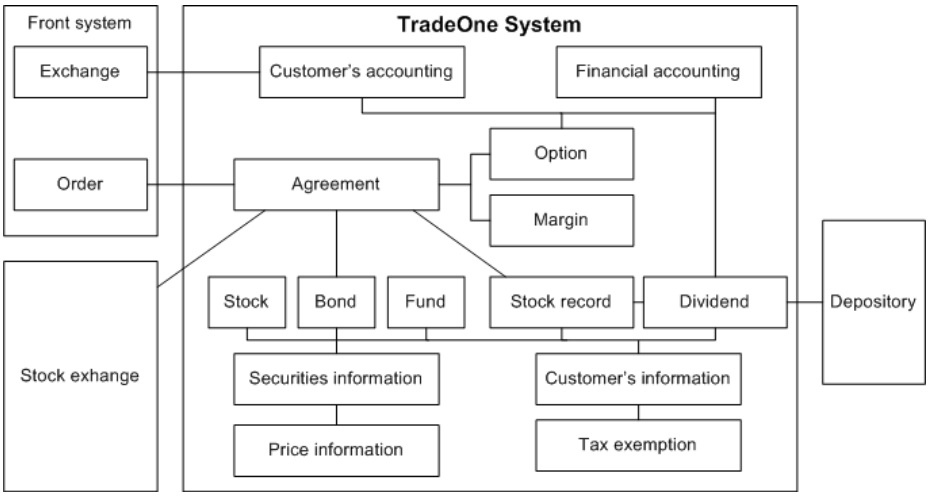

Figure 1: Overview of the TradeOne system

2.2. The Tax Exemption Subsystem

The tax exemption subsystem was developed by a small team of developers during 2000. There were six people in this development team, averaging four people per month for three and a half months. The development started with an initial design phase in which a system architect derived an initial framework for the VDM++ models. After this, four VDM++ experts developed models in parallel. All the VDM++ models

were reviewed by all the members of the project team. Finally two expert programmers implemented the system manually from the VDM++ model.

All the developers were over 40 years old and each had been developing software for more than 20 years. However, for all of them it was their first development using C++ and Java. For all but one the developers it was also their first application in the financial domain. Four members of the team had some basic prior knowledge of formal methods and so had little difficulty applying VDM++ on this subsystem. An external consultant was also used for the development of this subsystem.

2.3. The Option Subsystem

The option subsystem was developed by a larger team of ten developers during 2001. The development started with a group of domain experts writing functional requirements for the subsystem in Japanese. In parallel, one of the VDM experts from the tax exemption subsystem taught three new programmers VDM++ (in less than one week).

It is worth remarking that a Japanese version of VDMTOOLS supporting Japanese language identifiers was used in the option subsystem development. This was considered important because a larger group of people was involved in the VDM++ modelling. On average, 9.5 person-months were spent per calendar month.

2.4. Metrics Gathered

To permit comparisons between TradeOne and other projects, standard measuring principles based on the constructive cost model (COCOMO) [21] were followed. In COCOMO the size of an application can be measured in "delivered source instructions" (DSI). DSIs for the overall TradeOne system and the parts modelled using VDM++ are given in Table 1.

Table 1: Size of TradeOne

System	DSI (C++)
Total TradeOne	1,342,858
Tax exemption subsystem	18,431
Option subsystem	60,206

On the productivity side, the COCOMO estimates for man-months (MM) as well as duration are compared against the real effort spend and duration in Table 2.

Table 2: Productivity of TradeOne

Subsystem	COCOMO estimate	Real time	Time saving
Tax exception	Effort:38.5 MM Schedule:9 M	Effort:14 MM Schedule: 3.5 M	Effort:74% Schedule:61%
Option	Effort:147.2 MM Schedule:14.3 M	Effort: 60.1 MM Schedule:7 M	Effort: 60% Schedule: 51%

The corresponding VDM++ models were in total 11,757 DSI for the tax exemption subsystem and 68,170 DSI for the option subsystem. These figures include a test bed, test cases and informal comments explaining the VDM++ models. In total 3000 test cases were produced covering 100 % of the VDM++ model. In the option subsystem model more than 30,000 lines are dedicated to test cases alone.

