National Institute of
Advanced Industrial Science
and Technology
AIST

**AIST-PS-2008-013**

# Model checking education for software engineers in Japan (Preliminary version)

Hideaki Nishihara[1], Koichi Shinozaki[2], Koji Hayamizu[3],
Toshiaki Aoki[4], Kenji Taguchi[5], Fumihiro Kumeno[5],[6]

[1]Research Center for Verification and Semantics,
National Institute of Advanced Industrial Science and Technology(AIST)
[2]The Kansai Electric Power Co.
[3]Melco Power Systems Co.
[4]Research Center for Trustworthy e-Society
Japan Advanced Institute of Science and Technology
[5]Grace Center, Information Systems Architecture Research Division,
National Institute of Informatics

[6]Mitsubishi Research Institute

算譜科学研究速報
Programming Science
Technical Report

Research Center
for Verification and Semantics

独立行政法人 産業技術総合研究所 システム検証研究センター 発行
Published by Research Center for Verification and Semantics
National Institute of Advanced Industrial Science and Technology

# Model Checking Education for Software Engineers in Japan

Hideaki Nishihara[1], Koichi Shinozaki[2], Koji Hayamizu[3], Toshiaki Aoki[4], Kenji Taguchi[5], Fumihiro Kumeno[56]

[1] Research Center for Verification and Semantics
National Institute of Advanced Industrial Science and Technology(AIST)
[2] The Kansai Electric Power Co., Inc.
[3] Melco Power Systems Co.
[4] Research Center for Trustworthy e-Society
Japan Advanced Institute of Science and Technology
[5] Grace Center, Information Systems Architecture Research Division,
National Institute of Informatics
[6] Mitsubishi Research Institute

**Abstract.** This paper is the preliminary report of a joint research project on advocacy of a Body of Knowledge on Model Checking being carried out by six organizations which deliver model checking courses to software engineers in Japan. In this paper we will explain the main objective of the project and report the evaluation results of our model checking programs.

## 1 Introduction

Model checking has been used as a verification methodology for hardware and software systems and taught in computer science and software engineering curricula mostly at higher education. On the other hand, our organizations, National Institute of Advanced Industrial Science and Technology (AIST), Melco Power Systems Co. (MPS), Japan Advanced Institute of Science and Technology (JAIST) and National Institute of Informatics (TOPSE) have been delivering various courses on model checking to software engineering for over years. Our programs have very strong focus on practical aspects on model checking, which make ours different from those in academia.

Formal methods have been recognized as a rigorous software development methodology for safety critical systems and it is now recommended to be used in safety and security areas such as functional safety (IEC 61508) [2] and security assurance (ISO/IEC 15408) [1]. Particularly model checking is attracting software engineers as a rigorous verification methodology for the system development and we have been meeting their demands by providing educational courses, consutancy works and system developments using this technology.

We started our joint research project to standardize courses on model checking and soon realized that model checking education was still not mature yet due to lack of curriculum guidance based on a description of knowledge, which covers the whole area of model checking. SWEBOK standardized by IEEE CS and ACM for software engineering education [3] is too broad and model checking is only referred in Validation and Verification so that it is largely insufficient for our purpose. This observation motivated us to work on the Body of Knowledge on Model Checking (MCBOK). A precursor of this kind of BOK in Formal methods could be found in a work by Oliveira [8]. He presented a survey on the undergraduate curricula on formal methods as a part of FME-SoE (Formal Methods Europe, subgroup on Education). The paper shows a wide variety of formal methods courses in Europe, but does not present a well structured body of knowledge on formal methods. It is unfortunate that there has not been any follow-up activity from this group since then, even the demand for education of formal methods is becoming more important than it has ever been.

In this paper, we will report the first result of our joint project, i. e., evaluation of each courses based on student feedback. The paper is organized as follows. The next section briefly explains each course delivered by AIST, MPS, JAIST and TOPSE. Section 3 presents evaluation results based on questionnaires taken by each course, and in Section 4, we will conclude the paper.

## 2 Model Checking Courses

This section briefly explains four courses on model checking delivered by each organisation.

