

Modellbildung in der Anforderungsanalyse

Modellbildung in der Entwicklung
 Prof. Dr. Dr. h.c. Manfred Broy
 Gemeinsam mit Dr. Bernhard Schätz
 Fakultät für Informatik, TU München
 SS 2007

Fakultät für Informatik
 Lehrstuhl IV: Software & Systems Engineering

1

Inhalte

Vorlesungsinhalte:

1. Modellbegriff
2. Grundlagen
3. Anwendung
 1. Ontologien
 2. Anforderungsanalyse
 1. Grundlagen
 2. Modelle
 1. Datensicht
 2. Kontextsicht
 3. Funktionssicht
 4. Werkzeugbeispiel
4. Architekturen
5. Qualitätssicherung
6. Modellqualität
3. Modellgetriebene Entwicklung
4. Domänenspezifische Modellierung

Fakultät für Informatik
 Lehrstuhl IV: Software & Systems Engineering

2

Überblick

Anforderungsanalyse

- Interdisziplinär: Integration aller Beteiligten („Stakeholders“)
- Iterativ: Abstimmung der Anforderung aller Beteiligten
- Initial: Formale Grundlage für alle weiteren Entwicklungstätigkeiten

• **Integration in SE-Prozess:**

- Vorgabe Entwicklung
- Vorgabe Qualitätssicherung

• **Zentrale Aufgaben:**

- Herausarbeiten und Identifizieren
- **Spezifizieren und Dokumentieren**
- Analysieren und Validieren
- Verwalten

Fakultät für Informatik
 Lehrstuhl IV: Software & Systems Engineering

3

Fehlerarten im Entwicklungsprozess

Quellphase	Fatal	Schwer	Mittel	Leicht	Total
Anforderungen	5,0	5,0	3,0	2,0	15,0
Design	3,0	22,0	10,0	5,0	40,0
Umsetzung	2,0	10,0	10,0	8,0	30,0
Dokumentation	0,0	1,0	2,0	2,0	5,0
Fehlereinf.	0,0	2,0	5,0	3,0	10,0
Gesamt	10,0	40,0	30,0	20,0	100,0

Quelle: Jones, 1991, Applied Software Measurement

- Wann werden welche Fehler gemacht:
 - Fatal: Anforderungsdefinition
 - Schwer: Entwurf
 - Mittel, leicht: Entwurf, Umsetzung

Fakultät für Informatik
 Lehrstuhl IV: Software & Systems Engineering

4

Grundlagen

Aufgaben Anforderungsanalyse:

- Erfassung des Anwendungsgebietes
 - „Wozu dient das zu erstellende System?“
 - Unterstützte betriebliche Anläufe, gesteuerte technische Prozesse
- Abgrenzung des Leistungsumfanges
 - „Was leistet das zu erstellende System?“
 - Bereitgestellte Funktionen
- Einschränkung der Systemkonzeption
 - „Welchen Einschränkungen unterliegt das zu erstellende System?“
 - Technische Vorgaben, rechtliche Rahmenbedingungen

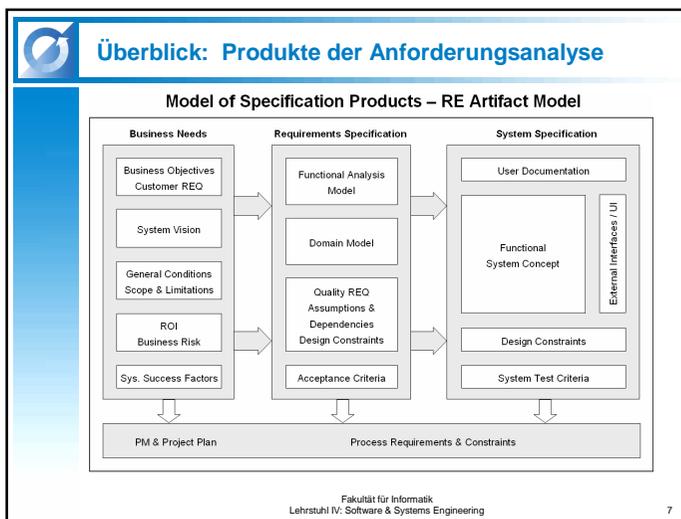
Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

