Ausgewählte Themen des Modellbasierten Sicherheits-Engineerings

Sommersemester 2012

LS14
Arbeitsgruppe
Software Engineering for Critical Systems

5.4.2012

Outline

Vorstellung der AG

Das Seminar - Wichtige Meta-Fähigkeiten

Werbung

Abschlussarbeiten

- Themen siehe: `http://www-jj.cs.tu-dortmund.de/secse/pages/teaching/thesis/index_de.shtml`
- Seminarthemen bereiten Abschlussarbeitsthemen vor

Hilfskräfte

- Themen siehe: `http://www-jj.cs.tu-dortmund.de/secse/pages/home/jobs_de.shtml`
- Mitarbeit in verschiedenen Projekten

Ablauf

Leistungsbestandteile

- Kommentierte Gliederung
- Review-Fassung
- Reviews
- Abgabe Ausarbeitung
- Abgabe Folien
- Vortrag
- Diskussion

Betreuung

- Vorgespräch (Verständnisfragen)
- Besprechung der Gliederung
- Besprechung der Reviews/ der Reviewfassung
- Besprechung der Folien

Ausarbeitung

Umfang

- ca. 15 Seiten Hauptinhalt, nicht mit gerechnet:
  - Titelblatt
  - Inhalts- / Tabellen- / Abbildungsverzeichnis
  - Bibliographie
- min 10 Seiten Reintext
  - Ohne Abbildungen
  - Ohne Kapitelumbrüche

Vorlagen (Bitte Einhalten)

Liegen im Latex und Word Format vor

`http://www-jj.cs.tu-dortmund.de/secse/pages/teaching/allgemeineInfo/index_de.shtml`

Ausarbeitung II

Inhalt

- Verständliche Darstellung des Inhalt
  - Zielgruppe: Studenten mit abgeschlossen Bachelor
  - Selfcontainment: Erklären der benötigten Begriffe
- Fazit mit eigener kritischer Stellungnahme

Einstiegsliteratur

Wichtig: Nutzung weitergehender Literatur!

Gliederung

- Kapitelüberschriften
- Kurze Übersicht über die Kapitelinhalte (ca. 100 Worte pro Kapitel)
- Literaturübersicht

Review

Zwei Reviews

- Jeder bekommt zwei Reviews
- Jeder erstellt zwei Reviews

Inhalt und Form

- ca. 1 Seite
- Kurze Zusammenfassung
- Positive Punkte
- Problem Punkte
- Verbesserungsvorschläge

Vortrag

Umfang

- Vortragsdauer: 35 Min (30-40 Min ok)
- anschließend Diskussion

Beamer und Präsentationsrechner (PDF) stehen zu Verfügung.

Zum Inhalt

- Spannungsrahmen erzeugen
- Benötigte Grundlagen kurz aber ausreichend

Was selbstverständlich sein sollte....

Plagiat

Durchgefallen und Benachrichtigung des Prüfungsausschusses!

Verspätete Abgabe

- Ohne Absprache wird die Teilleistung mit 5 bewertet
- Absprache muss von Betreuer bestätigt werden

Anwesenheit

Bei allen Vorträgen ist die Anwesenheit Pflicht!

Abgabeformat

PDF

Zeitplan

5.4.12 (10:00)	Themenvorstellung
10.4.12 (12:00)	Rückmeldung
2.5.12 (24:00)	Abgabe Gliederung
21.5.12 (24:00)	Abgabe Vorversion Ausarbeitung
11.6.12 (24:00)	Abgabe Reviews
25.6.12 (24:00)	Abgabe Ausarbeitung
2.7.12 (24:00)	Abgabe Folien
4-6.7.12	Vorträge

Noten...

Ausarbeitung und Gliederung 40%

Struktur, Verständnis, Form, Inhalt, Quellen, ...

Review 10%

Struktur, “Hilfeleistung“, ...

Vortrag 40%

Verständlichkeit, Aufbau, ...

Teilnahme an der Diskussion 10%

Häufigkeit, Qualität, ...

