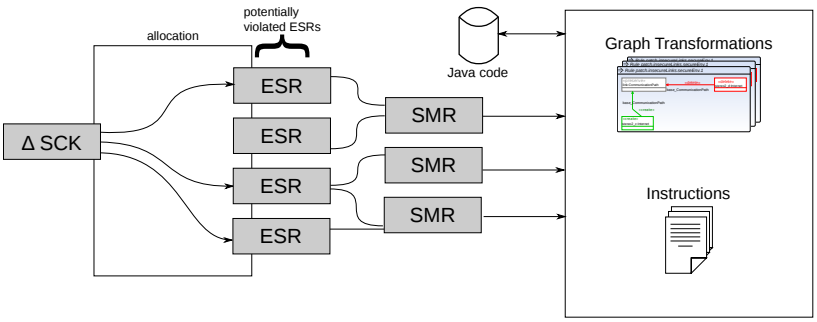
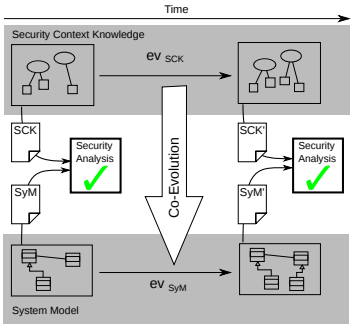
Master / Bachelor

Security-preserving co-evolutions / Sicherheitserhaltende Co-Evolutionen

This thesis can also be supervised and written in English. For the English version of the announcement see below.



(a) Komponenten für Co-Evolution



(b) Verhältnis von Evolution zu Co-Evolution

Motivation

Model-driven software engineering especially is used for compliance to requirements of software systems. Requirements are mostly rather abstract and can come from various sources (e.g. laws, domain-specific regulations, company policies).

Nowadays, many systems are used over a long timespan (i.e. long-living systems). Many of these are used to provide critical infrastructure, so that livelong compliance to security requirements is of vital significance.

The project *SecVolution* captures and manages knowledge of a specific project and general security knowledge (Security Context Knowledge (SCK)) as well as essential security requirements (ESR) and to make them part of a software model, thus supporting compliance to them during development and even run-time.

If the knowledge evolves, it needs to be checked if the system model still is compliant to the evolved knowledge (s. Abb. 1b).

During run-time, it is necessary to ensure that security assumptions made during design phase hold during the system execution.

—

Deutsche Version:

—

Modellgetriebene Softwareentwicklung eignet sich unter anderem dazu, die Einhaltung von Anforderungen an das System sicher zu stellen. Anforderungen (*requirements*) werden auf abstrakter Ebene im Entwicklungsprozess definiert oder sind bereits in Form von externen Quellen vorgegeben.

Software, die in kritischen Bereichen eingesetzt wird, wird oftmals über eine lange Laufzeit betrieben (*langlebiges System*). Langlebige Systeme sind insbesondere mit dem sich verändernden Wissen ihrer Umgebung konfrontiert, wie beispielsweise der Tatsache, dass bestimmte Verschlüsselungsalorithmen, die als sicher gelten, wenn ein System initial entwickelt wird, später unsicher werden können.

Das Projekt *SecVolution* beschäftigt sich damit, Wissen über ein Projekt und allgemeines Sicherheitswissen (Security Context Knowledge (SCK)) sowie generische Sicherheitsanforderungen (Essential Security Requirements (ESR)) zu sammeln und zu verwalten und als Annotationen an das Modell abzubilden, um die Einhaltung dieser Anforderungen bei der Modellierung / Implementierung sicherzustellen.

Ändert sich das Wissen (Evolution), muss überprüft werden ob das System-Modell die Sicherheitsanforderungen gegenüber dem veränderten Wissen noch erfüllt (s. Abb. 1b).

Aufgabenstellung/Ziele

Im Rahmen dieser Arbeit sollen Aspekte der Abb. 1a theoretisch untersucht werden. Die praktische Umsetzung soll anhand von Fallbeispielen und der Implementierung / Erweiterung vorhandener Werkzeuge / Ansätze erfolgen:

- Ermittlung gefährdeter Sicherheitsanforderungen (*ESR*s) durch Analyse von Differenzen in der Wissensbasis (ΔSCK)
- Erarbeitung von Sicherheits-Bewahrungs-Regeln (*security maintenance rule*s (SMRs)), die für ein gegebenes Szenario folgendes leisten:
  - Definition möglicher Wissensänderungen der Wissensbasis (ΔSCK)
  - Analyse, ob Sicherheitsanforderungen gefährdet sind (ESRs)
  - Bestimmen von nötigen Korrekturoperationen (Co-Evolutionen; *Reactions*)

Die Arbeit soll auf die bestehenden Vorarbeiten im Rahmen des Projekts SecVolution aufbauen.

Motivation

Model-driven software engineering especially is used for compliance to requirements of software systems. Requirements are mostly rather abstract and can come from various sources (e.g. laws, domain-specific regulations, company policies).

Nowadays, many systems are used over a long timespan (i.e. long-living systems). Many of these are used to provide critical infrastructure, so that livelong compliance to security requirements is of vital significance.

The project *SecVolution* captures and manages knowledge of a specific project and general security knowledge (Security Context Knowledge (SCK)) as well as essential security requirements (ESR) and to make them part of a software model, thus supporting compliance to them during development and even run-time.

If the knowledge evolves, it needs to be checked if the system model still is compliant to the evolved knowledge (s. Abb. 1b).

During run-time, it is necessary to ensure that security assumptions made during design phase hold during the system execution.

Objectives

The goal of this thesis is to investigate several aspects as depicted in fig. 1a. The investigation should start on a theroretical base. The practical realization should be done using a case study and implementing a new or extending existing tools / approaches. Conceivable sub-goals are:

- Determination of endangered security requirements (*ESR*s) by querying the knowledge base(ΔSCK)
- Defining so-called security maintenance rules (SMRs), that combine the following:
  - Correspondence to a specific evolution of the knowledge base (ΔSCK)
  - Analysis of potentially endangered (ESRs), e.g. using model queries
  - Determination of possible corrective actions (*co-evolutions*)

The thesis is in context of the SecVolution project and should be build by integrating with existing approaches and tools.

Organisatorisches

Kontakt:
Dipl.-Inf. Jens Bürger (buerger@uni-koblenz.de)