Grundlagen

Hauptschritte Anforderungsanalyse:

1. Abgrenzung des Gegenstandsbereichs
2. Erfassung der Fachterminologie
3. Dokumentation von Nebenbedingungen
4. Erstellung eines fachlichen logischen Datenmodells
5. Erstellung eines fachlichen logischen Prozessmodells
6. Erfassung der (fachlichen) Schnittstellen
7. Erfassung der fachlichen Anforderungen

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering



Definition „Requirement“

A requirement is

- a condition or capability needed by a user to solve a problem or achieve an objective
- a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document
- a documented representation of a condition or capability as in (1) or (2)

Quelle: IEEE Std. 610.12-1990

Anforderungsspezifikationen:

- Abstrakt
 - Nutzerorientiert („Aus der Sicht des Nutzers“, „Black Box“)
 - Neutral („Unabhängig von Entwurfsentscheidungen“)
- Prüfbar:
 - Validierbar („Das richtige System entwickeln“)
 - Verifizierbar („Das System richtig entwickeln“)

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

Qualitätskriterien Anforderungsdefinition

Qualitätskriterien Anforderungsspezifikation (IEEE 830-1998):

- Konsistenz: Die Anforderungen sind in sich und untereinander widerspruchsfrei.
- Vollständigkeit: Die Anforderungen decken alle Eigenschaften des Zielsystems ab.
- Korrektheit: Die Anforderungen beschreiben die gewünschten Eigenschaften des Zielsystems.
- Eindeutigkeit: Die Anforderungen lassen keine mehrdeutige Interpretation zu.
- Verfolgbarkeit: Die Anforderungen lassen eine Rückverfolgung zur Quelle zu.
- Änderbarkeit: Änderungen an Anforderungen sind einfach umsetzbar.
- Prüfbarkeit: Die korrekte Umsetzung der Anforderungen im Zielsystem ist nachprüfbar.

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

Best Practices

Best Requirements Practices for System Software:

1. Formal Requirements
2. Requirements Inspection
3. Requirements Tracing
4. Performance Requirements
5. Quality/Reliability Requirements
6. Requirements Notations
7. Requirements Segmentation
8. Function Point Measurement
9. Defect tracking

Source: Jones, 2000, Software Assessments, Benchmarks and Best Practices

Formalisierte, strukturierte, integrierte Anforderungsmodelle

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

Sichten der funktionalen Spezifikation

3. Specific requirements 3.1 External interface requirements 3.1.1 User interfaces 3.1.2 Hardware interfaces 3.1.3 Software interfaces 3.1.4 Communications interfaces 3.2 Functional requirements 3.2.1 Mode 1 3.2.1.1 Functional requirement 1.1 . 3.2.1n Functional requirement 1n 3.2.2 Mode 2 . 3.3 Performance requirements 3.4 Design constraints 3.5 Software system attributes 3.6 Other requirements	3. Specific requirements 3.1 External interface requirements 3.1.1 User interfaces 3.1.2 Hardware interfaces 3.1.3 Software interfaces 3.1.4 Communications interfaces 3.2 System features 3.2.1 System Feature 1 3.2.1.1 Introduction/Purpose of feature 3.2.1.2 Stimulus/Response sequence 3.2.1.3 Associated functional requirements 3.2.1.3.1 Functional requirement 1 . 3.2.1.3n Functional requirement n 3.2.2 System feature 2 . 3.3 Performance requirements 3.4 Design constraints 3.5 Software system attributes 3.6 Other requirements	3. Specific requirements 3.1 External interface requirements 3.1.1 User interfaces 3.1.2 Hardware interfaces 3.1.3 Software interfaces 3.1.4 Communications interfaces 3.2 Functional requirements 3.2.1 Information flows 3.2.1.1 Data flow diagram 1 3.2.1.1.1 Data entities 3.2.1.2 Permitt processes 3.2.1.1.3 Topology . 3.2.2 Process descriptions 3.2.2.1 Process 1 3.2.2.1.1 Input quantities 3.2.2.1.2 Algorithm formula of process 3.2.2.1.3 Affected data entities .
---	---	--

Quelle: IEEE Standard 1998-830

Benötigte Sichten:

