LEHRSTUHL 14
SOFTWARE ENGINEERING

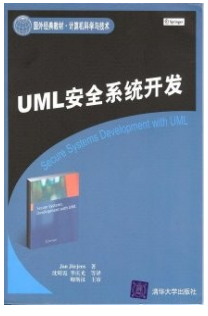
Willkommen zum Seminar-Kickoff Ausgewählte Themen des Modellbasierten Sicherheits-Engineerings

Wer bin ich?

- Professor für Software Engineering an der TU Dortmund
- Wissenschaftskoordinator „Enterprise Engineering“ am ISST
- Leiter der Fraunhofer-Attract-Projektgruppe „Architectures for Auditable Business Process Execution (Apex)“
- Wissenschaftlicher Direktor des EU-Projektes „Security Engineering for Lifelong Evolvable Systems (Secure Change)“
- Senior Member am Robinson College (Univ. Cambridge)

Vorher u.a.:

- Royal Society Industrial Fellow bei Microsoft Research Cambridge
- Research Fellow am Robinson College (Univ. Cambridge)
- Postdoc an der TU München
- Promotion zu „Principles for Secure Systems Design“ (Univ. Oxford)
- Forschungsaufenthalte am LFCS (Univ. Edinburgh) und Bell Labs (Palo Alto)
- Studium an Univ. Bremen und Univ. Cambridge



Wer ist meine Forschungsgruppe?

- Misha Aizatulin (Microsoft Research Cambridge)
- H. Selcuk Beyhan (Logica (Germany))
- Stephan Braun (TUD)
- Francois Dupressoir (Microsoft Research Cambridge)
- Michael Giddings (Open University)
- Thorsten Humberg (Fraunhofer ISST)
- Christopher McLaughlin (Gartner)
- Martin Ochoa (TUD / Siemens)
- Sebastian Pape (TUD)
- Dr. Thomas Ruhroth (TUD)
- Stefan Taubenberger (Münchener Rückversicherung)
- Daniel Warzecha (Fraunhofer ISST)
- Dr. Sven Wenzel (TUD)
- Christian Wessel (TUD)

Vertrauenswürdige IT-Systeme

IT Systeme durchziehen heute fast alle Funktionen in Wirtschaft und Gesellschaft. IT hat direkten (oft invasiven) Einfluss auf fast alle Aspekte menschlichen Lebens.

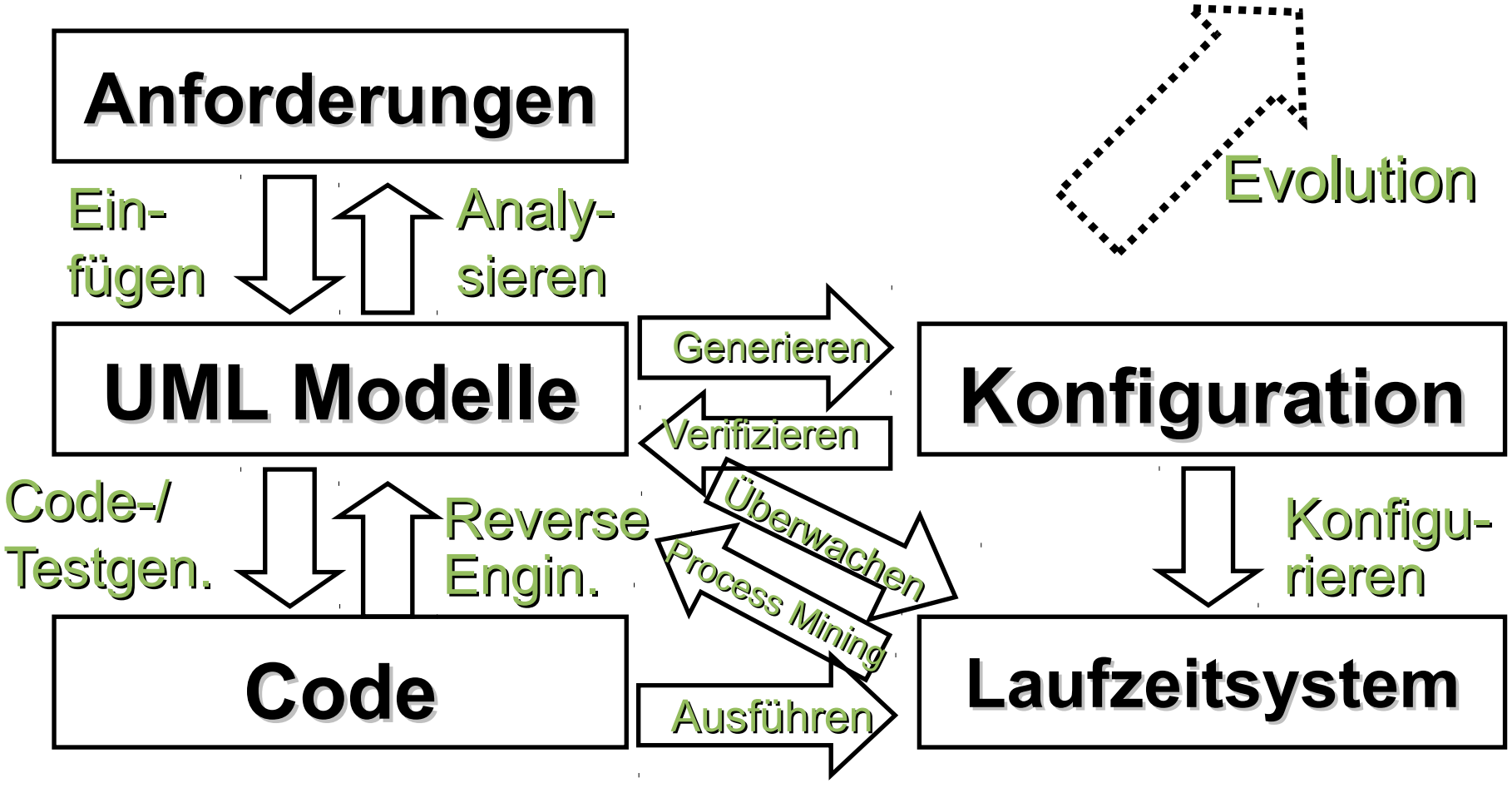
Die Erwartungen an die Vertrauenswürdigkeit dieser Systeme sind daher in den letzten 10 Jahren stark gestiegen. Diese Erwartungen werden oft nicht erfüllt. Teil des Problems ist, dass die bislang verwendeten System- und Software-Entwicklungsmethoden mit den gestiegenen Erwartungen bei gleichzeitig steigender Systemkomplexität nicht mithalten konnten.