### 2.1 AIST

CVS/AIST (Research Center for Verification and Semantics, National Institute of Advanced Industrial Science and Technology) aims to encourage Japanese industry to adopt Formal Methods as standard verification methods and CVS/AIST has experiences in applying model checking in industrial fields. As a direct way to contribute the aim, we are developing a model checking training course series for engineers to transfer technologies.

Courses in the series aim to give principles and experiences of model checking. Many parts of courses are based on examples, and some of them are modified materials in our experiences. In the courses tool-dependent descriptions and knowledge are avoided, thus every example in the courses is checked by plural tools (precisely they are NuSMV [7] and SPIN [4]).

The series consists of the following three courses: the elementary course, the intermediate course, and the advanced course. The elementary course gives an overview of model checking and a skill to execute model checking procedures. The intermediate course gives some typical techniques in model checking: behaviours in some kinds of products of models and abstraction of models. The advanced course deals with some search algorithms to give knowledge about efficient verifications.

These courses take three or four days, in order to let students digest the contents in the course itself, especially examples and exercises.

### 2.2 JAIST

Recently, keywords 'formal methods and model checking' are attracting Japanese industry. However the details of those technologies are not known well, and the keywords alone are spreading in the industry. JAIST recognizes that providing such information about advanced technologies to them is one of its mission. This is our motivation to hold seminars. Accordingly, the objective of the seminar is that its participants become to identify whether those technologies are userful in their fields or not.

Though there are many model checking tools, we focus on one of them, SPIN model checker [4], in the seminar. This is because concurrent processes of SPIN are similar to multi-tasks of RTOS(Real-Time Operating Systems) which is used in embedded software. Moreover, the syntax of Promela which is the input language of SPIN is similar to that of imperative languages like C. We think that those facts make it easier to learn model checking technologies for engineers in the industry.

Target participants of our seminar are engineers in the filed of embedded systems. We think that it is important to show successful examples using technical terms appearing in embedded system developments. Thus, we show examples from the fileds of system programming and protocol, for instance, mutual exclusion, scheduling and alternating bit protocol. These ones are already known as the model checking effectively works. The number of the examples is about 90, and their total lines of code is about 4000.

### 2.3 TOPSE

The Top SE program is a non-accredited course at Masters level fully funded by the government and is operated through a close collaboration between industry and academia at the National Institute of Informatics. The overview and the curriculum design of the whole program are discussed in [5]. We deliver the following five courses on model checking: Model Checking Foundations and Applications, Real-time Model checking, Software Model checking and Modelling and verification of concurrent models. These courses focus on efficient use and proper application of model checking tools that use the automata theory, specifically SPIN [4], SMV [7], LTSA [6] and FDR. The key learning objectives of the Foundations and Applications courses are to learn how to detect and correct faults in design specifications in UML state machine models. More detailed explanation was presented in [5].

### 2.4 MPS

MPS has an in-house curriculum which aims to teach practical techniques of model checking.We especially focus on the practical side, and thus we put a higher priority on the practical techniques than on the theory in the curriculum. The objective of the curriculum is to give such a skill that a participant can apply model checking to actual software development immediately.

The curriculum consists of the following three courses: the basic course, the application course, and the practice course. In the basic course, temporal logic and state transition systems are explained with lectures and practices. The application course and the practice course are mainly composed of practical exercises that are verifications of flow charts and source codes. Especially, in the practice course, the course materials are the softwares developed by the participants themselves. We use the model checking tool NuSMV [7] in our curriculum. In addition, in the application course and the practice course, we use a GUI tool for NuSMV "Support Software for Model Checking" which was jointly developed by KEPCO and MPS. It is our attempt at making the hurdle lower as much as possible to applying model checking to actual software development, by reducing tasks in using it.

The curriculum is used in the situation that the participants and the lecturers are both enginieers in a company. Therefore our courses not only give some knowledges or procedures, but also bring up practical engineers of model checking.

## 3 Evaluation and Observation of Questionnaires

In this section, we will summarize the results of questionnaires carried out by four organizations in terms of understandability, usefulness and feasibility in Table 1.