- Schnittstellen
- Systemdaten
- Betriebsmodi
- Funktionen
- Entwurfsbeschränkungen

Passende Modelle:

- Strukturmodelle
- Datenmodelle
- Verhaltensmodelle
- Ablaufmodelle
- Strukturmodelle

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

Hauptsichten in der Anforderungsanalyse

Hauptsichten in der SE-Anforderungsanalyse

- Datenmodellierung: Relevante Information
 - Konzeptuelle Sicht: Welche Informationen werden vom System beeinflusst/beeinflussen das System?
- Kontextmodellierung: Betroffene Umgebung
 - Prozesssicht: In welche Prozesse ist das System eingebettet?
 - Schnittstellensicht: Über welche Schnittstellen ist das System eingebettet?
- Funktionsmodellierung: Angebotene Leistung des Systems
 - Funktionssicht: Welche Funktionen bietet das System?
 - Nutzungssicht: Wie werden die Funktionen genutzt?
 - Verhaltenssicht: Wie werden die Funktionen integriert?

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

Datenmodellierung

Datenmodellierung: Konzeptuelles Datenmodell

- Modellierung der Konzepte der Anwendungsdomäne
- Fokussierung auf für fachlichen Prozess relevanten Objekte der Anwendungsdomäne
- Schwerpunkt: Modellierung betrieblicher Prozesse
- Charakterisierung der Struktur der relevanten Informationen
 - Aufbau der Daten
 - Beziehungen zwischen Daten
- Varianten
 - Datenrelationsmodelle: Daten und ihrer Beziehungen
 - Objektorientierte Modelle: Objektwelten

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

13

Konzeptmodellierung

Konzeptmodellierung

- Anwendungsschwerpunkt: Modellierung der Zusammenhänge fachlicher Informationen (in betrieblichen Informationssystemen)
- Aufgaben:
 - Klassifikation und Attributierung: Identifikation der Objekte der Prozesse in der Anwendungsdomäne
 - Assoziation und Rollenbildung: Festlegung von Beziehungen zwischen den Objekten der Anwendungsdomäne
 - Generalisierung und Spezialisierung: Identifikation von Gemeinsamkeiten der Objekte der Anwendungsdomäne
- Konzepte: Klasse, Assoziation/Aggregation, Restriktion, Rolle, Generalisierung, Attribut

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

14

Konzeptmodellierung

Modellierungsschritte:

1. Identifikation von Klassen
 - Beeinflussende/beeinflusste Objekten des fachlichen Prozesses
 - Eigenständige Struktur, langlebig
2. Identifikation von Assoziationen
 - Fachliche Beziehungen zwischen Objekten
 - Rollenorientiert, langlebig
3. Identifikation von Aggregationen und Restriktionen
 - Rangordnungen oder semantische Zusammenhänge
 - Eigenständige Struktur, invariant