Offene Systeme

Aus Flexibiliäts- und Kostengründen sind moderne IT Systeme meist über offene Infrastrukturen realisiert, zum Beispiel:

- Internet
- Mobile Netze

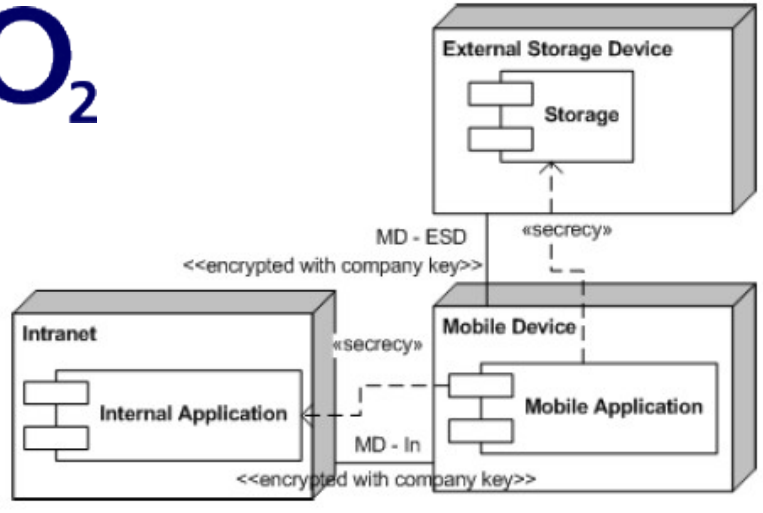
Aufgrund ihrer Offenheit sind sie dem Zugriff von Personen ausgesetzt, die in verschiedenem Maße vertrauenswürdig sind. Dieser Zugriff muss daher systemseitig reguliert werden. Aus Flexibiliäts- und Kostengründen wird dies oft auf der Softwareebene gelöst. Eine vertrauenswürdige IT braucht also sichere Software.


Anforderungen
Ein-fügen
Analy-sieren
Evolution
UML Modelle
Generieren
Verifizieren
Konfiguration
Code-/ Testgen.
Reverse Engin.
Überwachen
Process Mining
Konfigu-rieren
Code
Ausführen
Laufzeitsystem

Beispielanwendung: Mobile Informationssysteme bei O_2

- Anwendung von UMLsec auf mobile Kommunikations-Architekturen bei O_2 (Germany). [ICSE 08]
- Alle 62 Sicherheitsanforderungen aus der Security Policy erfolgreich verifiziert.
- Modellbasierte Techniken bringen Zusatzaufwand.
- Macht sich bezahlt bei wichtigen Sicherheitsanforderungen und Konzentration auf kritische Architekturanteile (auch im Vergleich mit anderen Qualitätssicherungs-Ansätzen mit vergleichbarer Verlässlichkeit)
- UMLsec adäquat für mobile Architekturen.

