What Top-Level Software Engineers Tackle after Learning Formal Methods: Experiences from the Top SE Project

Fuyuki Ishikawa, Kenji Taguchi,
Nobukazu Yoshioka, Shinichi Honiden
GRACE Center/TopSE Project
National Institute of Informatics, Japan

TOC

- Report educational activities on FM for engineers in the industry
  - Overview of the Top SE Project
  - Lecture Courses
  - Graduation Studies
  - Statistics and Discussion

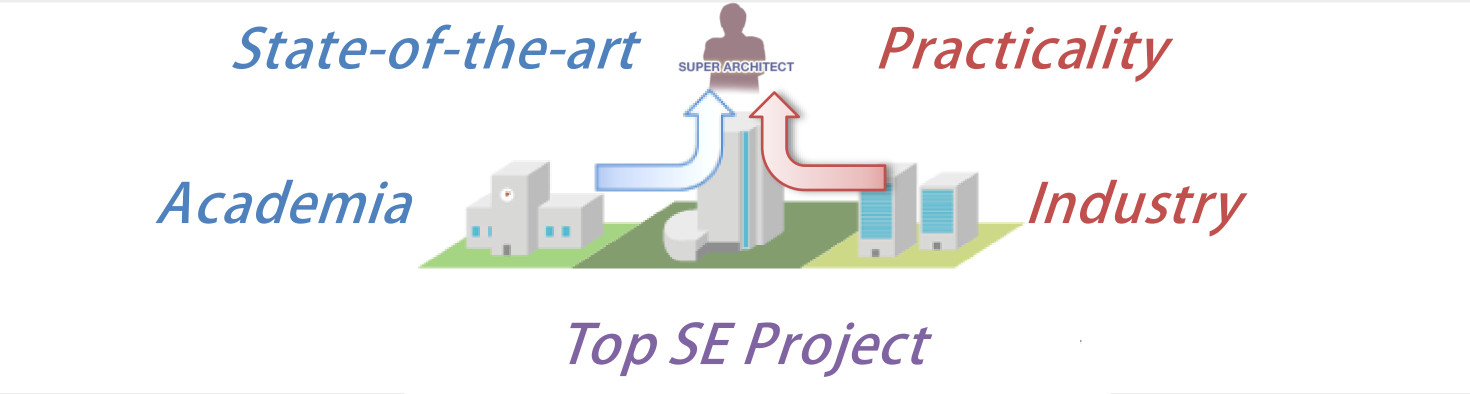
Top SE Project: Background/Motivation

Background

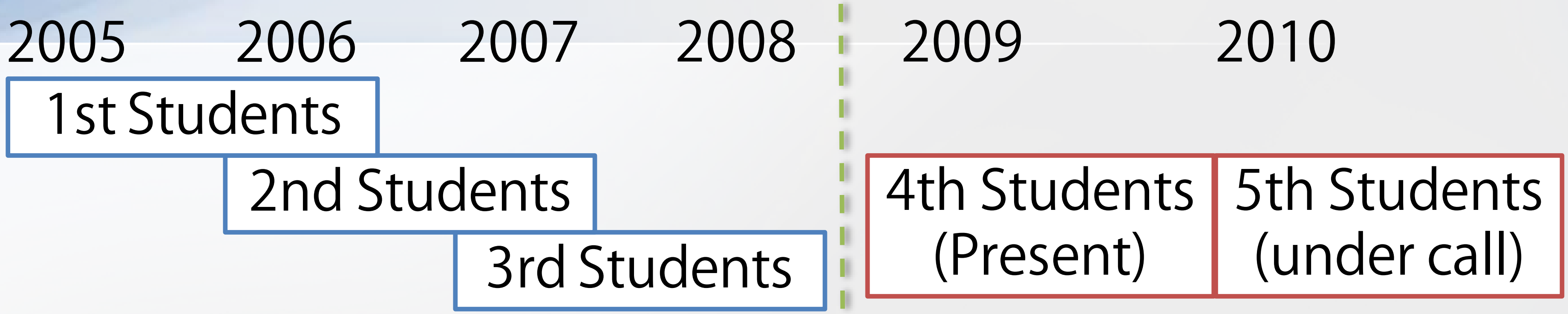
- Gaps between academia and industry regarding efficient and reliable approaches for SE (e.g., FM)