Themen Rückmeldung

Mail mit 5 Themenwünschen (nach Priorität geordnet)

- ASAP
- vorname.nachname@cs.tu.dortmund.de (Thomas Ruhroth)
- Name, Matrikelnummer, Studiengang, Semester
- relevante Vorlesungen und Seminare
- weitere qualifizierende Vorkenntnisse

Deadline

Di 10.4.12 (12:00)

Grundlagen des expliziten Model Checkings

- Qualitätssicherung durch Tests und Simulationen kann keine vollständige Sicherheit über die Korrektheit eines Programms liefern
- $\rightarrow$ Model Checking betrachtet alle möglichen Verhalten eines Systems
- Geeignet formulierte Eigenschaften können überprüft werden

Vortrag

- Vorstellung der allgemeinen Grundlagen
- Modellierung der Systeme
- Logische Darstellung von Anforderungen
  - Insb. durch CTL
- Algorithmische Durchführung der Überprüfung

Model Checking in der Praxis

Model Checking

- Theoretisch fundiert
- Betrachtet "'geeignete"' Systembeschreibungen
  - Zustandsautomaten
  - PROMELA
- Hohe Komplexität

Praktische Softwareentwicklung

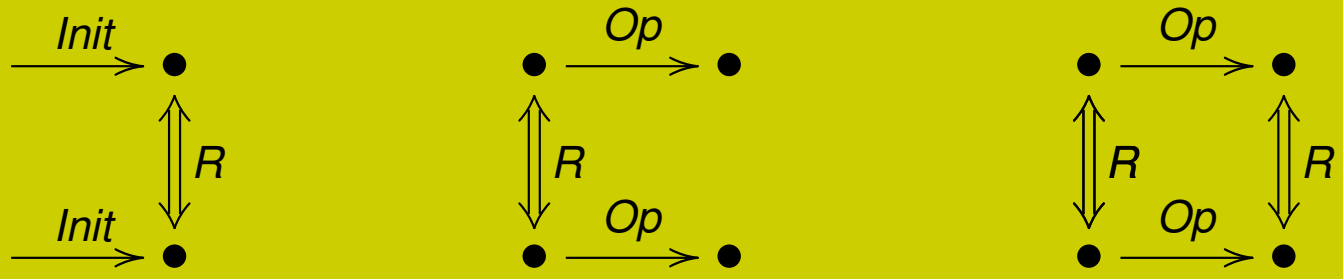
- C/C++, Java, ...
- Komplexe Projekte
- Praktischer Einsatz von Model Checking ist problematisch

Vortrag

- In welchen Bereichen kann Model Checking praktisch eingesetzt werden?
- Welche Werkzeuge existieren und wie arbeiten diese?
  - Java Pathfinder
  - Zing

Einführung ins Refinement Checking

- Systeme können auf verschiedenen Abstraktionsleveln beschrieben werden
- *Refinement* bezeichnet die Veränderung der Spezifikation eines Systems, ohne funktionale Änderungen
- *Refinement Checking* ermöglicht den Vergleich zweier Spezifikation auf (funktionale) Äquivalenz



Modell-getriebene Test-Generierung

Motivation

- Testen komplexer Systeme
- Testqualität steigern
- Automatisierung der Testerstellung

Inhalt

- Konzept MBT
- Ansätze zur Testfall-Generierung
  - Model Checking
  - Markov-Ketten
  - ...
- MODEST
  - Tool zur generativen Programmierung
- Weitere Ansätze, Tools, ...