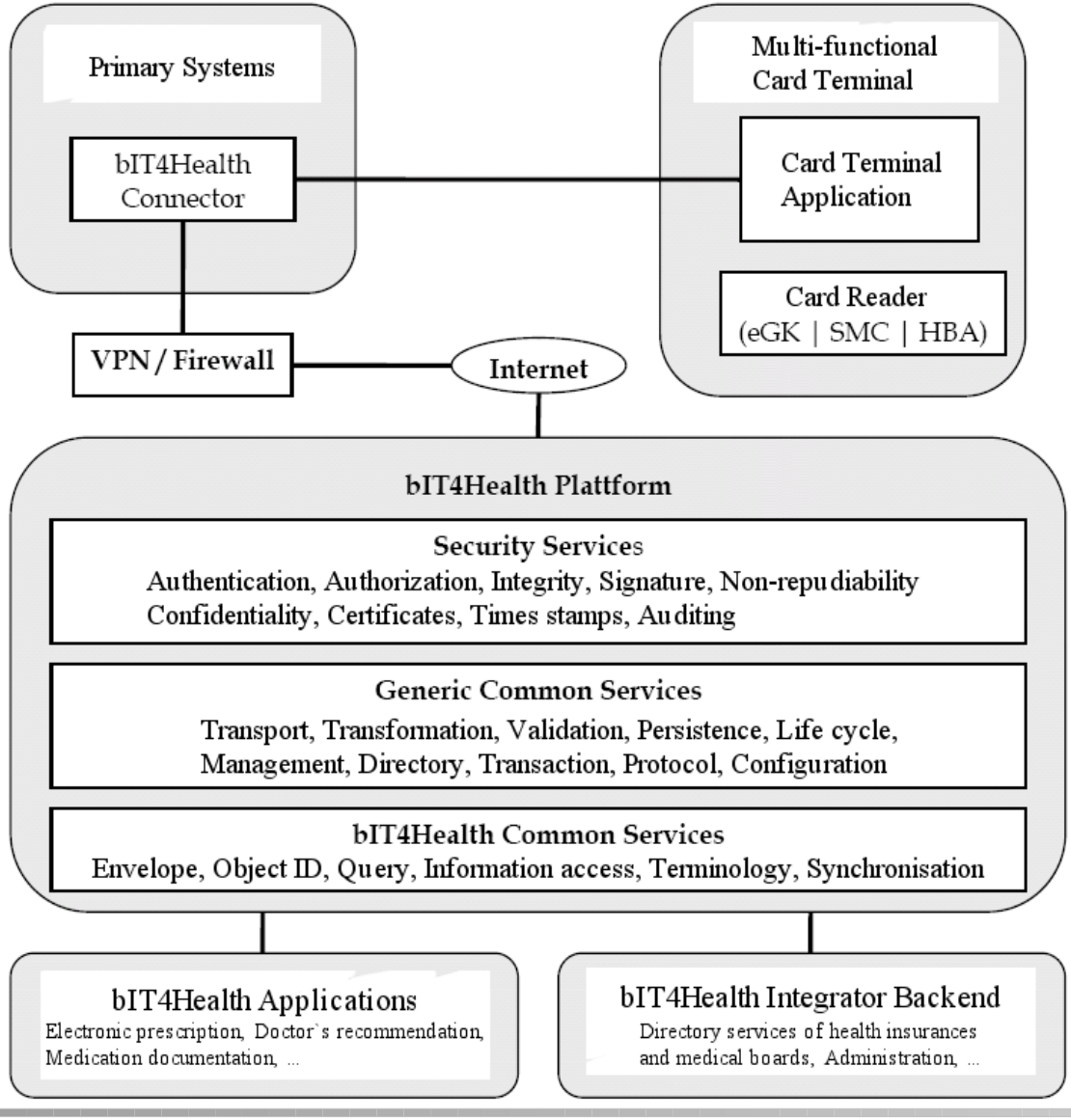
O_2



Beispielanwendung: Informationssysteme im Gesundsheitsbereich: Die Gesundheitskarte

- Architektur mit UMLsec analysiert.
- Einige Schwachstellen aufgedeckt (fehlender Vertraulichkeitsschutz für digitale Rezepte).

[Meth. Inform. Medicine 08]



Beispielanwendung: Bank-Informationssystem bei der HypoVereinsbank

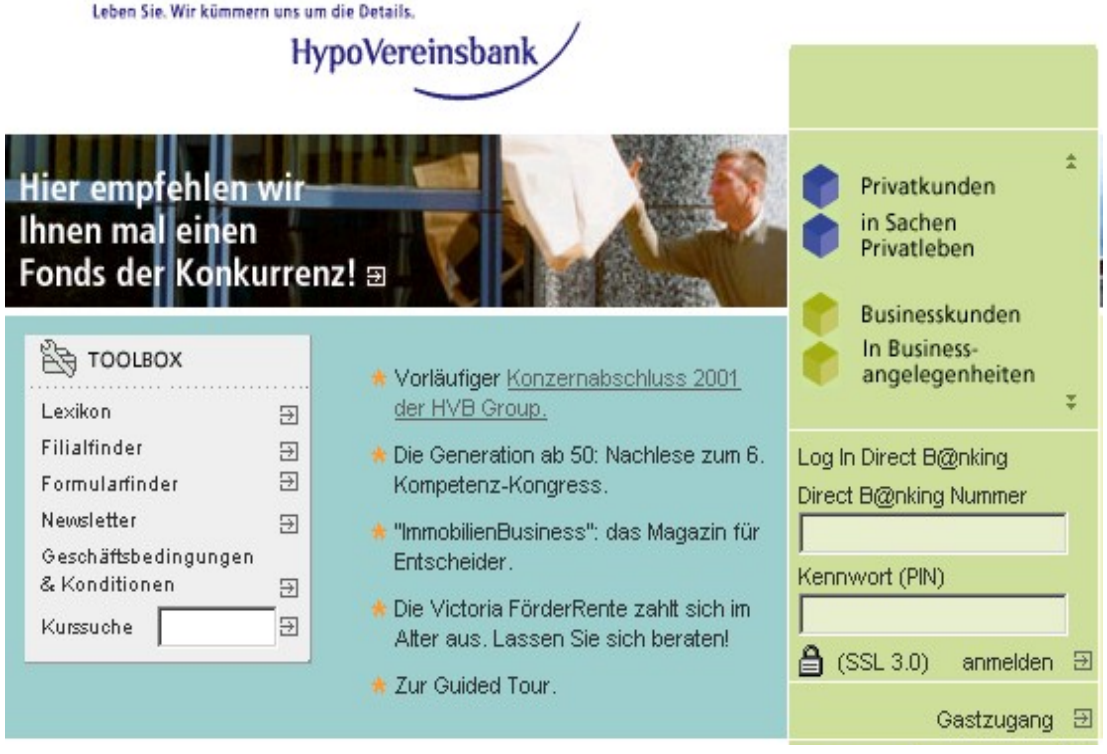
Modellbasierte Sicherheitsanalyse von webbasierter Bankanwendung ("digitaler Formularschrank").