Objective & Approach

- Produce top-level software engineers by introducing scientific approaches (in academia) into industry
- Let academic/industrial experts jointly develop and provide an educational program



Top SE Project: History and Present Status



2005 2006 2007 2008 2009 2010

1st Students

2nd Students

3rd Students

4th Students (Present)

5th Students (under call)

Gradual Development

- 30 students per year
- 20 lecture courses
- 25 lecturers

  (15 from academia, 10 from industry)

(on average)

Government-Funded Set-up

(Free program for 1.5 year)

Renewal and Sustainable Operation

(Fee-paying program
for 1 year,
about $5,000/student)

Top SE Project: Lecture Courses

Series	Lecture Courses
Foundations (2)	Foundations in mathematical theory, Foundations in practical SE
Architecture (3)	Component-based development, Software patterns, Aspect-orientation
Formal Specification (3)	*Foundations, Applications, and Security aspects*
Model Checking (4)	*Foundations, Applications, Concurrency aspects, Real-time aspects*
Requirements Analysis (4)	Goal-oriented analysis, Elicitation and Identification, Security aspects, Early analysis
Implementation Techniques (3)	Testing, *Program analysis, Verification of implementation models*
Management (2)	Metrics, Development management

Features in the Program

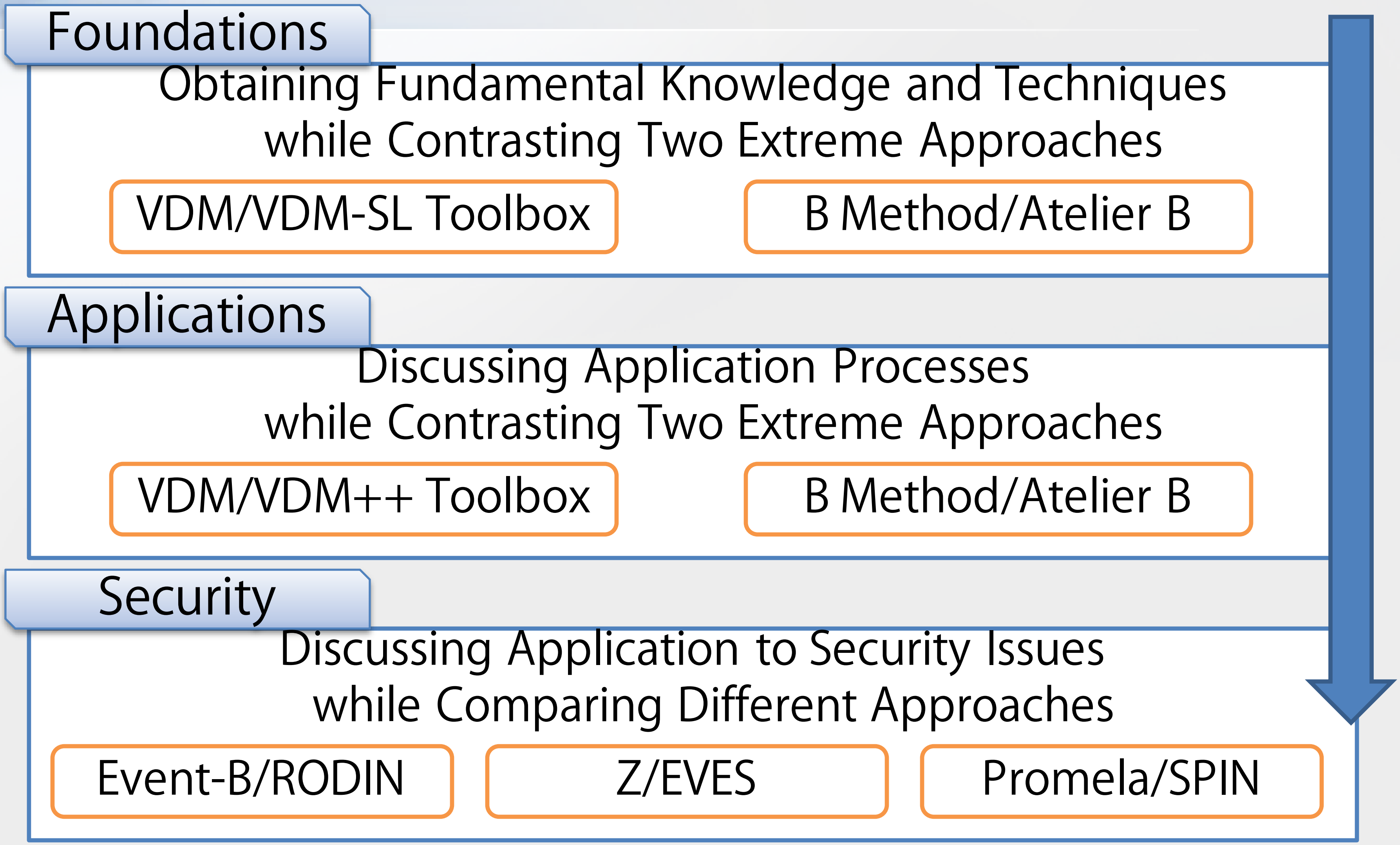
- Lecture courses (1.5h * 15 per course)
  - Learn different methods/tools in each area to see common principles and different strategies
  - Have group exercises to discuss how to apply the methods/tools using real application examples
- Graduation study (3 month -)
  - Tackle problems identified by themselves
    - Problems in their projects
    - Problems in applying learnt methods/tools