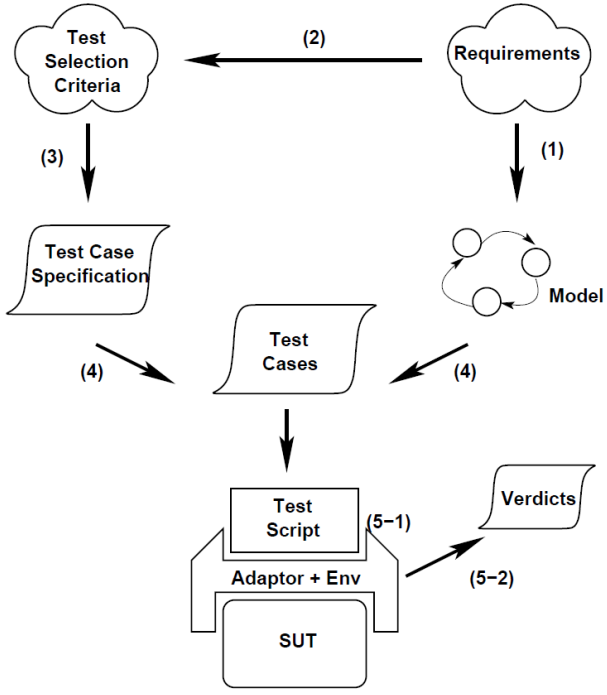


Figure: Quelle: A Taxonomy of Model-Based Testing, Utting et al., 2006

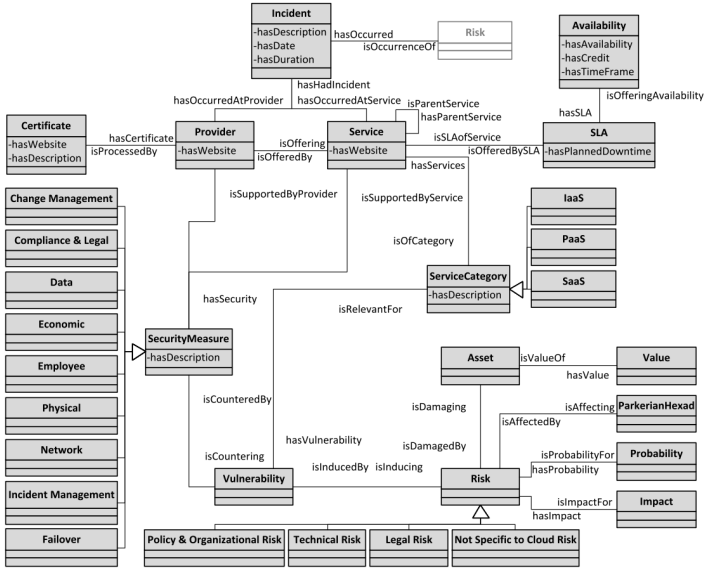
Ontologien für Sicherheit und Compliance in Clouds

Motivation

- Cloud Computing in aller Munde
- Software soll sicher sein
- Spezifikation und Prüfung von Sicherheitsanforderungen erfordert Wissen über die Cloud

Cloud-Ontologien

- Referenzmodelle f.d. Cloud
- Ontologien f. Sicherheitsbegriffe



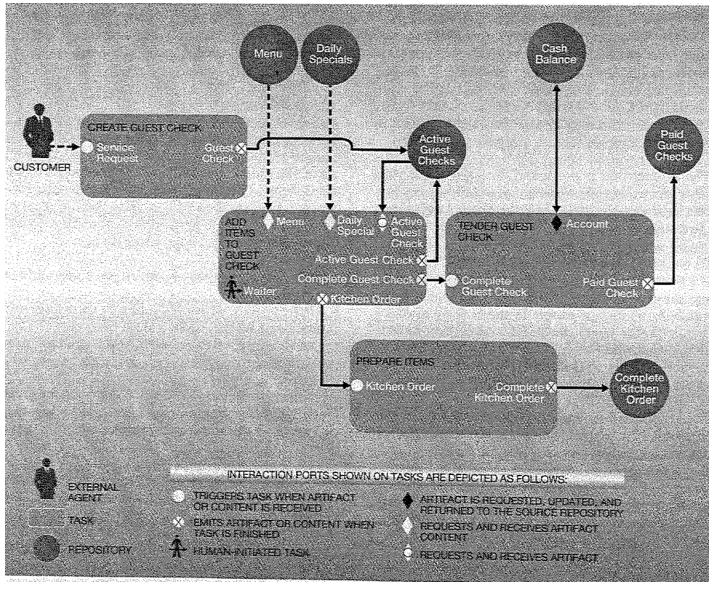
Ziel

- Ontologien recherchieren
- Alternativen vergleichen

Artefakt-basierte Modellierung von Geschäftsprozes

Motivation

- Geschäftsprozesse werden oft ereignisbasiert modelliert
- Prüfung von Prozessen möglich



Artefakt-basierte Ansätze

- Dokumente im Mittelpunkt

Ziel

- Vorstellung Artefakt-basierter Modellierung
- Abgrenzung von ereignisbasierten Ansätzen
- Recherche von Analyseverfahren