From a quality perspective, the defect rates at integration test were 0.65/kDSI for the tax exemption subsystem was and 0.71 defects/kDSI for the option subsystem. These defects were readily fixed and are believed to have originated in the requirement gathering phase. No defects have been identified in the running implementations of these subsystems. The TradeOne product as a whole had a defect rate of 1.12 defects/kDSI at integration test and 0.67 defects/kDSI in the running implementation. This is in itself better than normal industrial standards. To compare to defect rates elsewhere, the order of defects in NASA software is reputed to be around around 0.1 defects/kDSI, and at least 10 times normal development costs are required to reach this level of correctness. For normal-quality commercial released code, the

figure is around 30 defects per 1000 DSI. This comparison suggests that the correctness reached in the two subsystems developed using VDM++ is impressive.

3. THE FELICA NETWORKS PROJECT

3.1. High-level Project Description

In the production of a new generation of IC chip for electronic payment with an electronic purse FeliCa Networks decided to use VDM++ for the development of the operating system software (the firmware inside the chip). This new generation IC chip contains new features and so is more complex than previous generations. The new generation IC chip must operate to the strict timing requirements provided by earlier generations despite the increase in complexity.

During the development of this product there have been 50 to 60 people affiliated with this project, with an average age of a little over 30 years. No members had knowledge of or experience with the formal method at the time of project launch. VDM training in Japanese was provided for the development team by CSK. In addition an external VDM consultant from CSK Systems was used throughout the project. This version of the firmware took about three years to develop and it was delivered on schedule and millions of IC chips have been distributed.

3.2. Project Teams and Metrics Gathered

The development team was divided into groups: one responsible for validation and verification, and one for implementation. Subsequently the members of each group completed questionnaires assessing their impression of the use of VDM++. In addition an interview was held with the twelve team members who most frequently referred to the formal VDM++ model. This feedback showed that none of the groups had negative reactions to using VDM++ but the members from the implementation team did not see the benefit of it as much as the others. VDM++ was felt to be an efficient communication tool between the team members and between the teams. They all acknowledged that new knowledge was required but also that there is a substantial difference between the skills required in writing and reading a formal model. In general it was felt that advanced mathematics was not necessary for writing or validation the VDM++ model.

The main results related to specification development were:

- A 383 page protocol manual written in natural language (Japanese). This is a manual for other departments within the company as well as for outside customers.
- A 677 page external specification document written in VDM++ (approximately 100 kDSI including comments). Approximately 60 % of that can be considered as test cases formulated in VDM.

The specification includes the specification of 86 different commands in the firmware and the file system security specification. From this VDM++ model a C/C++ code implementation of approximately 110 kDSI, including comments, was implemented by hand as firmware for a single IC chip.

The validation team developed a large collection of VDM++ test cases that were executed using the VDMTools interpreter. Using the VDMTools test coverage analysis facility, it was possible to display test coverage information on the VDM++ model after executing the entire test suite. Here 82 % of the VDM++ model was covered, and the remaining parts of the model were manually inspected. In order to support this development with an extremely high number of test cases, the speed of the interpreter was improved by CSK Systems by more than a factor of 100.

From a quality perspective more errors were found in the early phases of the development than in other similar projects at FeliCa Networks. In total 440 defects were detected in the requirements and the specifications. Of these, 278 were directly a result of the use of VDM++. Of these, 162 were found by review of the model whereas 116 were discovered using the VDMTools interpreter with test cases against the executable VDM++ model. It is important also to note that, at the time of writing, no defect has been discovered since release.