  (with lecturers as supervisors)
- Successive PhD work at a graduate univ.

TOC

- Report educational activities for engineers from the industry
  - Overview of the Top SE Project
  - <u>**Lecture Courses**</u>
  - Graduation Studies
  - Statistics and Discussion

Formal Specification Series



Foundations

Obtaining Fundamental Knowledge and Techniques while Contrasting Two Extreme Approaches

VDM/VDM-SL Toolbox

B Method/Atelier B

Applications

Discussing Application Processes while Contrasting Two Extreme Approaches

VDM/VDM++ Toolbox

B Method/Atelier B

Security

Discussing Application to Security Issues while Comparing Different Approaches

Event-B/RODIN

Z/EVES

Promela/SPIN

Model Checking Series

Foundations

Obtaining Fundamental Knowledge and Techniques

SPIN

Applications

Discussing Application Processes
with Comparing Different Tools

SPIN

SMV

LTSA

Concurrency

Discussing Difficulties in Verification and
Implementation with Concurrency

CSP (FDR, JCSP)

Performance

Obtaining Fundamental Knowledge and Techniques
and Discussing Application Processes with Timed Models

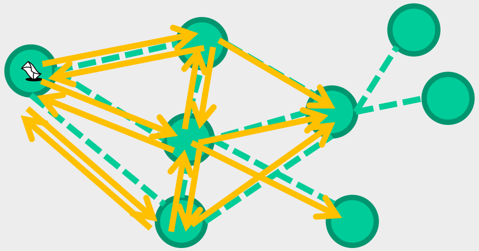
UPPAAL

Implementation Techniques Series

- Complements by introducing methods/tools on source codes
  - Program Analysis Course
    - JML
  - Verification of Implementation Models
    - Java PathFinder

Group Exercise: Example of VDM and B

- Group exercises for VDM and B
  - Formalize and validate a real, complex standard specification written in natural languages
    - Use small parts of OLSR, a standard protocol for routing management in ad-hoc networks
  - Discuss modeling/validation strategies
    - What to model? (or what to abstract away?)
    - What properties to check?
    - What ambiguities need to be resolved?