Geschichtete Architektur (SSL Protokoll, darauf Client Authentisierungs-Protokoll)

Anforderungen:

- Vertraulichkeit
- Authentisierung

[SAFECOMP 03]



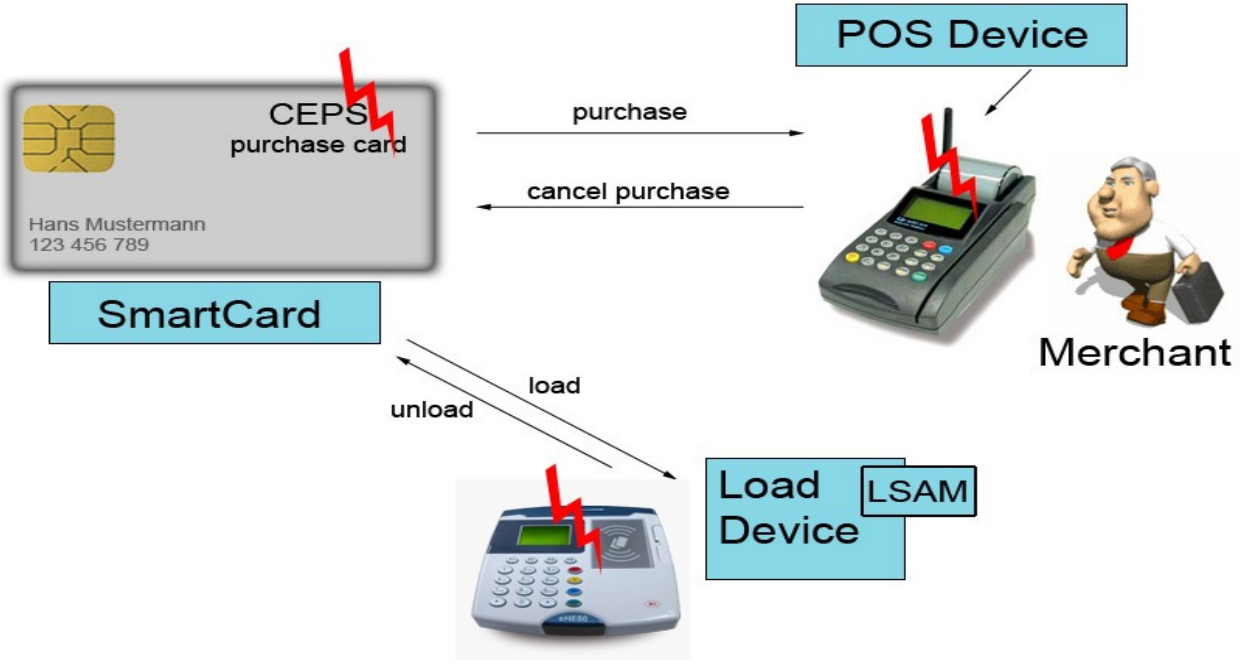
Beispielanwendung: Mobiles Bezahlsystem

Common Electronic Purse Specifications:
Globaler Standard für e-Geldbörsen (Visa et al.).

Smartcard enthält Kontostand, sichert Transaktionen mithilfe Krypto.

Formale Analyse von Load und Purchase Protokollen: signifikante Schwachstellen: Kauf-Umleitung, Betrug Ladegerätbetreiber vs. Bank.

[IFIPSEC 01, ASE 01]



Beispielanwendung: Biometrische Authentisierung

Smartcard basiertes System.

Analysiert mit UMLsec parallel zur Entwicklung durch Firma in gemeinsamem Projekt.

Entdeckten drei signifikante Schwachstellen in verschiedenen Versionen (Fehlbedienungszähler umgangen durch Löschen / Wiederholen von Nachrichten; Smartcard unzureichend authentisiert durch Mischen von Sitzungen).

Endgültig entwickelte Version sicher.

[ACSAC 05]

Beispielanwendung: Internes Informationssystem bei BMW

MetaSearch Engine: Personalisierte Suche im Firmen-Intranet (passwort-geschützt).

BMW Group

Einige Dokumente sehr sicherheitskritisch.

Über 1.000 potentielle Benutzer, 280.000 Dokumente, 20.000 Anfragen pro Tag.

Nahtlos in unternehmensweite Sicherheitsarchitektur integriert. Bietet Sicherheitsdienste für Anwendungen (Benutzerauthentisierung, rollenbasierte Zugangskontrolle, globales Single-Sign-On), Ansatzpunkte für weitere Sicherheitsdienste.

Erfolgreich mit UMLsec analysiert. [ICSE 07]

Thema 1: Sicherheitslücken in Cloud Computing Umgebungen

Auslagerungen von Prozessen und Diensten in Cloud Computing Umgebungen bietet nicht nur den Benutzern Möglichkeiten z.B. zur Kostenreduktion sondern auch Angreifern neue Angriffsvektoren z.B. die Möglichkeit durch Angriffe auf die Virtualisierung oder die Interfaces der Cloud unberechtigten Zugang zu erhalten. Der Vortrag soll eine Übersicht über bereits bekannte Angriffe auf Cloud Computing Umgebungen geben und dabei einzelne Angriffe detailliert vorstellen. Vorgestellt werden könnten z.B. cross-VM side-channel attacks, die dem Angreifer unberechtigten Zugriff auf Informationen anderer Cloud Computing Benutzer ermöglichen, oder XML Signature Element Wrapping attacks, die den Angreifer unberechtigt Befehle ausführen lassen.

