Specifying model changes with UMLchange to support security verification of potential evolution

S. Wenzel b, D. Poggenpohl b, J. Jürjens a,b,⁎, M. Ochoa c,1

* Research problem

systems are neither developed in one step nor are they carved in stone. Hence, the task of evolving software systems such that the desired security requirements are preserved through a system's lifetime is of great importance in practice. We propose a model-based approach to support the evolution of software systems and preserving consistency of security requirements.

* Research objective(s)

allows the verification of potential future evolutions using an automatic analysis tool. An explicit model evolution implies the transformation of the model and defines a difference Δ between the original model and the transformed one. The proposed approach supports the definition of multiple evolution paths, and provides tool support to verify evolved models based on the delta of changes. This idea is visualized in The starting point of our approach is a software system model M which was already verified against certain security properties. Then, this model can evolve within a range of possible evolutions (the evolution space).

Study Context & Methodology/Experimental Design

use models specified using the Unified Modeling Language (UML) and the security extension UMLsec. UMLsec is given in form of a UML profile. Stereotypes are used together with tags to formulate the security requirements such as secrecy, integrity, and authenticity, and other security-relevant information. Constraints give criteria to determine if the requirements are met by the system design, by referring to a precise semantics of the used fragment of UML.

* Results

The industrial validation has shown that the approach enables secure model evolution. Evolving systems can be developed by pointing out possible security-violating modifications of previously secure models. We also showed that the implementation of the techniques described in this paper leads to a significant efficiency gain compared to the simple re-verification of the entire model.

* Main points/arguments

The approach of evolution-based security analysis with the UMLchange profile has been implemented in the model analysis tool CARiSMA.3 The tool is built on the basis of Eclipse and fully integrates into the Eclipse GUI by providing different views for defining analyses and presenting results to the user.

* Quality of related work

There are different approaches to deal with evolution that are relat- ed to our work. Within Software Evolution Approaches, [27] derives several laws of software evolution such as “Continuing Change” and “Declining Quality”. [28] argue that it is necessary to treat and support evolution throughout all development phases. They extend the UML metamodel by evolution contracts to automatically detect conflicts that may arise when evolving the same UML model in parallel. Similarly [29] discusses the verification of consistency through refinement, and [30] discusses consistency of models for incremental changes of models. These works can be integrated with the approach presented in this paper to enhance the detection of inconsistencies and conflicts.

* Strengths/ Weaknesses

CARiSMA has been implemented as a plugin based ar- chitecture. Using the modularity provided by this method, CARiSMA is distributed as plugins, of which the core plugin includes the main func- tionality. Furthermore, CARiSMA offers extension points facilitating the contribution of functionality of other plugins to provide different checks. CARiSMA can be started standalone as an RCP application or within an existing Eclipse. It is also possible to smoothly integrate it into existing modeling tools such as TOPCASED,4 IBM Rational Software Architect5 or other tools based on Eclipse.

* Significance & Contributions

providing a general approach for the specification and analysis of a number of sufficient conditions for the preservation of different security properties of the starting models in the evolved models. The approach contains a novel UML profile, called UMLchange, which allows modelers to specify possible evolution paths of a model and to verify them with respect to quality properties such as consistency of security requirements.