4. CONCLUDING REMARKS AND FUTURE EXPECTATIONS

The figures derived using the COCOMO model productivity metrics in the TradeOne project are very impressive. From a cost perspective these results are outstanding from a first project using formal methods, in particular since most of the people involved had limited practical prior knowledge of formal techniques. In addition, the metrics for defect rates are also very encouraging. In the FeliCa Networks project it is felt that the first application using VDM++ did not change the overall cost expectations. However, the period for the specification development became longer than in any similar project in FeliCa Networks. This increased level of effort in early design stages is widely anticipated when formal methods are employed. It is important to manage the expectations of project management in this regard. FeliCa Networks feels that their main advantage in this project has been the improved quality and they expect that long-term productivity gains will follow later.

Communication is the key to software development (in particular between specification, implementation and validation teams), and VDM++ appears to have become a valuable communication tool. It is important that management, project owners and project members are fully aware of this and that they understand that it is not a method that will replace existing methods but rather one that needs to be used to complement other techniques such as UML. Finally, it is important that various methods and improvements are combined and established as part of the corporate and organizational culture. Thus, the next generation of the FeliCa Networks firmware is also currently being developed using VDM++. This is done primarily because of the improvements in communication between the different teams as well as the high quality resulting from this approach.

Several characteristics of the VDM technology have, we believe, contributed to its success in recent applications. These include the emphasis placed on developing robust tools which are open to linkages with existing tool sets supporting, for example, UML design. We have also focussed on analysis approaches, such as the use of test scenarios and test coverage, that provide a route into the use of a formalism without presenting a high initial hurdle to users. Our expectation is that this will lead on the use of tools supporting more advanced static analysis, proof support and test generation.

Significant extensions to VDMTools have been made in the area of distributed, embedded and real-time systems [22, 23, 24, 25]. With these in mind, it is expected that we will see more new applications in in domains such as automotive systems in Japan in the future.

Acknowledgments

We are grateful to Miki Chiba, Taro Kurita, Yasumasa Nakatsugawa and Shin Sahara for allowing us to write about these projects, and for their valuable comments on drafts of this material. In addition we would like to thank Hugo Macedo, Rintaro Suehiro and Marcel Verhoef for their help with translation and with drafting this paper. Finally, we acknowledge the support of the EU Integrated Project on Industrial Deployment of System Engineering Methods providing high Dependability and Productivity (www.project-deploy.eu).

Bibliography

[1] Nix, C., Collins, B.: The use of Software Engineering, including the Z notation, in the development of CICS. Quality Assurance **14** (1988) 108

[2] Hinchey, M.G., Bowen, J.P., eds.: Applications of Formal Methods. Prentice Hall (1995) ISBN 0-13-366949-1.

[3] Larsen, P.G., Fitzgerald, J., Brookes, T.: Applying Formal Specification in Industry. IEEE Software **13**(3) (May 1996) 48–56

[4] Hall, A.: Using formal methods to develop an atc information system. IEEE Software **12**(6) (March 1996) 66–76

[5] Craigen, D., Gerhart, S., Ralston, T.: Formal Methods Reality Check: Industrial Usage. IEEE Transactions on Software Engineering **21**(2) (February 1995) 90–98

[6] Miller, S.P., Tribble, A.C., Whalen, M.W., Heimdahl, M.: Proving the shalls: Early validation of requirements through formal methods. International Journal on Software Tools for Technology **8**(4/5) (August 2006) 303–319