Literatur: [RTSS09], [GJLS09], [ESA-02]
Ansprechpartner: Sebastian Pape

Thema 2: Homomorphe Verschlüsselung

LEHRSTUHL 14
SOFTWARE ENGINEERING

Die Auslagerung von Prozessen und Berechnungen in Cloud Computing Umgebungen birgt insbesondere in Hinsicht auf Datenschutz und Compliance zahlreiche Risiken. Eine Möglichkeit diesen Risiken entgegenzuwirken ist die Verschlüsselung von Daten in der Cloud Computing Umgebung. Wählt man ein homomorphes Verschlüsselungsverfahren, so können in der Cloud trotz der Verschlüsselung noch hinreichend sinnvolle Berechnungen ausgeführt werden. Der Vortrag soll homomorphe Verschlüsselung anhand des "ring learning with errors" (Ring LWE) Problems vorstellen und derzeitige Einsatzmöglichkeiten sowie Probleme beim Einsatz homomorpher Verschlüsselung aufzeigen.

Literatur: [O10], [LPR10], [LNV11]
Ansprechpartner: Sebastian Pape

Thema 3: Ontologien für Sicherheit und Compliance in Clouds

Cloud Computing hat sich zu einem echten Hype entwickelt. Gerade für kleinere und mittelständische Unternehmen ist es sinnvoll, Prozesse in eine Cloud auszulagern statt selbst in teure IT-Ressourcen zu investieren. Bei der Verarbeitung von vertraulichen Daten stellt sich jedoch zunehmend die Frage nach Sicherheitsanforderungen an die Cloud und inwiefern Compliance-Regularien beachtet werden.

In der Literatur sind bereits erste Ontologien und Referenzmodelle für Sicherheitsanforderungen und Compliance-Regularien vorgestellt worden. Im Vortrag soll grundlegend das Konzept von Ontologien und semantischen Netzen am Beispiel von Cloud Computing und Sicherheit bzw. Compliance erläutert und die bestehenden Ontologien diskutiert werden. (Das Thema kann ggf. auf zwei Teilnehmer ausgedehnt werden.)

Literatur: [GMT11], [MT11]

Ansprechpartner: Sven Wenzel

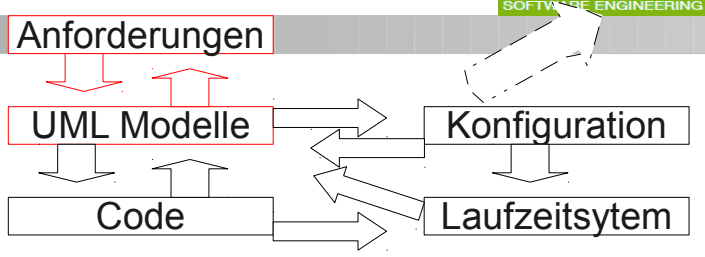
Thema 4: Artefakt- und Event-basierte Modellierung von Geschäftsprozessen

Geschäftsprozesse werden i.d.R. entweder Ereignis-basiert oder Artefakt-basiert modelliert. In diesem Vortrag soll eine Gegenüberstellung der beiden Ansätze erfolgen und eine mögliche Kombination beider diskutiert werden.(Das Thema kann ggf. auf zwei Teilnehmer ausgedehnt werden.)

Literatur: A. Nigam, N.S. Caswell; Business artifacts: an approach to operational specification, IBM Sys.J. 42(3), 2003

Ansprechpartner: Sven Wenzel

Thema 5: Modellierung mit UMLsec



Ziel:

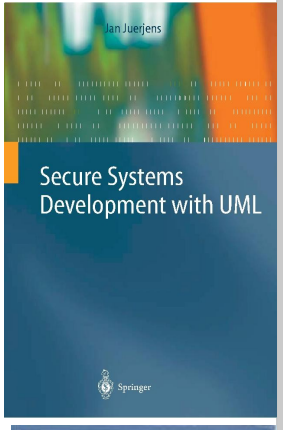
- Dokumentation und automatische Analyse von sicherheits-relevanten Informationen (z.B. Sicherheits-Eigenschaften und -Anforderungen) als Teil der Systemspezifikation.

Idee:

[FASE 01, UML 02]

- UML für System-Modellierung.
- Sicherheitsrelevante Informationen als Markierungen (Stereotypen) einfügen. Definiere dazu UML-Erweiterung UMLsec.
- Formale Semantik mit stromverarbeitenden Funktionen als Grundlage für Verifikation.

[Jour. Logic & Algebr. Program. '08]



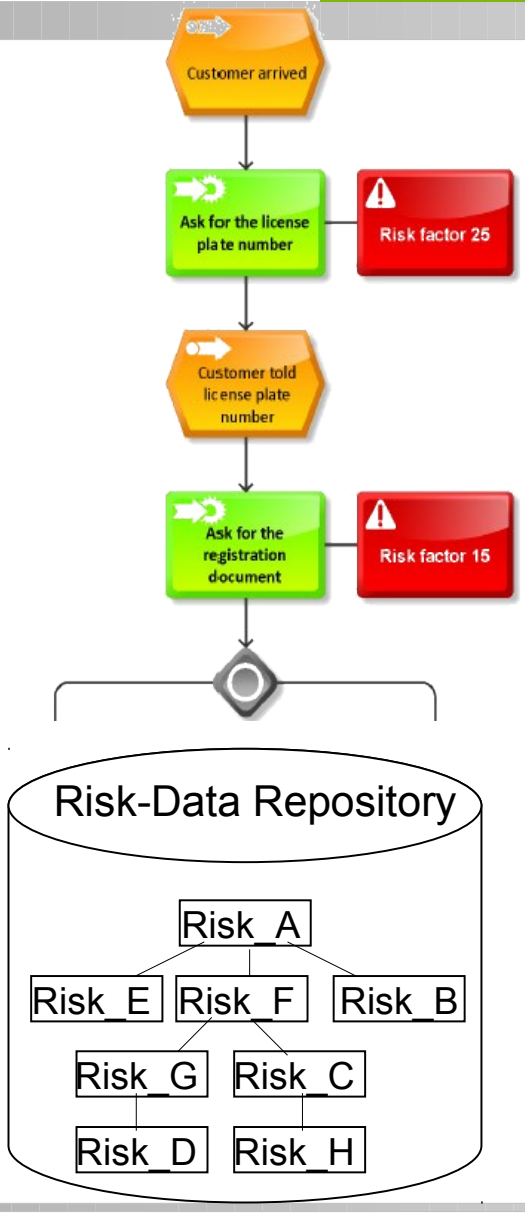
Thema 6: Governance, Risk & Compliance auf Geschäftsprozessen