- Share information on the topology
- Choose nodes that forward messages for complete but more efficient multicasting

TOC

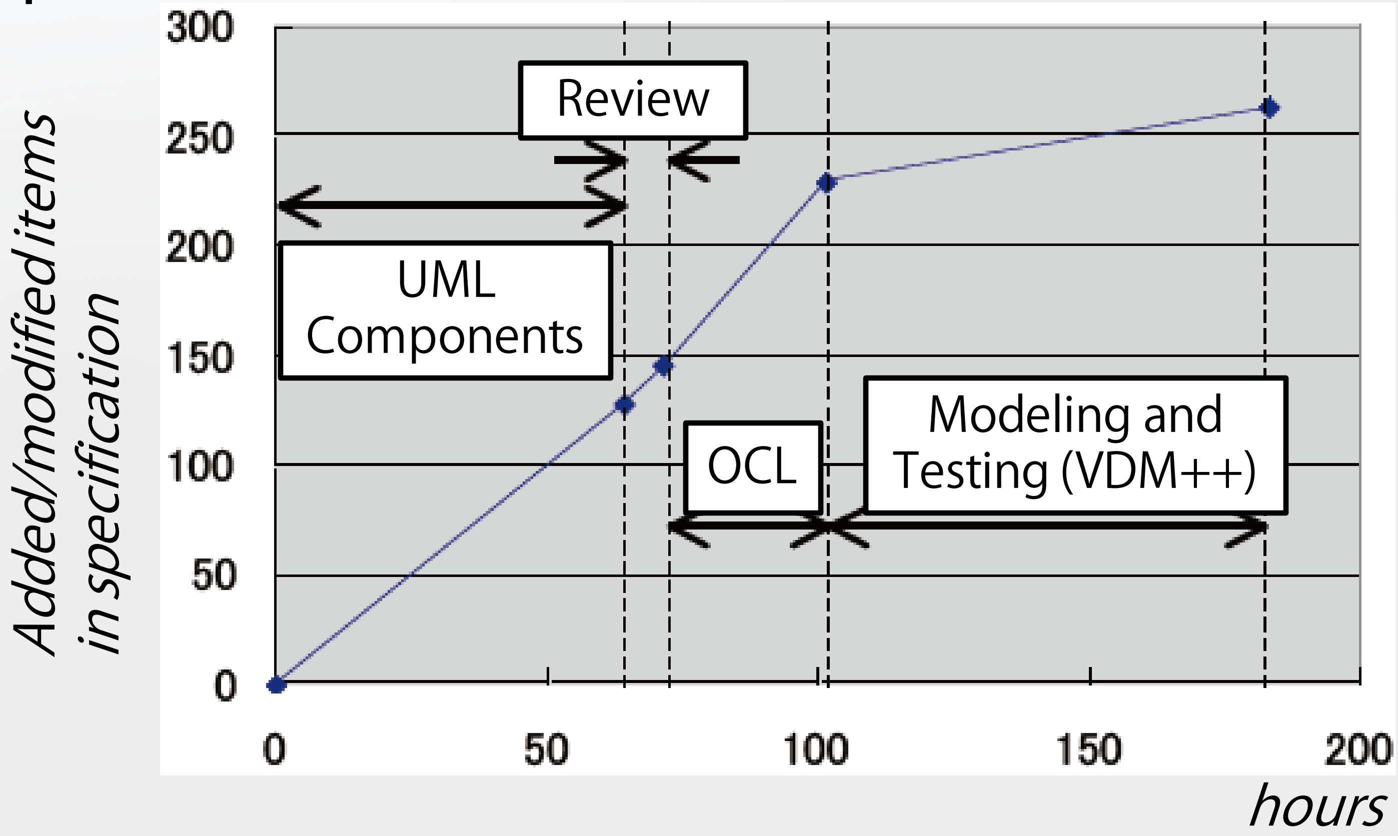
- Report educational activities for engineers from the industry
  - Overview of the Top SE Project
  - Lecture Courses
  - **Graduation Studies**
  - Statistics and Discussion

Types of Graduation Studies

- Case study
  - Tackle problems in a certain project by choosing and applying learnt methods/tools
- Domain-specific finer-grained support
  - Tackle problems in applying learnt methods/tools by developing domain-specific methods/tools
- Bridging gaps between methods/tools
  - Tackle problems in connecting different methods/tools by developing methods/tools
- Extension of methods/tools
  - Tackle problems in learnt methods/tools by extending them