As the participants of the courses are software engineers, the questionnaires address practical issues, e. g., the degree of understanding of the tools and techniques taught, the usefulness of the tools and techniques to students' own problems and feasibility of the techniques to be used in the industrial context.

Questionnaires were taken before we started working on this joint project so that we used different questionnaire forms, however, some questions are shared by them. Table 1 shows the result of their shared and fundamental questions. Each question has four or five choices which represent the degree of goodness and badness, and the number of the positive answers among them are only shown, that is, the second or the third choice from the best one. The understandability, usefulness and feasibility stand for the number of the participants who could understand the contents of each course, the participants who feel that the model checking is useful in their fields, and the participants who feel that the model checking can be practically feasible in their fields respectively.

On understandability, most of the courses mark high scores. the AIST intermediate course takes lower points since the contents were rather theoretical compared with other courses. In addition, it must be noted that the results obtained in the TOPSE are somewhat different from other courses due to the fact that participants of the program come to learn not only model checking courses but also some other courses such as software architecuture and requirements engineering. Even taking these backgrounds of each course into account, we can observe that model checking is not so difficult for engineers to learn. The scores of the usefulness and the feasibility have also high similarily to the

**Table 1.** Questionnaire Results

| Courses | Students | Understandability | Usefulness | Feasibility |
|---|---|---|---|---|
| AIST (elementary) | 75 | 59(79%) | 70 (93%) | N/A |
| AIST (intermediate) | 23 | 7 (29%) | 24 (100%) | 19 (79%) |
| TOPSE (foundations) | 36 | 22 (61%) | N/A | 25 (69%) |
| TOPSE (applications) | 7 | 5 (71%) | N/A | 5 (71%) |
| JAIST | 59 | 56 (92%) | 56 (92%) | 49 (80%) |
| MPS (in-house training) | 13 | 13 (100%) | N/A | 8 (62%) |
| Total | 213 | 162/214 (76%) | 150/158 (95%) | 106/139 (77%) |

understandability. These results show that the Japanese industry has a potential to appreciate model checking, and that model checking is accepted as a practical method in Japanese industry.

We would like to quote some notable comments from the free description of the questionnaires. Many of the participants were surprised at the analysis power of the model checking tools as they know that concurrent and non-deterministic behaviour of the systems is very hard to analyze by hand. On the other hand, some participants were curious about how the model checking tools are integrated into their own system development processes. Unfortunately, we do not have suitable answers for this comment right now because the software processes to apply the model checking tools are not established yet. One way to meet this request is that we show successful examples to them. Making such examples for the education is one of our future works.

These results imply that teaching the model checking technologies to the engineers are fruitful. They were convinced of the relevance to learn the usage and application of the tools. On the other hand, they did not feel that to learn the theory of the model checking even though the theory exists behind the tools. They might be just interested in how the model checking tools are taken into their daily works. However, we still believe that teaching the tools to the engineers is important. Though, right now, their interest is the usage of the tools, the interest will be extended to advanced topics like the theory and principle around the tools. The model checking has the limitations in some senses such as state explosion problems and descriptive power. Those limitations would be the motivation to learn technologies which are complement with the model checking and more advanced issues.

In summary, the overall score of feedback from the participants are very positive. We can conclude that our courses are well accepted by our participants. We hope that both the industries and academia tackle practical problems not only by taking engineering practices but also scientific approaches into their education programs.

## 4 Future Direction

As described the previous section, our past activities show participants' interests and their understanding for model checking. Our courses explained in Chapter 2 have no or a few vacant seats every time, and indeed we have over 200 respondents of questionnaires. We can consider that model checking is a noteworthy technology in Japan. It had not been thought that applying model checking to actual software developments was realistic, but questionnaire results show us it is not true now. There are many participants who comment that they want to apply model checking to actual software development with enough knowledges. We can see from the fact that many people in Japanese industry are understanding applicability of model cheking and want to intorduce it to software development processes.