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

15

Konzeptmodellierung

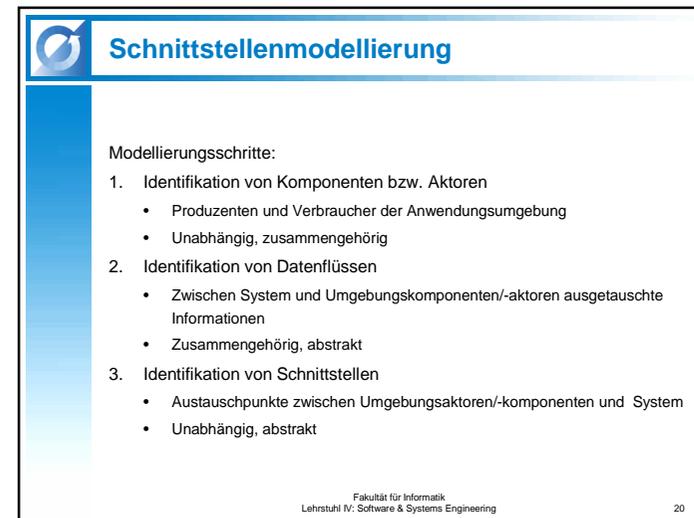
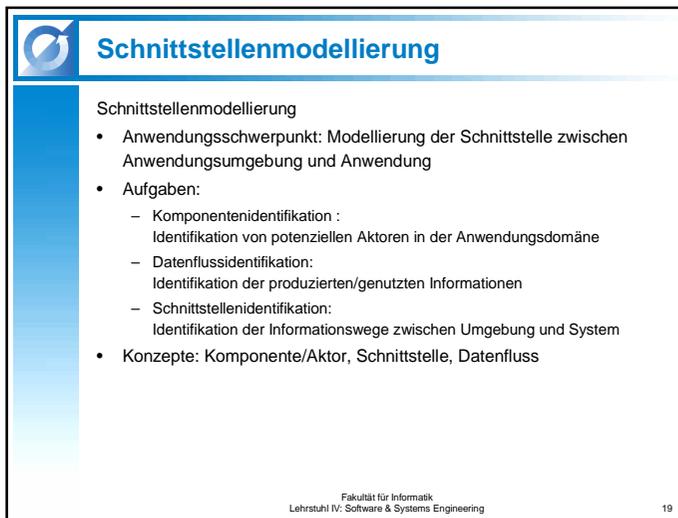
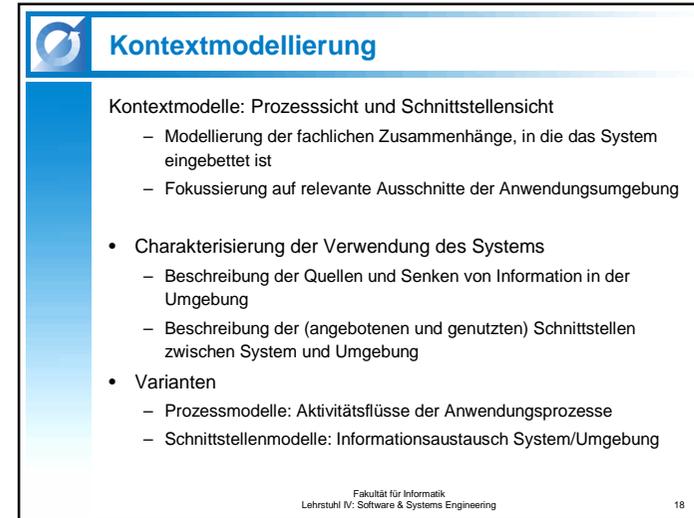
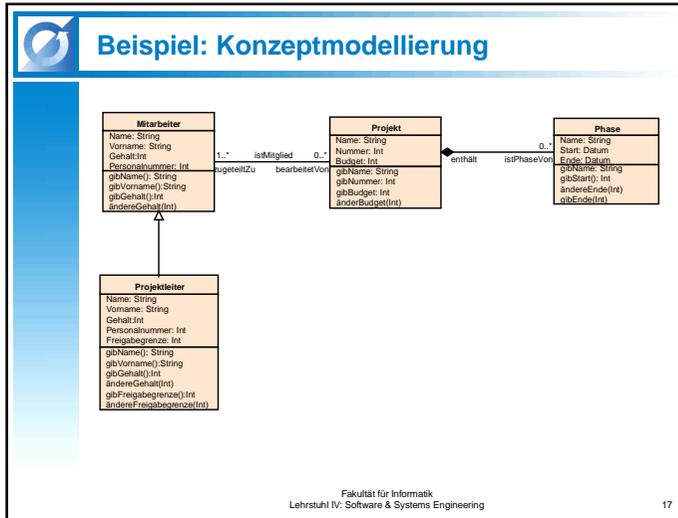
4. Identifikation von Attributen
 - Problemrelevante Eigenschaften von Anwendungsobjekten
 - Objektspezifisch, lösungsunabhängig
5. Identifikation von Generalisierungen
 - Gemeinsamkeiten von Objekten der Anwendungsdomäne
 - Minimal, invariant