LEHRSTUHL 14 SOFTWARE ENGINEERING

- Idee: Automatische Analyse von Geschäftsprozessmodellen auf operationale Risiken, z.B. gegenüber Benutzerberechtigungen zur Laufzeit, sowie der Benutzer-berechtigungen gegenüber der Sicherheitspolitik,
- automatische Risiko-Identifikation und -Bewertung
- Laufendes Projekt (Fraunhofer Attract): Architekturen für auditierbare Geschäftsprozessausführung (Apex).

[FASE '08]

[ICSE '08]



Die Aufgabenstellung geht von der Frage aus, welche Schäden (Art, Umfang, Häufigkeit) für die betriebsinternen Abläufe des Unternehmens durch Computer- und Internetkriminalität entstehen können und in welchem Umfang das Risiko solcher Schäden durch einen geeigneten finanziellen und technischen Sicherheitsaufwand gesenkt werden kann (Return on Security Investment, ROSI).

Es werden Methoden diskutiert, die dem Unternehmen eine Aussage über das vertretbare Volumen bei Investionen in das IT-Sicherheitsmanagement liefern. Im Rahmen des Bearbeitung soll daher ein Überblick über die verfügbaren Methoden gewonnen werden, die Sicherheit bzw. Gefährdung IT-gestützter Betriebsabläufe analysieren und IT-Risiken quantitativ bewerten.

Thema 8: UML-Maschinen und formale Sicherheit

LEHRSTUHL 14
SOFTWARE ENGINEERING

Für die formale Prüfung von Sicherheit werden häufig formale Modelle genutzt. Für UMLsec ist das formale Modell durch UML-Maschinen gegeben. Dieser auf State-Maschinen aufbauende Formalismus kann genutzt werden, um UMLsec eine Semantik zu geben. Gleichzeitig eignen sich diese Maschinen einen Angreifer zu modellieren. Die auf den Maschinen definierten Verfeinerungsbegriffe können anschließend genutzt werden, um Bedrohungen des UMLsec-Modells zu analysieren.

Dieses Thema soll zwischen zwei oder drei Teilnehmern aufgeteilt werden. Eine Kooperation, um die Vorträge aufeinander abzustimmen, wird erwartet. Eine natürliche Trennung ist gegeben durch die formalen Grundlagen der UML-Maschinen bzw. der aus ihnen zusammengesetzten Systeme und der Nutzung der Verfeinerung, um die Abwesenheit bestimmter Angriffsprobleme zu zeigen.

Literatur: Jürjens, J. Umlsec: Extending uml for secure systems development. UML 2002 —The Unified Modeling Language (2002).
Ansprechpartner: Thomas Ruhroth

Thema 9: Grundlagen des expliziten Model Checking

An viele der heute verwendeten Computersysteme werden Sicherheitsanforderungen gestellt, die allein mit testbasierten Verfahren nicht sichergestellt werden können. Model Checking bietet die Möglichkeit, für ein gegebenes System geforderte Eigenschaften formal nachzuweisen.

Ziel des Vortrags ist die Darstellung des grundsätzlichen Vorgehens beim Model Checking, einschließlich der möglichen Formalismen (insbesondere Kripke-Strukturen), und praktische Grenzen der Anwendbarkeit. Weiterhin soll mit CTL eine der für die Spezifikation genutzten Logiken vorgestellt werden.

Literatur: z.B. Clarke, Grumberg, Peled "Model Checking"

Thema 10: Model Checking mit LTL

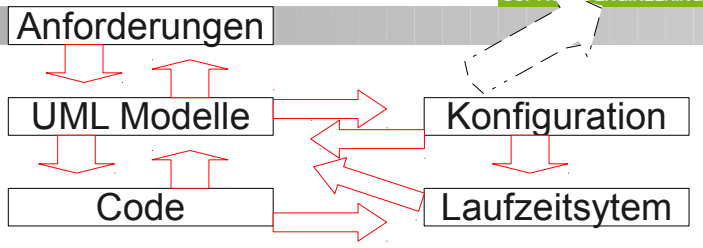
Die "Linear temporal logic" (LTL) stellt neben CTL eine weitere Möglichkeit dar, Anforderungen an ein System zu formalisieren. Diesen Formalismus vorzustellen ist Teil des Vortrags. Weiterhin sollen existierende Optimierungsmethoden erklärt und in ihren Eigenschaften verglichen werden, die den praktischen Einsatz von Model Checkern erleichtern bzw. teilweise erst möglich machen. Als Referenz kann hierbei das Programm "SPIN" dienen. SPIN ist ein Model Checker mit Unterstützung für LTL.