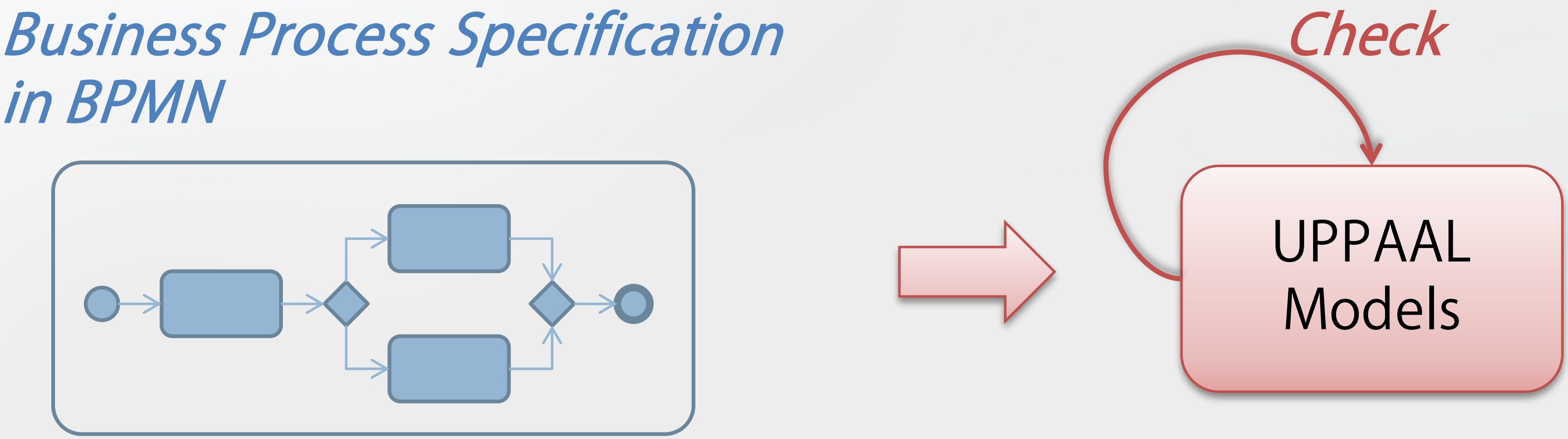
Examples of Graduation Studies (1)

- *Case-study type*: Run an experimental project and evaluate effects of introducing formal specifications



Examples of Graduation Studies (2)

- *Domain-specific support type*: Develop a tool to verify business processes with real-time properties considering resource constraints