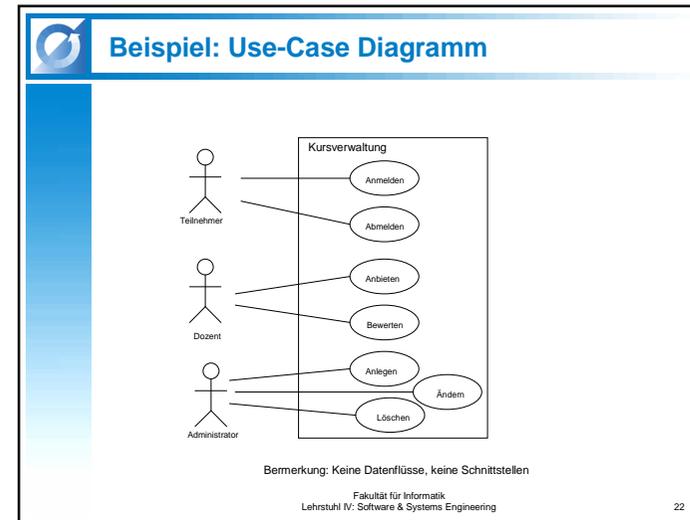
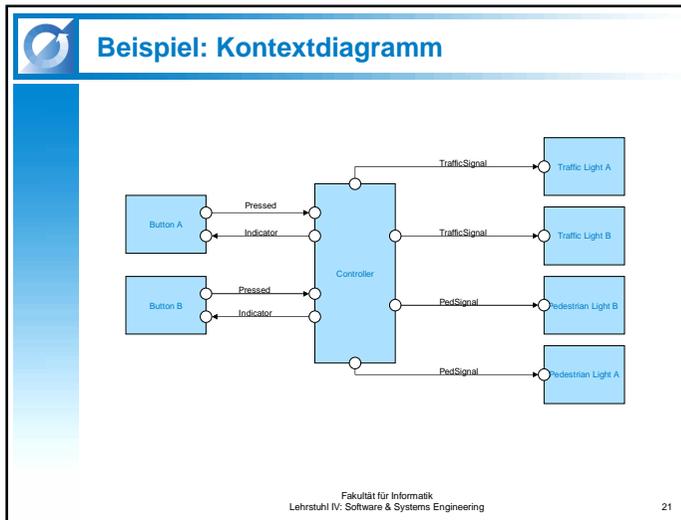
Potentiell Zusätzlich:

- Identifikation von Operationen
- Identifikation des Objektlebenszyklus

Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering

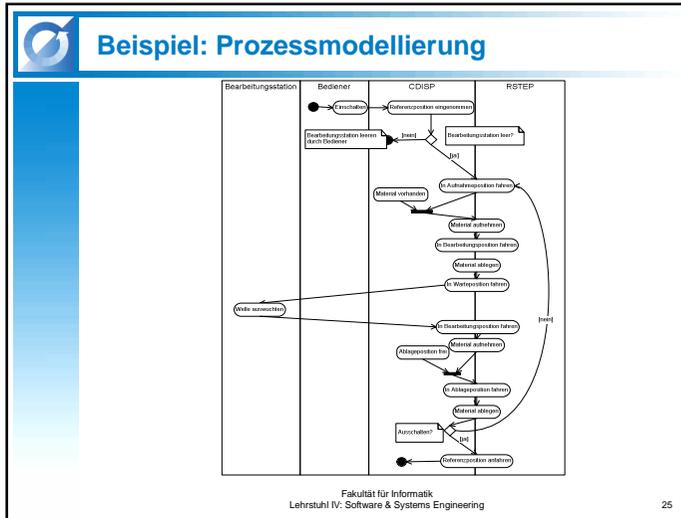
16





- ### Prozessmodellierung
- Prozessmodellierung
- Anwendungsschwerpunkt: Modellierung des betrieblichen/technischen Prozesses der Anwendungsdomänen
 - Aufgaben:
 - Identifikation Akteure:
Festlegung der Akteure im Anwendungsprozess
 - Identifikation Aktivitäten:
Festlegung der möglichen Aktivitäten der einzelnen Akteure
 - Identifikation Abhängigkeiten:
Festlegung der Synchronisierungen zwischen den Aktivitäten
 - Konzepte: Akteur, Aktivität, Aktivitätsfluss, Synchronisierungspunkt, Entscheidungspunkt
- Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering
- 23

