Model-Driven Engineering of Dependable Critical Systems with UML (Full-day*tutorial)

Jan Jürjens
Software and Systems Engineering, TU Munich†

1 June 2004

1 Motivation

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

Systems whose correct functioning human life and substantial commercial assets depend on need to be developed very carefully. Systems that have to operate under the possibility of system failure or external attack need to be scrutinized to exclude possible weaknesses.

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.

*This tutorial could also be changed to half-day.

†80290 München, Germany. Tel. +49 (89) 289-28166, Fax +49 (89) 289-25310, juerjens@in.tum.de – http://www4.in.tum.de/~juerjens

- 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:

- 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. It includes an interactive tool demo with advanced tool support for UML.

2 Outline

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

- UML basics, including extension mechanisms
- Applications of UML to dependable systems, in particular
  - safety-critical systems
  - security-critical systems
  - real-time systems
  - performance-critical systems
- Extensions of UML (UML-RT, UMLsec, UMLsafe, ...)
- Using UML as a formal design technique for the development of critical systems.
- Critical systems development methods.

- 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.
- Case studies.
- Interactive tool demo.

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

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).

2.2 Other Critical System Domains

We demonstrate how to generalize the approach to other critical systems domains, with a focus on dependable systems.

3 Goals and Objectives

By the end of the tutorial, the participants will have knowledge on how to use the UML 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 (i.e. system and software developers, architects, and technical managers) and researchers interested in critical systems development using UML (in particular for dependable, security-critical, or real-time systems).

5 Expected background

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

6 Format

Lecture with examples of running programs. Generous time for question and answers will be provided. CDs of code are made available to attendees in advance, so they can run the samples on their own systems and ask questions as time permits.

7 History

The proposed tutorial is an adaption and extension of a series of about 30 tutorials presented at international conferences (see http://www4.in.tum.de/~juerjens/csdumltut then click on History (for slides and audio, need: user Participant, password Iwasthere)).

Feedback from these tutorials was gained through questionnaires distributed and collected at the tutorials and was used to improved the tutorial.

8 Biography

Jan Jürjens leads the Competence Center for IT-security at the Software & Systems Engineering chair at TU Munich (Germany). Holding a Doctor of Philosophy in Computing from the University of Oxford, he is the author of "Secure Systems Development with UML" (Springer-Verlag, 2004) and numerous publications on computer security and safety and software engineering. 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 within the GI, the executive boad of the committee on Modeling of the GI, the advisory board of the Bavarian Competence Center for Safety and Security, the working group on e-Security of the Bavarian regional government, and the IFIP Working Group 1.7 "Theoretical Foundations of Security Analysis and Design". He has been leading various security-related projects with industry.

See http://www4.in.tum.de/ ~juerjens for more information.

9 Equipment

Beamer (for laptop), overhead projector.

Tutorial notes, printouts of slides, and CDs with program samples will be distributed.