Compliance-Checking auf BPMN-Modellen

Motivation

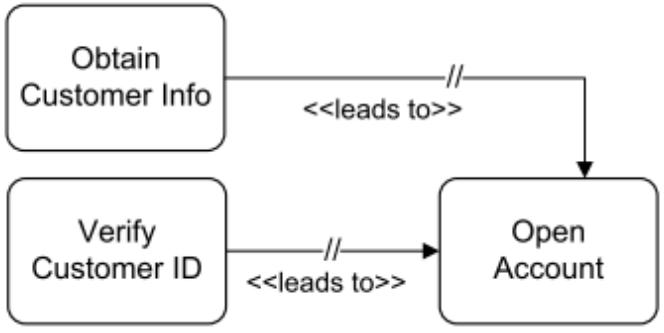
- BPMN zur Modellierung von Prozessen etabliert
- Korrektheit von Prozessen / Compliance zu Regularien ist wichtig

BMPN-Q

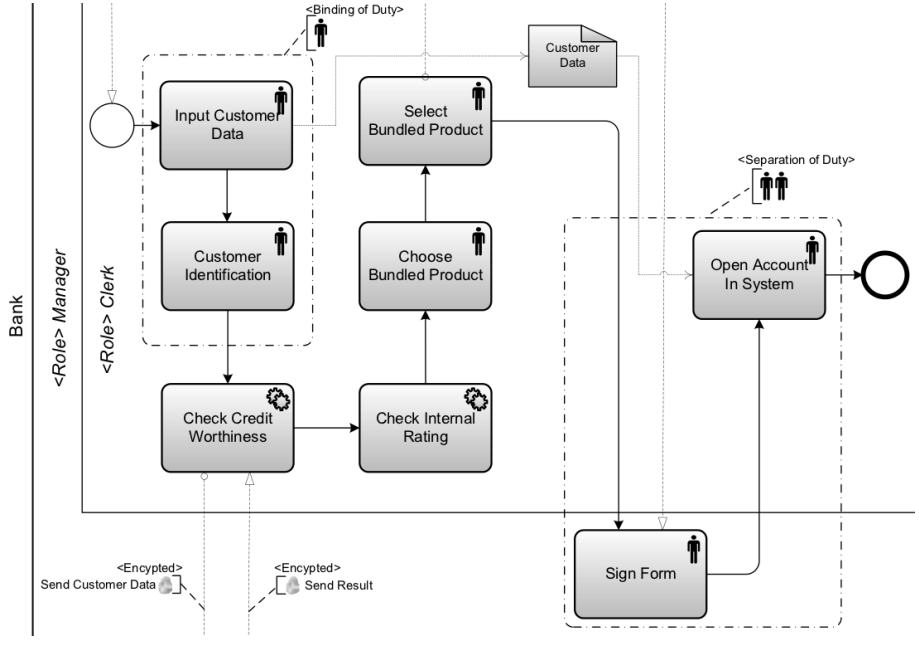
- Querysprache f. BPMN
- Compliance-Checks möglich

Ziel

- BPMN-Q vorstellen
- Sicherheitsanalysen diskutieren
- Alternativen recherchieren