[7] Bjørner, D., Jones, C., eds.: The Vienna Development Method: The Meta-Language. Volume 61 of Lecture Notes in Computer Science. Springer-Verlag (1978)
[8] Fitzgerald, J., Larsen, P.G.: Modelling Systems – Practical Tools and Techniques in Software Development. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK (1998) ISBN 0-521-62348-0.
[9] Overture Group: The VDM Portal. *http://www.vdmportal.org* (2007)
[10] Brookes, T., Fitzgerald, J., Larsen, P.: Formal and Informal Specifications of a secure System Component: Final Results in a Comparative Study. In Gaudel, M.C., Woodcock, J., eds.: FME'96: Industrial Benefit and Advances in Formal Methods, Springer-Verlag (March 1996) 214–227
[11] Clement, T., Cottam, I., Froome, P., Jones, C.: The development of a commercial "shrink-wrapped application to safety integrity level 2: the dust-expert story. In: Safecomp'99, Toulouse, France, Springer Verlag (September 1999) LNCS 1698, ISBN 3-540-66488-2.
[12] Smith, P.R., Larsen, P.G.: Applications of VDM in Banknote Processing. In Fitzgerald, J.S., Larsen, P.G., eds.: VDM in Practice: Proc. First VDM Workshop 1999. (September 1999) Available at www.vdmportal.org.
[13] Puccetti, A., Tixadou, J.Y.: Application of VDM-SL to the Development of the SPOT4 Programming Messages Generator. In Fitzgerald, J., Larsen, P.G., eds.: VDM in Practice. (September 1999) 127–137
[14] Fitzgerald, J.S., Larsen, P.G.: Triumphs and Challenges for the Industrial Application of Model-Oriented Formal Methods. In Margaria, T., Philippou, A., Steffen, B., eds.: Proc. 2nd Intl. Symp. on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2007). (2007) Also Technical Report CS-TR-999, School of Computing Science, Newcastle University.
[15] Fitzgerald, J.S., Larsen, P.G.: Balancing Insight and Effort: the Industrial Uptake of Formal Methods. In Jones, C.B., Liu, Z., Woodcock, J., eds.: Formal Methods and Hybrid Real-Time Systems, Essays in Honour of Dines Bjørner and Chaochen Zhou on the Occasion of Their 70th Birthdays, Volume 4700, Springer, Lecture Notes in Computer Science (September 2007) 237–254 ISBN 978-3-540-75220-2.
[16] Ogino, T., Hirao, Y.: Formal methods and their applications to safety-critical systems of railways. QR of RTRI **36**(4) (December 1995) 198–203
[17] Terada, N., Fukuda, M.: Application of Formal Methods to the Railway Signaling Systems. Quarterly Report of RTRI **43**(4) (2002) 169–174
[18] Larsen, P.G.: Ten Years of Historical Development: "Bootstrapping" VDMTools. Journal of Universal Computer Science **7**(8) (2001) 692–709
[19] Dennis, L.A., Collins, G., Norrish, M., Boulton, R., Slind, K., Robinson, G., Gordon, M., Melham, T.: The PROSPER Toolkit. In: Proceedings of the 6th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Berlin, Germany, Springer-Verlag, Lecture Notes in Computer Science volume 1785 (March/April 2000)
[20] Fitzgerald, J., Larsen, P.G., Mukherjee, P., Plat, N., Verhoef, M.: Validated Designs for Object–oriented Systems. Springer, New York (2005)
[21] Boehm, B.: Software Engineering Economics. Prentice-Hall International (1981)
[22] Verhoef, M., Larsen, P.G., Hooman, J.: Modeling and Validating Distributed Embedded Real-Time Systems with VDM++. In Misra, J., Nipkow, T., Sekerinski, E., eds.: FM 2006: Formal Methods, Lecture Notes in Computer Science 4085 (2006) 147–162
[23] Verhoef, M., Larsen, P.G.: Interpreting Distributed System Architectures Using VDM++ – A Case Study. In Sauser, B., Muller, G., eds.: 5th Annual Conference on Systems Engineering Research. (March 2007) Available at http://www.stevens.edu/engineering/cser/.
[24] Verhoef, M., Visser, P., Hooman, J., Broenink, J.: Co-simulation of Real-time Embedded Control Systems. In Davies, J., Gibbons, J., eds.: Integrated Formal Methods: Proc. 6th. Intl. Conference. Lecture Notes in Computer Science 4591, Springer-Verlag (July 2007) 639–658
[25] Fitzgerald, J.S., Larsen, P.G., Tjell, S., Verhoef, M.: Validation Support for Real-Time Embedded Systems in VDM++. In Cukic, B., Dong, J., eds.: Proc. HASE 2007: 10th IEEE High Assurance Systems Engineering Symposium, IEEE (November 2007) 331–340