Business Process Specification in BPMN

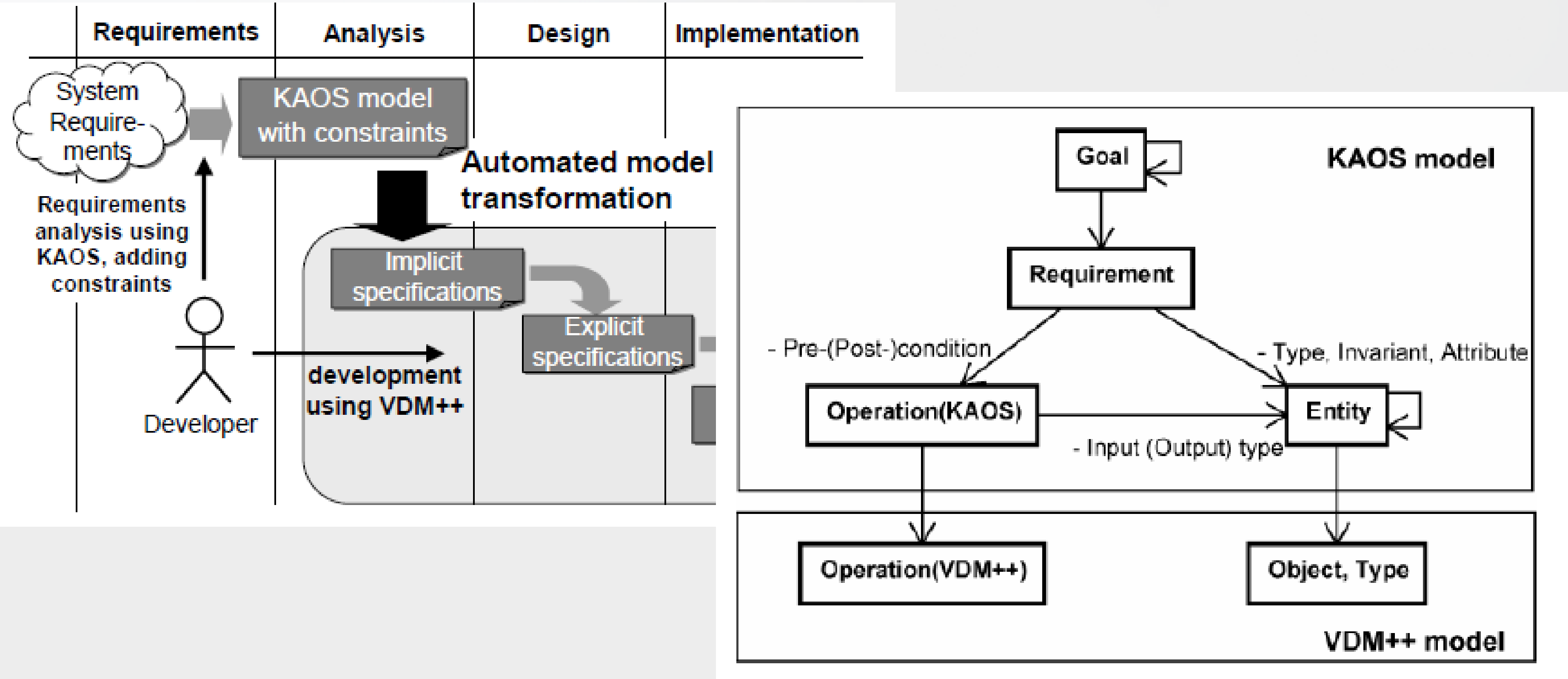
Check

UPPAAL Models

Annotation on time aspects and resource aspects (e.g., number of human workers, process instances)

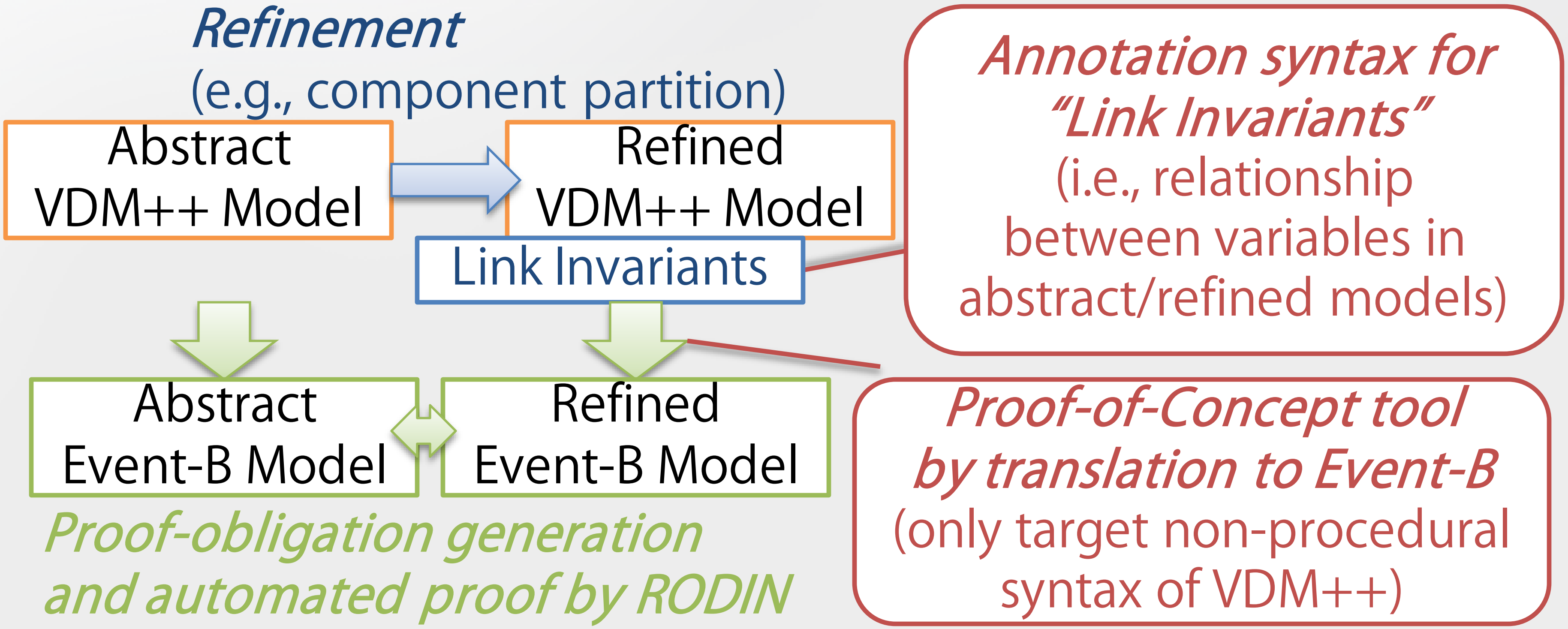
Examples of Graduation Studies (3)