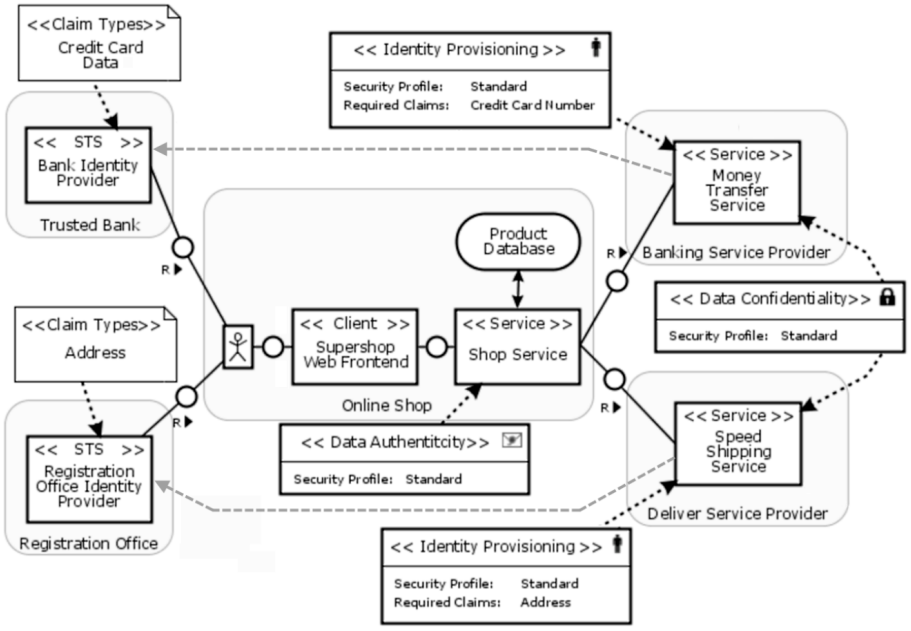


Modelling security goals in Business processes



- Business process models visualize tasks, information flow, involved people, etc.
- Optimized to be read by people not related to IT security
- Model also security goals helps within the software design process
- Examplary languages for business process models: EPK or BPMN

Modelling Security Requirements for Service-Oriente Architectures



- Service-Oriented Architectures (SOA) provide and orchestrate business services
- Web services are the technical foundation of SOA
- Security Requirements can be specified to secure communication etc.
- Simplify security by model-driven approach
- Example: SecureSOA

A Domain-Specific Language for Computing on Encrypted Data

- auf Haskell basierende DSL
- für Cloud Computing und Homorphic Encryption
- ermöglicht Beweise und Verifikation

Listing 2 Static semantics for expressions and values ("reference" semantics).

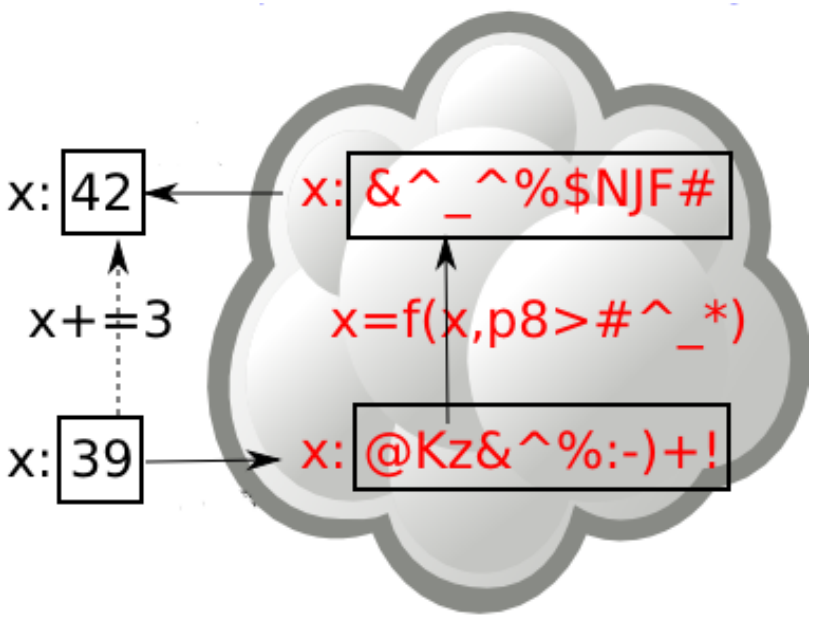
$$\frac{}{\Gamma \vdash y : (Y, \mathrm{P})} \qquad \frac{\Gamma \vdash e : (Y, \mathrm{S})}{\Gamma \vdash \texttt{reveal}\ e : (Y, \mathrm{P})} \qquad \frac{\Gamma[x \mapsto \tau_1] \vdash e : \tau_2}{\Gamma \vdash \lambda x.e : \tau_1 \to \tau_2}$$

$$\frac{}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{\Gamma \vdash e_1 : \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1\, e_2 : \tau_2} \qquad \frac{\Gamma \vdash e_1 : (Y, \alpha) \qquad \Gamma \vdash e_2 : (Y, \beta)}{\Gamma \vdash \mathrm{op}_i(e_1, e_2) : (Y, \alpha \sqcup \beta)}$$

$$\frac{y \in Y}{\Gamma \vdash_v (y, \alpha) : (Y, \alpha)} \qquad \frac{\Gamma \vdash \lambda x.e : \tau_1 \to \tau_2}{\Gamma \vdash_v \lambda x.e : \tau_1 \to \tau_2}$$