On the other hand, appropriately considered education programs are necessary to learn model checking. Such programs should include theoretical backgrounds, since model checking is based on mathematics and logics. Moreover they should include how to use tools efficiently and successful examples in which model checking is applied to software development processes, since Japanese industry is sure to want them. But now every educational organization develops curricula and materials in its own purpose. A sufficiently adaptable and tidy curriculum is needed that considers both of industry's wants about usage and applications, and theories dealt with at appropriate levels.

In the state described above, in order to let model checking spread in industry, it is necessary to prepare an education program for software engineers, working as a reference. As a collabolation with industry and accademia, we plan the following projects for the program:

1. making an MCBOK,
2. making a reference curriculum, and
3. making a system to authorizing the skill.

Good curricula should be based on wide knowledge and practices in the subject, including theories, applications, and successful examples. These knowledge have not been summarized yet, and thus we will make a model checking body of knowledge (MCBOK) at first. The next is to make a curriculum based on the MCBOK. With MCBOK, one can construct various appropriate curriculums on model checking and teach it as to various needs and requirements in industry. Our curriculum will be for software engineers, and it will be a reference. It will make possible to compare several curricula (including our own ones) strictly, by mapping them to the reference curriculum. It will also make it possible to cooperate in educations or training in model checking among educational organizations. The MCBOK and the reference curriculum will make a skill in model checking clear, and moreover engineers having a skill in model checking will be authorized by them.This authorization will encourage software engineers to learn about model checking, and will contribute to standardization of model checking for industry.

## 5 Acknowledgments

The Top SE program is fully supported by the Special Coordination Funds for Promoting Science and Technology, Fostering talent in emerging research field by Ministry of Education, Culture, Sports, Science and Technology, Japan.

We thank Hiroki Takamura and Satoru Yoshida for valuable comments.

## References

1. ISO/IEC 15408. *Information technology - Security techniques - Evaluation criteria for IT security - Part1, Part2 and Part3.* ISO/IEC, 2005.
2. IEC 61508. *Functional safety of electrical/electronic/programmable electronic safety-related systems.* Bureau Central de la Commission Electrotechnique International, 2000.
3. A. Abran, J. W. Moore, P. Bourque, and R. Dupuis. *Guide to the Software Engineering Body of Knowledge 2004 Version SWEBOK.* IEEE, 2004.
4. G. J. Holzmann. *The SPIN model checker: Primer and reference manual.* Addison Wesley, 2004.
5. Shinichi Honiden, Yasuyuki Tahara, Nobukazu Yoshioka, Kenji Taguchi, and Hironori Washizaki. Top se: Educating superarchitects who can apply software engineering tools to practical development in japan. In *ICSE*, pages 708–718. IEEE Computer Society, 2007.
6. J. Magee and J. Kramer. *Concurrency: State Models & Java Programs, Second Edition.* John Wiley & Sons, 2006.
7. K. L. McMillan. *Symbolic Model Checking.* Kluwer Academic Publishers, Norwell, MA, USA, 1993.
8. José Nuno Oliveira. A survey of formal methods courses in european higher education. In C. Neville Dean and Raymond T. Boute, editors, *TFM*, volume 3294 of *Lecture Notes in Computer Science*, pages 235–248. Springer, 2004.

日本における技術者向けモデル検査の教育について (in English)
(算譜科学研究速報)
発行日：2008 年 8 月 28 日
編集・発行：独立行政法人 産業技術総合研究所 (システム検証研究センター)
同連絡先：〒560-0083 大阪府豊中市新千里西町 1-2-14 三井住友海上千里ビル 5F
TEL：06-4863-5025
e-mail：`informatics-inquiry@m.aist.go.jp`

Model checking education for software engineers in Japan (Preliminary version)
(Programming Science Technical Report)
Aug. 28, 2008
(Research Center for Verification and Semantics (CVS))
National Institute of Advanced Industrial Science and Technology (AIST)
5F Mitsui Sumitomo Kaijo Senri Bldg., 1-2-14, Shinsenrinishi-machi, Toyonaka,
Osaka 560-0083 Japan
TEL：+81-6-4863-5025
e-mail：`informatics-inquiry@m.aist.go.jp`