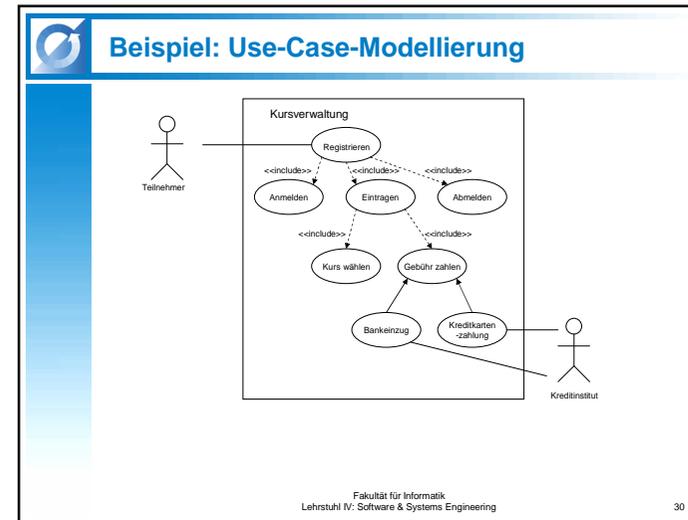
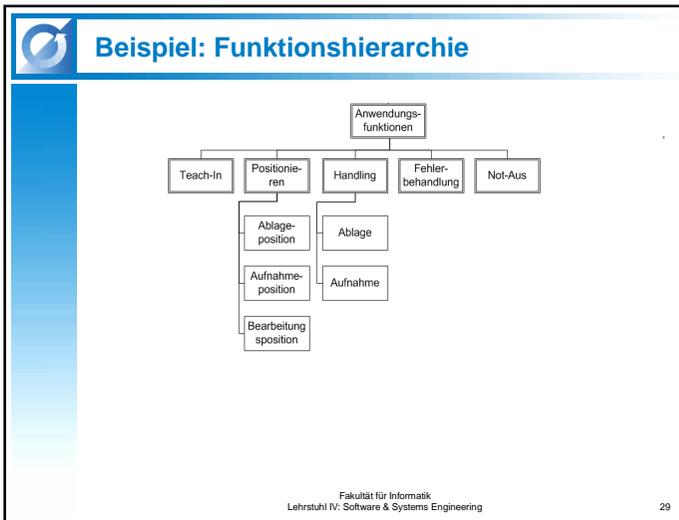
- ### Prozessmodellierung
- Modellierungsschritte:
1. Identifikation Akteure
 - Beeinflussende/beeinflusst Objekte des fachlichen Prozesses
 - Unabhängig, abstrakt
 2. Identifikation Aktivitäten
 - Potenzielle Aktivitäten der am Anwendungsprozess beteiligten Akteure
 - Abstrakt, atomar
 3. Identifikation Synchronisierungen
 - Abhängigkeiten zwischen auslösende und ausgelösten Aktivitäten
 - Unabhängig, atomar
- Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering
- 24



- ### Funktionsmodellierung
- Funktionsmodelle: Funktions-, Nutzungs und Verhaltenssicht
- Modellierung der vom System bereitgestellten Funktionen zur Erfüllung der Aufgaben des Anwendungsgebietes
 - Beschränkung auf die Nutzungsschnittstelle des Systems
 - Vermeidung von Vorwegnahme von Entwurfsentscheidungen
 - Charakterisierung die Verwendung des Systems
 - Beschreibung des Aufbaus der Gesamtfunktionalität des Systems
 - Beschreibung Nutzung der Einzelfunktionalitäten des Systems
 - Beschreibung des Zusammenspiels der bereitgestellten Funktionen
 - Varianten
 - Funktionsstrukturmodelle: Funktionskompositionshierarchie
 - Nutzungsfallmodelle: Funktionsanwendungsszenarien
 - Verhaltensmodelle: Zustands/ereignisbeschreibungen
- Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering
- 26