- *Bridging-gaps type*: Develop a method and tool to derive VDM++ skeleton from requirements obtained by KAOS [Nakagawa, ASE07]



Examples of Graduation Studies (4)

- *Extension type*: Define a VDM++ extension to specify Event-B-type refinement relationships as well as a translator from the extended VDM++ to Event-B [Kawamata, SEFM09]



Refinement
(e.g., component partition)

Abstract
VDM++ Model

Refined
VDM++ Model

Link Invariants

Annotation syntax for "Link Invariants"
(i.e., relationship between variables in abstract/refined models)

Abstract
Event-B Model

Refined
Event-B Model

Proof-of-Concept tool by translation to Event-B
(only target non-procedural syntax of VDM++)

Proof-obligation generation and automated proof by RODIN

TOC

- Report educational activities for engineers from the industry
  - Overview of the Top SE Project
  - Lecture Courses
  - Graduation Studies
  - **Statistics and Discussion**

Statistics on Lecture Courses

For the 3rd students (30)

Series	Course	Students completed (attended)
Model Checking	Foundations (SPIN)	17 (21)
	Apps. (SPIN, SMV, LTSA)	12 (15)
	Performance (UPPAAL)	5 (10)
	Concurrency (CSP)	8 (10)
Formal Specs.	Foundations (VDM, B)	20 (27)
	Applications (VDM, B)	14 (20)
	Security (Event-B, Z, SPIN)	4 (5)
Impl. Techniques	Analysis (JML)	6(14)
	Verification (JPF)	5 (6)

Statistics on Graduation Studies: Methods

28 in total on FM, among the 1st-3rd students (61)

Series	Method/Tool	Num. of Studies
Model Checking	SPIN	8
	UPPAAL	2
	CSP (FDR/JCSP)	3
	Tool-independent	1
Formal Specs.	VDM	5
	Event-B	3
Impl. Techniques	JML (ESC/Java2)	1
	Java PathFinder	1
Combination	SPIN + SMV, SPIN + JPF, VDM + SPIN, VDM + Event-B	4

Statistics on Graduation Studies: Types

28 in total on FM, among the 1st-3rd students (61)

Classification	Num. of Studies
Case Study	6
Domain-Specific, Finer-Grained Support	11
Bridging Gaps between Different Methods/Tools	7
Extension of Methods/Tols	4

Some Note

- VDM is so popular (next to SPIN), which could be surprising?
  - Because of the Japanese companies: CSK (VDM Toolbox) and Sony/Felica (application to chips on so large number of mobile phones)
- Many students chose Domain-Specific support
  - Their comments were like "I like the method/tool and found it useful, but cannot make our all colleagues learn, think over and use the general one directly"
- While innovation in methods/tools is too difficult for them
  - As non-experts in semantics and formalisms

Summary

- Reported educational activities in the Top SE project
  - Target engineers from Japanese industry
  - Teach different methods/tools to recognize common principles and different approaches
    - Involve group exercises to work on real examples, which make students consider and discuss application strategies
  - Involve graduation studies, where students tackle problems they identify by themselves

⇨ Should be a good source of useful suggestions

Thank you!