Homomorphic Encryption

- Arbeiten auf verschlüsselten Daten
- Generell: Homomorphe Verschlüsselung
- Speziell: Auf Basis von “Ring learning with errors”



JIF - Typed Java for Secure Information Flow

Ungewollter Informationsfluss

```
l = false;
if (h) {
    l = true;
}
```

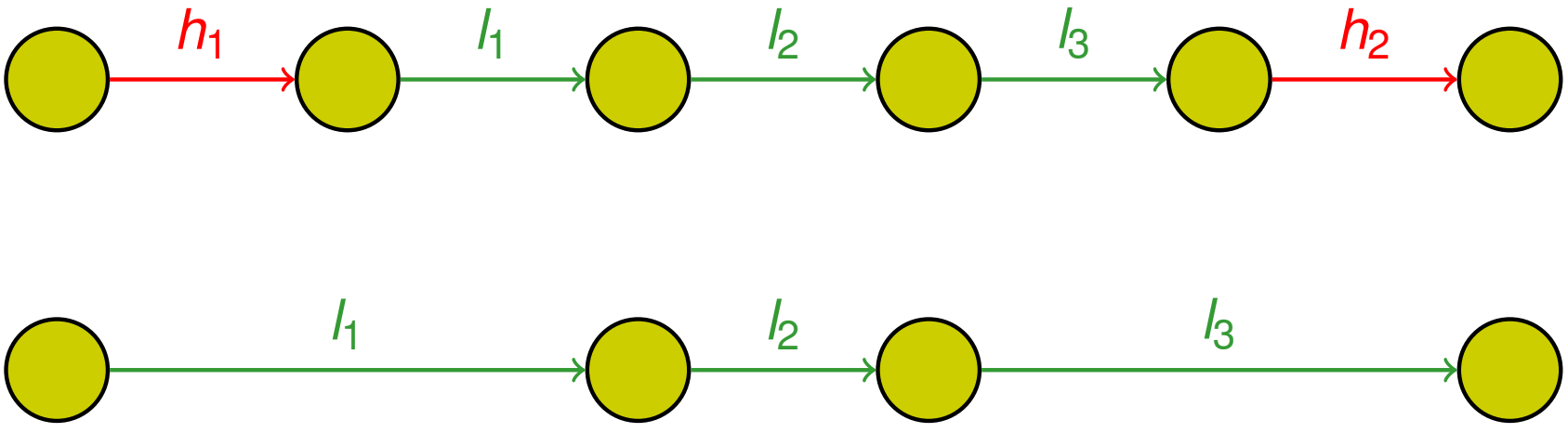
JIF

- Erweiterung für Java
- Statische Analyse
- Basiert auf einem Typsystem
- Toolsupport

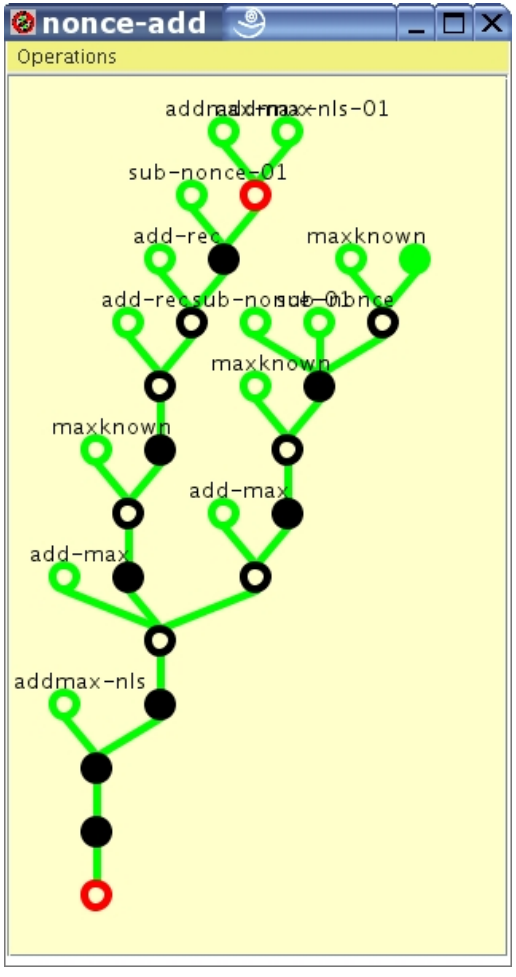
Modular Assembly Kit for Security (MAKS)