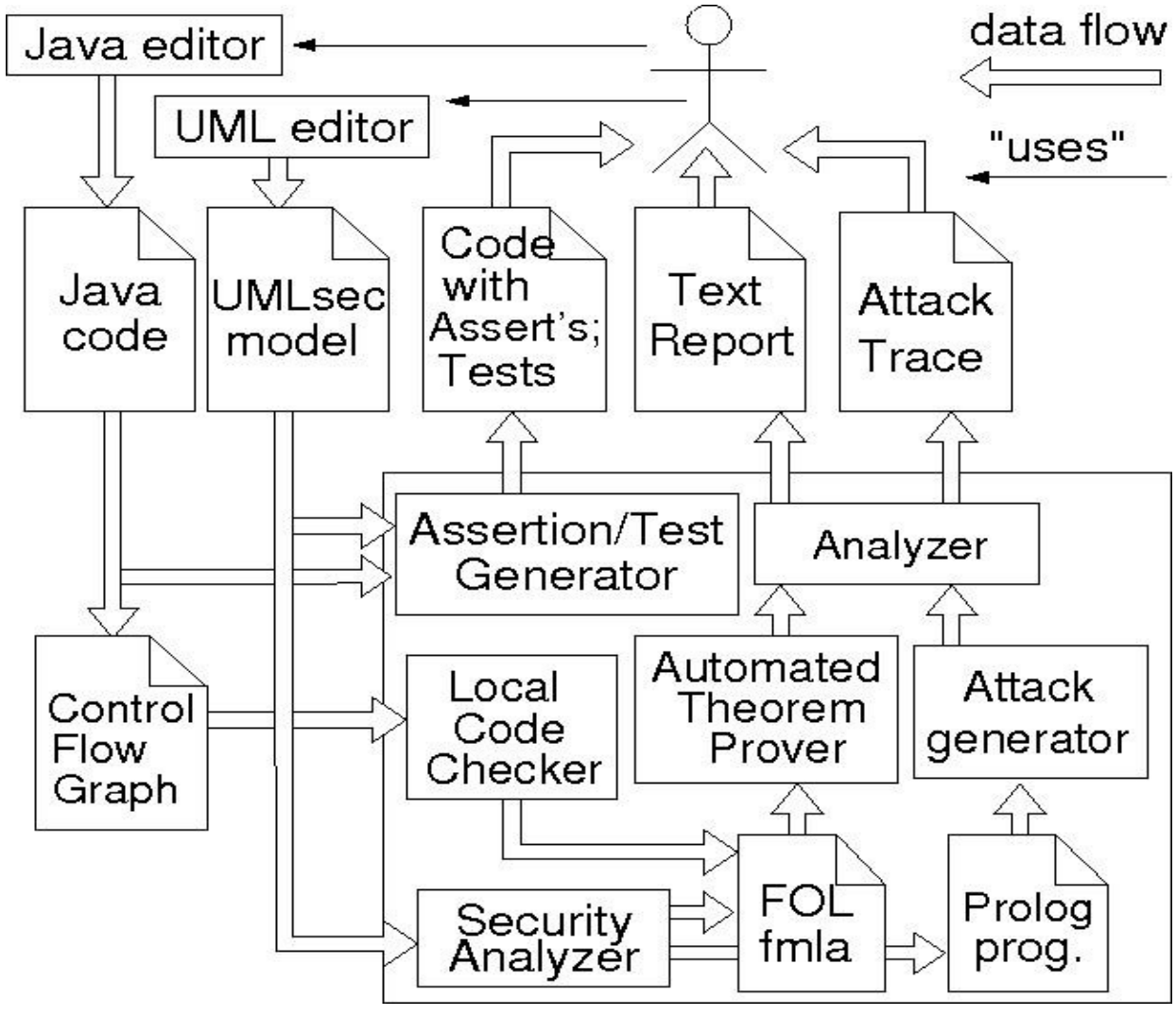
Literatur: z.B.

- Stephan Kleuker "Formale Modelle der Softwareentwicklung"
- Gerard Holzmann "The SPIN model checker"

Thema 11: Werkzeugunterstützung

LEHRSTUHL 14 SOFTWARE ENGINEERING





[UML 04, FASE 05, Jour. Softw. Tools & Techn. Transf. (STTT) 07]

Agenda

LEHRSTUHL 14
SOFTWARE ENGINEERING

- Vorgaben Ausarbeitung
- Vorgaben Vortrag
- Milestones
- Notenbildung
- Sonstiges Organisatorisches
- Themenvergabe

Vorgaben Ausarbeitung

- 15 Seiten Ausarbeitung
  - Ohne Inhalts- / Tabellen- / Abbildungsverzeichnis, Titelblatt, Bibliographie und Index
  - Min. 10 Seiten Reintext (Ohne Abbildungen / Kapitelumbrüche etc.)
  - Formatvorlagen verwenden und einhalten
    - Liegen im Latex und Word Format vor
    - http://www-jj.cs.tu-dortmund.de/jj/teaching/formatVorlagen
  - Fazit mit eigener kritischer Stellungnahme sollte enthalten sein

Vorgaben Ausarbeitung

- Gegebene Literatur ist Grundlage und Kernliteratur
  - Andere Quellen zulässig und gewünscht
- Wissenschaftliches Arbeiten
  - Quellenrecherche, -prüfung und -angaben
  - Fehlerhafte Quellenangaben oder Plagiatsversuche führen zur Abwertungen oder Ablehnung der Ausarbeitung
  - Formulierungen
- Abgabe elektronisch und 1 mal gedruckt
  - Gedruckte Version mit Unterschrift der ehrenwörtlichen Erklärung

