Integrating Security in the Development Process with UML

Folker den Braber
SINTEF ICT, Norway

Mass Soldal Lund
SINTEF ICT, Norway

Ketil Stølen
SINTEF ICT, Norway

Fredrik Vraalsen
SINTEF ICT, Norway

INTRODUCTION

Today, most business processes and communications as well as a lot of everyday life situations involve IT technology. Apart from requirements on functionality, this development of IT systems has increased the need for security. Security issues are reaching the main headlines in the media on a regular basis. Virus, worms, *mis*configuration and program bugs are common problems in a world where new releases and updates are almost as frequently announced as spam e-mail pop-ups in our inboxes.

Having a closer look at the world of IT technology through security-colored glasses, we observe that security is often one step behind functionality. Security issues are mainly addressed as a reaction to existing problems. Like firemen and emergency units, security measures do not come into action before it is too late.

The concept of IT security contains a lot of different aspects. One of the IT security aspects is security risk analysis, in the sequel referred to as security analysis. Security analysis is an inevitable and crucial activity for every system developer, system user or system owner in order to get control over and knowledge about the security level of the actual system.

Security analyses are costly and time consuming and cannot be carried out from scratch every time a system is updated or modified. This motivates the need for specific methodology addressing the integration of security analysis and system development, providing access to, storage of, and maintenance of analysis results.

CORAS (2004) provides such a methodology in the form of so-called “model based security analysis”. The CORAS methodology combines traditional risk analysis techniques like HazOp (Redmill, Chudleigh, & Catmur, 1999), FTA (IEC 1025, 1990) and FMEA (Bouti & Ait Kadi, 1994) with system development techniques like UML (OMG, 2003b) and UP (Jacobson, Rumbaugh, & Booch, 1999). It builds on international standards for risk management: the Australian/New Zealand AS/NZS 4360 (1999), “Risk Management”; the ISO/IEC 17799 (2000), “Code of Practice for Information Security Management”; the ISO/IEC 13335 (2001), “Guidelines for the management of IT Security”; and system documentation in the form of the Reference Model for Open Distributed Processing (RM-ODP) (ISO/IEC 10746, 1995).

BACKGROUND

The CORAS methodology incorporates a documentation framework, a number of closely integrated security analysis techniques, and a risk management process based upon widely accepted standards. It gives detailed recommendations for the use of modeling with UML and similar languages in conjunction with security analysis in the form of guidelines and specified diagrams. Security analysis requires a firm, but nevertheless easily understandable, basis for communication between different groups of stakeholders. Graphical, object-oriented modeling techniques have proven well suited in this respect for requirements capture and analysis. We claim they are as equally suited as part of a language for communication in the case of security analysis. Class diagrams, use case diagrams, sequence diagrams, activity diagrams, dataflow diagrams, and state diagrams represent mature paradigms used daily in the IT industry throughout the world. They are supported by a wide set of sophisticated case tools, are to a large extent complementary, and together support all stages of system development.

The CORAS methodology may be separated into three different components: tools, processes and languages. This is shown in Figure 1. The language part defines common languages to support the methodology. The process part includes instructions for how to "execute" the methodology, that is, descriptions of what should be done, and how and when it should be done. The CORAS methodology for model based security analysis (MBSA) integrates aspects from partly complementary risk analysis methods and state-of-the-art modeling methodology. During execution of the methodology, one may require the use of different tools. The CORAS methodology includes a computerized platform that can be integrated with third party modeling tools and risk analysis tools. It also includes two languages: a UML-based specification language, the UML profile for security analysis (OMG, 2003c), targeting security risk analysis, and an XML markup for exchange of risk analysis data (World Wide Web Consortium, 2000).

In the section called *Model-Based Security Analysis*, the core of model-based security analysis, based on integration of risk management and system development, is explained.

Related Approaches

The CORAS methodology, with its unique approach to security analysis from a modeling point of view, addresses a problem area in which also other methods and technologies exist. Some important ones are mentioned here.

The Common Criteria (CC)

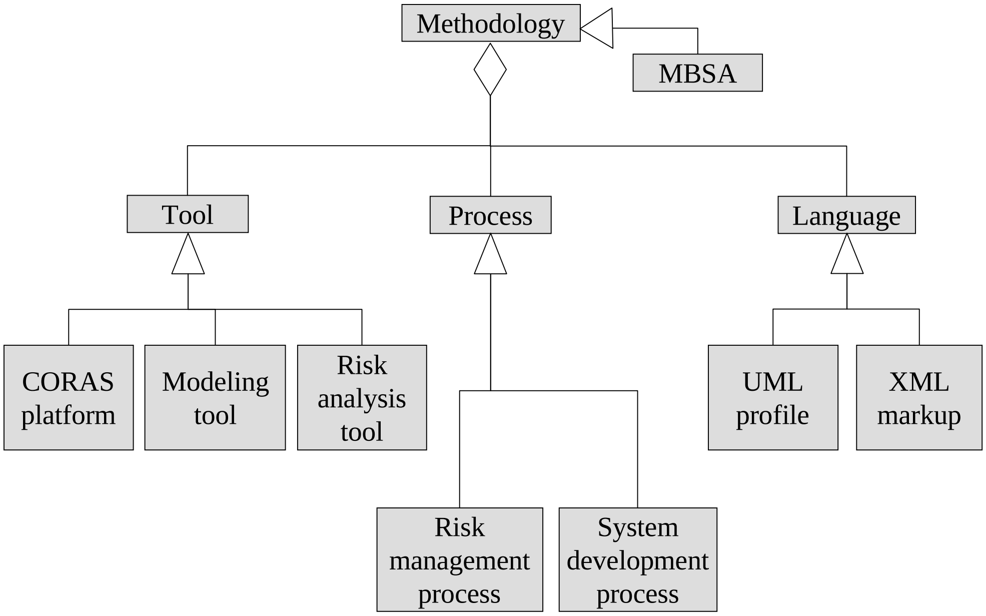
Since 1990, work has been going on to align and develop existing national and international schemes into one mutually accepted framework for testing IT security functionality. The Common Criteria (CC) (CCO, 2002) represents the outcome of this work. The Common Criteria project harmonizes the European "Information Technology Security Evaluation Criteria (ITSEC)" (Communications-Electronics Security Group, 2002), the "Canadian Trusted Computer Product Evaluation Criteria (CTCPEC)", and the American "Trusted Computer System Evaluation Criteria (TCSEC)" and "Federal Criteria (FC)". Increasingly, it is replacing national and regional criteria with a worldwide set accepted by the International Standards Organization (ISO15408) (ISO/IEC, 1999).

The CC and CORAS are orthogonal approaches. The CC provides a common set of requirements for the security functions of IT products and systems as well as a common set of requirements for assurance measures that are applied to the functions of IT products and systems during a security evaluation. CORAS provides a concrete methodology for model based security analysis.

Surety Analysis (SA)

Surety Analysis (SA), developed in Sandia National Laboratories (2003), is a methodology based on the creation of

Figure 1. Structure of the CORAS methodology


Methodology
MBSA
Tool
Process
Language
CORAS platform
Modeling tool
Risk analysis tool
UML profile
XML markup
Risk management process
System development process