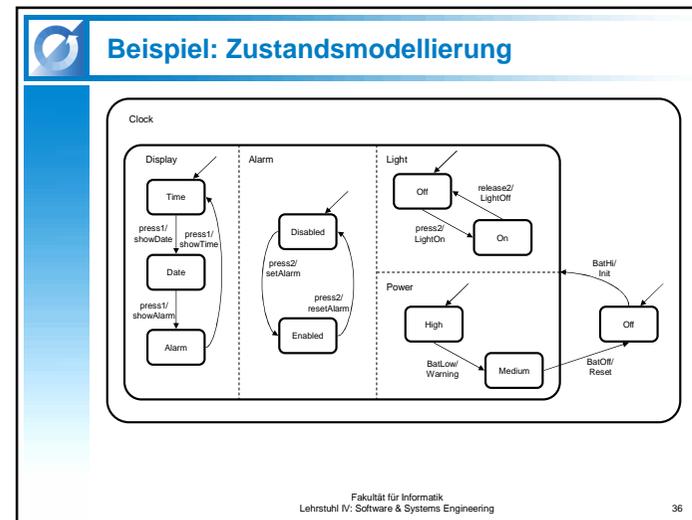
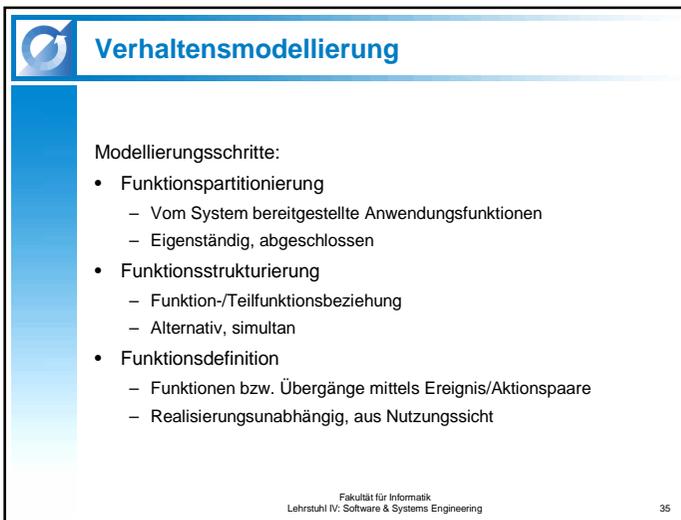
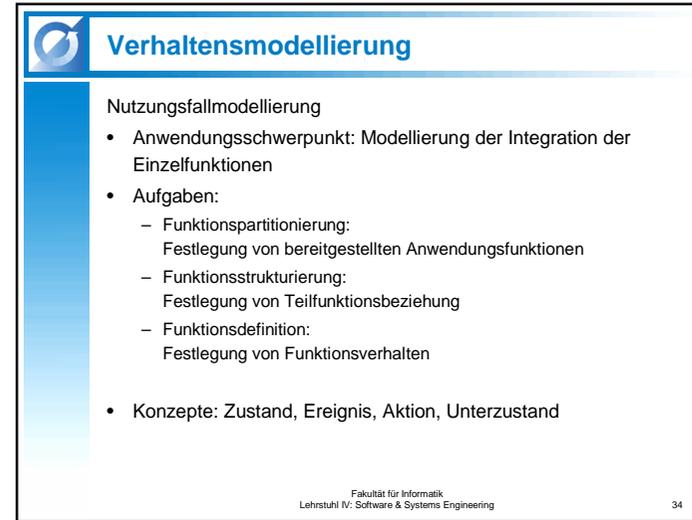
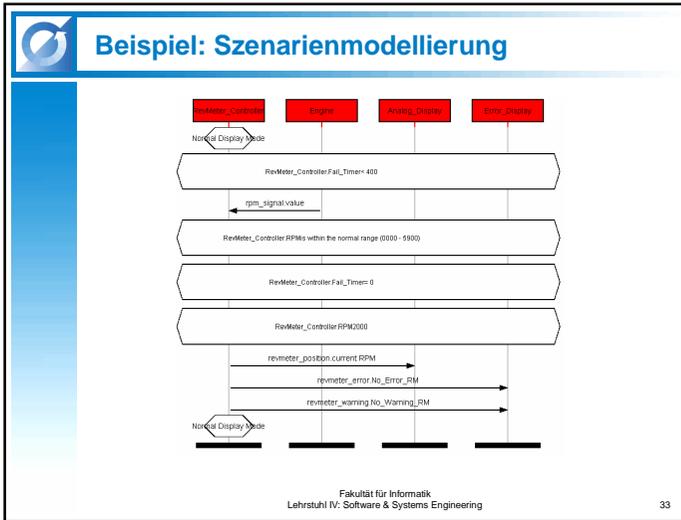
- ### Funktionsstrukturmodellierung
- Funktionsstrukturmodellierung
- Anwendungsschwerpunkt: Modellierung des Aufbaus der Systemgesamtfunktionalität
 - Aufgaben:
 - Funktionsidentifikation :
Identifikation von bereitgestellten Nutzungsfunktionen
 - Funktionsstrukturierung:
Identifikation von Unterfunktionen der Gesamtfunktionalität
 - Konzepte: Funktion, Funktionskomposition
- Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering
- 27