Vorgabe Vortrag

LEHRSTUHL 14
SOFTWARE ENGINEERING

- Dauer 45 Minuten
  - Ist einzuhalten. Kleine Abweichungen in Ordnung
- Ziel ist es den anderen Seminarteilnehmern den Themenblock verständlich näher zu bringen
- Fokus auf die wichtigen Aspekte
- Anschließende Diskussion in der Seminarrunde
- Spezieller Medieneinsatz sollte abgesprochen werden
- Standardpräsentationsequipment wird vorhanden sein

Milestones

- 10.10.2011 Kickoff mit Themenvergabe
- 14.11.2011 Vorschlag Gliederung und Quellenverzeichnis
  - Erster Vorschlag für eine Gliederung der Seminararbeit
  - Ergebnis der Quellensuche als Quellenverzeichnis
  - Elektronisch als PDF an inhaltlichen Betreuer
  - Deadline : 14.11.2011 12:00 Uhr

Milestones

- 12.1.2012 Erste vollständige Version Ausarbeitung und Vortrag
  - Erste möglichst vollständige Version der Ausarbeitung und der Präsentation für den Vortrag
  - Elektronisch als PDF oder PPT an inhaltlichen Betreuer
  - Deadline : 12.01.2012 12:00 Uhr

Milestones

- So 12.02.2012 Finale Version Ausarbeitung und Vortrag
  - Finale Version der Ausarbeitung und der Präsentation für den Vortrag
  - Elektronisch als PDF oder PPT an inhaltlichen Betreuer
  - Deadline : 12.02.2012 24:00 Uhr
  - Ausarbeitungen und Folien werden allen Teilnehmern vor dem Seminar zur Verfügung gestellt
- Mi 15. + Do 16.02. 2012 Blockseminar
  - Genaue Uhrzeiten nach Absprache

Notenbildung

- Ausarbeitung 50%
  - Inhalt
  - Struktur
  - Verständnis
  - Form
  - Quellen
- Vortrag 40%
  - Verständlichkeit
  - Aufbau
- Teilnahme an Diskussionen 10%

Sonstiges Organisatorisches

LEHRSTUHL 14
SOFTWARE ENGINEERING

- Relevante Literatur wird vom jeweiligen inhaltlichen Betreuer zur Verfügung gestellt
- Nach Ende des Kickoffs bitte schnellstmöglich eine Mail mit 5 Themenwünschen (nach Priorität geordnet) und Kontaktdaten an

  jan.jurjens@cs.tu-dortmund.de

  - Name, Matrikelnummer, Studiengang, Semester
  - Bereits gehörte relevante Vorlesungen und Seminare
  - Gegebenenfalls weitere qualifizierende Vorkenntnisse
- Deadline: morgen, Di 11. Okt., 23.59 Uhr MESZ

Werbeblock

LEHRSTUHL 14
SOFTWARE ENGINEERING

Hiwi-Tätigkeiten

Es gibt verschiedene Möglichkeiten für eine Beschäftigung als Hiwi am Fraunhofer ISST oder am LS 14 / TUD:

- Unterstützung der folgenden Projekte (bespielsweise durch Java-Programmierung eines UML-Analyse Werkzeuges oder konzeptuelle Arbeiten im Bereich modell-basierte Sicherheitsanalyse): "Secure Change", "Architectures for Auditable Business Process Execution (APEX)", „SecureClouds“, „ClouDAT“
- Unterstützung in der Lehre (Tutorien, Folienerstellung etc)

Informationen unter: http://jan.jurjens.de/jobs/hiwis.html

Abschlussarbeiten (z.B. Bachelor)

Abschlussarbeiten können in inhaltlicher Beziehung zu einer Hiwi-Tätigkeit am Fraunhofer ISST oder LS 14 / TUD durchgeführt werden.

Sie können insbesondere in Zusammenhang mit Anwendungsprojekten am ISST durchgeführt werden, wodurch sich vielfältige Möglichkeiten zu Kooperation mit Unternehmen ergeben, zB:

- Apex: Versicherungen / Banken (Münchener Rückversicherung, Signal Iduna, Wüstenrot), Softwarehersteller (SAP, IDS Scheer)
- Secure Change: Telekom / Smartcards (Telefonica, Gemalto)
- Csec: Microsoft Research Cambridge
- Secure Clouds / ClouDAT: Cloud-Software-Anbieter (LinogistiX), IT-Berater (Admeritia, ITESYS, TÜV-IT)

Informationen unter: http://jan.jurjens.de/jobs/hiwis.html

Einige Beispiel-Themen für Abschlussarbeiten

- Formale Abbildung von regulatorischer Compliance auf Security Policies
- Modellierung und Automatische Sicherheits-Analyse für Cloud Computing Systems
- Business Process Mining
- Spezifikation von IT-Sicherheitszielen für die Geschäftsprozessmodellierung und deren Integration in die Ausführung im Workflow
- Design und Entwicklung einer Schnittstelle zwischen der Business Prozess Management Suite ARIS und dem Sicherheitsanalysetool UMLsec zur Compliance Analyse in der Versicherungsdomäne
- Generierung von Geschäftprozessen mit OpenArchitectureWare unter Berücksichtigung von Sicherheitseigenschaften
- Werkzeuggestützte Modell-basierte Sicherheitsanalyse
- Werkzeugunterstützte Analyse von sicherheitskritischen SAP-Berechtigungen im Finanzbereich
- Modell-basiertes Return on Security Investment (ROSI) im IT-Sicherheitsmanagement