Example: $SR_{\mathrm{V,N,C}}(Tr)$ – Strict Removal

$SR_{\mathrm{V,N,C}}(Tr) := \forall\, \tau \in Tr : \tau\,|_{\mathrm{V \cup N}} \in Tr$



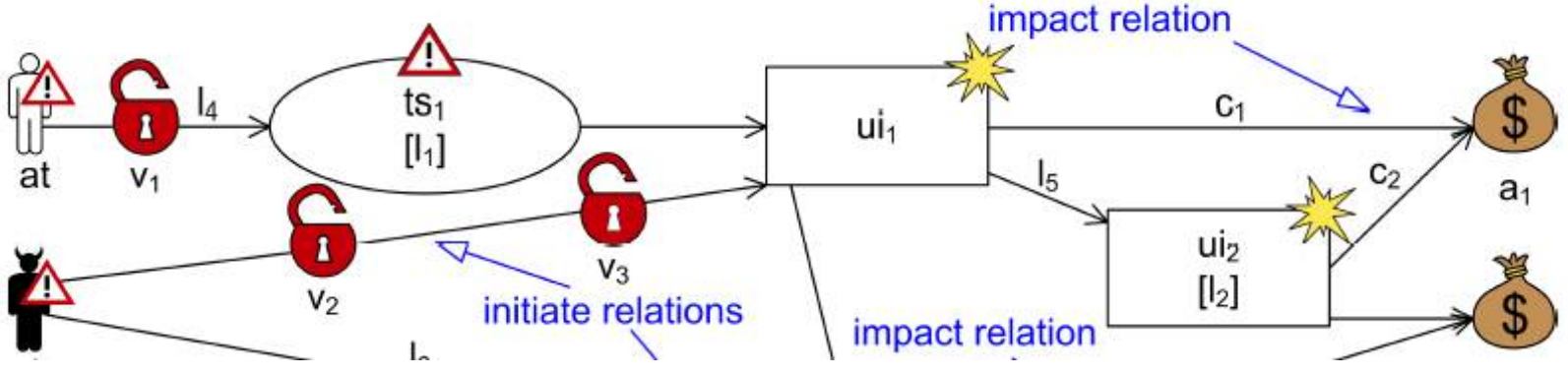
KIV - Ein (semi)-automatisches Beweissystem



- Algebraische Spezifikationen
- Graphische Beweisunterstützung
- Beweise sind “exportierbar”
- Halbautomatisch

Model-based risk assessment with CORAS

- Sicherheitsrisikoanalyse:
  - Wie sicher ist ein Internetbankkonto?
  - Wie gef"ahrlich kann ein Fehler einer einzelnen Person f"ur die Firma sein?
- CORAS, eine Methode f"ur Sicherheitsrisikoanalysen
  - verwendet UML Profile zur Analyse und Darstellung
  - Tool f"ur Dokumentation, Wartung und Analysereport
- Vortrag:
  - Vorstellung des Coras-Modells inkl. Beispiel
  - kurze Vorstellung der Tools



System Modelling and Simulation with Core Gnosis/D

- Core Gnosis
  - ausf"uhrbare Modellierungssprache
  - zum modellieren und simulieren komplexer Systeme mit
    - stochastischen Eing"angen
    - ortsbezogene Ressourcen
  - existiert ein sematischer Calculus zur mathematischen Darstellung
- Vortrag:
  - Vorstellung der Core Gnosis inkl. Beispiels
  - ggf. Vorstellung der mathematischen Hintergr"unde

Themen Rückmeldung

Mail mit 5 Themenwünschen (nach Priorität geordnet)

- ASAP
- vorname.nachname@cs.tu.dortmund.de (Thomas Ruhroth)
- Name, Matrikelnummer, Studiengang, Semester
- relevante Vorlesungen und Seminare
- weitere qualifizierende Vorkenntnisse

Deadline

Di 10.4.12 (12:00)

Thank you

Questions?