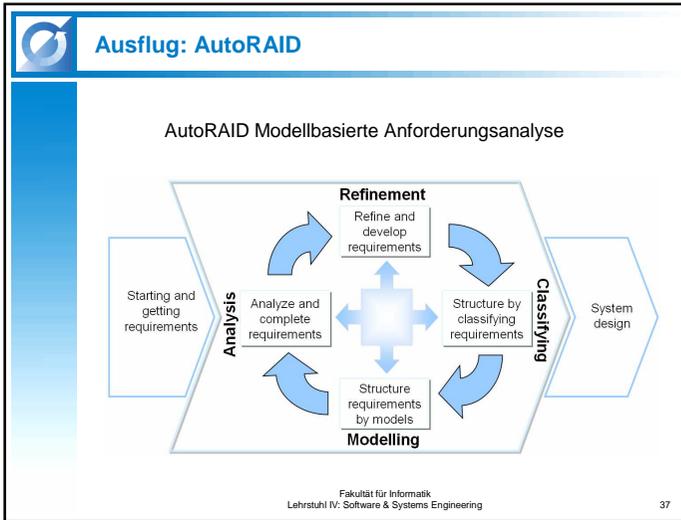
- ### Funktionsstrukturmodellierung
- Modellierungsschritte:
- Identifikation der Nutzungsfunktionen
 - Funktionalitäten zur Erfüllung eines fachlichen Ziels im Anwendungsprozess
 - Fachlich zusammenhängend, realisierungsneutral
 - Identifikation der Funktionshierarchie
 - Teilfunktionalitäten und Funktionsvarianten
 - Konjunktiv, disjunktiv
- Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering
- 28



- ### Nutzungsfallmodellierung
- Nutzungsfallmodellierung
- Anwendungsschwerpunkt: Modellierung des Aufbaus der Systemgesamtfunktionalität
 - Aufgaben:
 - Identifikation Akteure:
Festlegung von teilnehmenden Akteuren der Anwendungsdomäne
 - Definition Vorbedingung und Nachbedingung:
Festlegung von Voraussetzungen und Resultaten der Nutzung
 - Spezifikation Standardfall und Erweiterungen:
Festlegung von Einzelschritten der Funktionsnutzung
 - Konzepte: Akteur, (Vor-/Nach-)Bedingung, Interaktion, Nutzungsschritt
- Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering 31

- ### Nutzungsfallmodellierung
- Modellierungsschritte:
1. Identifikation der Akteure
 - Akteure, die an der Abwicklung des Nutzungsfalls beteiligt sind
 - Verursachende oder betroffene Akteure
 2. Identifikation der Vorbedingung
 - Notwendige Voraussetzung zur Abwicklung des Nutzungsfalls
 - Realisierungsunabhängig, aus Nutzungssicht
 3. Identifikation der Nutzungsszenarien
 - Sequenz von Einzelschritten des Standard/Ausnahme-Nutzungsfalls
 - Realisierungsunabhängig, aus Nutzungssicht
 4. Identifikation der Nachbedingung
 - Sichergestelltes Resultat aus der Abwicklung des Nutzungsfalls
 - Realisierungsunabhängig, aus Nutzungssicht
- Fakultät für Informatik
Lehrstuhl IV: Software & Systems Engineering 32





AutoRAID user interface

Architectural Constraint Display RPM

Type: Architectural Constraint Identifier: 7

Title: Display RPM

Description: The display RPM consists of a scale and a damped pointer showing the actual revolutions per minute of the engine.

Patron: Oelbergel

Status: Suggested Priority: Low

Superrequirements	Subrequirements	Mode/Views	Motivated Components
Identifier	Title	Patron	Document
1	Comfortable view...	Jun	Spec-Rev-Meter...
31	Rev Meter	Jun	

Fakultät für Informatik
 Lehrstuhl IV: Software & Systems Engineering

38