Critical Systems Development with UML and Model-based Testing (Full-day tutorial)

Jan Jürjens
Software and Systems Engineering
Munich University of Technology*

1 Motivation

The high quality development of critical systems (be it dependable, security-critical, real-time, performance-critical, or hybrid systems) is difficult. Many critical systems are developed, fielded, and used that do not satisfy their criticality requirements, sometimes with spectacular failures.

Part of the difficulty of critical systems development is that correctness is often in conflict with cost. Where thorough methods of system design pose high cost through personnel training and use, they are all too often avoided.

UML offers an unprecedented opportunity for high-quality critical systems development that is feasible in an industrial context.

- As the de-facto standard in industrial modeling, a large number of developers is trained in UML.
- Compared to previous notations with a user community of comparable size, UML is relatively precisely defined.
- A number of analysis, testing, simulation, transformation and other tools are developed to assist the every-day work using UML.

However, there are some challenges one has to overcome to exploit this opportunity, which include the following:

*juerjens@in.tum.de – http://www4.in.tum.de/~juerjens

- Adaptation of UML to critical system application domains.
- Correct use of UML in the application domains.
- Conflict between flexibility and unambiguity in the meaning of a notation.
- Improving tool-support for critical systems development with UML.

The tutorial aims to give background knowledge on using UML for critical systems development and to contribute to overcoming these challenges.

In particular, we consider model-based testing, where test sequences are generated from an abstract system specification to provide confidence in the correctness of an implementation. For critical systems, finding tests likely to detect possible failures or vulnerabilities is particularly difficult, as they usually involve subtle and complex execution scenarios (and sometimes the consideration of domain-specific concepts such as cryptography and random numbers).

2 Outline

The tutorial presents the current academic research and industrial best practice by addressing the following main subtopics:

- UML basics, including extension mechanisms
- Applications of UML to
  - dependable systems
  - security-critical systems
  - embedded systems (development using IEC 61508)
  - real-time systems
- Extensions of UML (UML-RT, UMLsec, ...)
- Using UML as a formal design technique for the development of critical systems.
- Critical systems development methods.
- Case studies.

- Modeling, synthesis, code generation, testing, validation, and verification of critical systems using UML, in particular: Using the standard model interchange formats (XMI) for tool integration and to connect to validation engines. Existing tools.

- Model-based testing

As an example application domain, we focus on security-critical systems. We also show how to generalize the approach to the other application domains mentioned above.

The tutorial includes a demo of a prototypical tool for the formal analysis of UML models for critical requirements, which is based on XMI.

2.1 Secure systems development with UML

We present an extension of the Unified Modeling Language (UML) for secure systems development, called UMLsec, using UML's standard extension mechanisms.

We start by giving an overview of UML (the UML diagrams) and model management (packages, subsystems). We explain the UML extension mechanisms (stereotypes, tags, constraints, profiles).

We proceed to outline UMLsec, after discussing the requirements on an UML extension for secure systems development. We give the UMLsec profile.

We show how to formulate security requirements on a system and security assumptions on underlying layer in UMLsec. We explain how to use this information for risk analysis and how to evaluate the system specification against the security requirements, by making use of a formal behavioral model for a core of UML. Being able to formulate security concepts in the context of a general-purpose modeling language allows encapsulation of established principles of security engineering to avoid common vulnerabilities introduced by developers without in-depth training in security issues. The formal foundation of the approach allows the discovery of even non-obvious weaknesses that security experts may not detect without use of formal tools.

We sketch a design process to be used with the UML extension and discuss applicability of the approach with examples from various application domains (such as Java security and electronic payment schemes).

We discuss tool-support and present applications and examples (Java security, electronic purses).

3 Goals and Objectives

By the end of the tutorial, the participants will have knowledge on how to use the UML and model-based testing for a methodological approach to critical systems development. They will be able to use this approach when developing or analyzing critical systems, by making use of existing solutions and of sound methods of critical systems development.

4 Intended audience

The tutorial addresses practitioners and researchers in critical systems development interested in using UML and model-based testing (in particular for dependable, security-critical, or real-time systems).

5 Expected background

Basic knowledge of object-oriented software is assumed. No specific knowledge of UML or the various application domains is assumed.

6 Format

Lecture with examples of running programs. Generous time for question and answers will be provided.

Tutorial notes, printouts of slides will be distributed.

7 History

The proposed tutorial is a sequel to a highly successful series of tutorials presented at the conferences IFIP SEC 2002, DOCsec 2002, SAFECOMP 2002, AI 2003, ETAPS 2003, CSS 2003, and FME 2003 (see http://www4.in.tum.de/~juerjens/fdcsu History (user Participant, password Iwasthere)).

8 Equipment

Beamer (for laptop), overhead projector.

9 Biography

Jan Jürjens is a researcher at Munich University of Technology (Germany). He is the author of a book on Secure Systems Development with UML (Springer-Verlag, to be published in 2003) and over 20 papers in international refereed journals and conferences, mostly on computer security and software engineering, and has given invited talks at workshops and leading research institutes, including an invited talk to be presented at the UML Forum Tokyo 2003. He has created and lectured a course on secure systems development at the University of Oxford and tutorials at leading international conferences. He is the initiator and current chair of the working group on Formal Methods and Software Engineering for Safety and Security (FoMSESS) within the German Society for Informatics (GI). He is a member of the executive board of the Division of Safety and Security of the GI. He is on the advisory board for the Bavarian Competence Center for Safety and Security (KoSiB). He has been leading various security-related projects with industry. He is organizer or PC member of several international workshops.

Received awards include a scholarship from the German National Merit Foundation (Studienstiftung des deutschen Volkes) and a best student paper award from IFIP SEC. He has studied Mathematics and Computer Science at the Univ. of Bremen (Germany) and the Univ. of Cambridge (GB) and received a M.Sc. degree with best possible grade from the Univ. of Bremen. He has done research towards a PhD at the Univ. of Edinburgh (GB), Bell Laboraties (Palo Alto, USA), and the Univ. of Oxford (GB) and is currently being considered for a DPhil (Doctor of Philosophy) in Computing from the Univ. of Oxford.