



AK Traceability  
Workshop 07.12.07  
"Prüfung von Traceability Links"

zu Gast bei TU Darmstadt  
Fachgebiet Echtzeitsysteme  
Prof. Dr. rer. nat. Andy Schürr



# Ablauf

- 10.00 Begrüßung und Vorstellung der Teilnehmer
- 10.30 Einführung, Erörterung der Thematik
- 11.00 Integration and Automatic Maintenance of Traceability  
Links between System Development Tools (Andy Schürr)
- 12.00 Mittagspause
- 13.00 Artefakte, Linktypen und Besonderheiten von  
OOSE/RUP (Matthias Riebisch)
- 14.00 Model-based Requirements Engineering with  
AutoRAID/AutoFocus (Eva Geisberger)
- 15.00 Web of Models WOM (Tobias Zimmermann)
- 16.30 Festlegung der nächsten Schritte
- 17.00 Abschluss des Workshops



# Traceability Links

- verbinden Artefakte und Elemente aus verschiedenen Entwicklungsphasen
- beschreiben Spur eines Entwicklungsvorgangs
- drücken Abhängigkeiten und Entwurfsentscheidungen aus
- Notwendig für Verständnis, Prüfung von Änderungen, Analyse von Abhängigkeiten



# Anforderungen an Links

- Vollständigkeit, Korrektheit & Widerspruchsfreiheit
  - Menschen müssen Erstellung, Aktualisierung und Verifikation durchführen
- Verbinden von Artefakten mit hohem Detaillierungsgrad
  - Hohe Anzahl von Links

**Hoher Aufwand für Traceability**



# Traceability – State of the Art

- Definition von Kategorien und Typen
- Bezogen auf Requirements Engineering Aktivitäten
- Fehlende Integration in Entwicklungsmethoden und -Prozesse
- Fehlende Definitionen von Syntax und Semantik
- Regeln nur bezogen auf Syntax, semantische Definitionen nicht genügend formalisiert
- Tool-Unterstützung nur sehr begrenzt möglich



# Kriterien für Links

- Wie erstellen  
(Notwendigkeit ermitteln)
- Wie Korrektheit prüfen
  - Ist eine Beziehung zwischen zwei Artefakten erlaubt
    - Syntax-basierte Regeln:
      - kontextfreie Syntax
      - kontextsensitive Syntax
    - Aus Metamodell ableiten
  - Ist eine bestimmte Beziehung korrekt
  - Existiert eine Abhängigkeit
  - Gibt es eine Beziehung
  - Ist die Entwurfsentscheidung Korrekt
    - Semantik-basierte Regeln:
      - statische Semantik
      - dynamische Semantik
    - Aus Entwurfsregeln ableiten



# Prüfung bezüglich Korrektheit und Vollständigkeit

- Unterstützung durch Links, wenn korrekt und vollständig
- Kriterien unterschiedlich je nach Modelltyp, Entwicklungsmethode und Domäne
- Regeln (formal) definieren für Werkzeugunterstützung
- Beziehungstypen und definierte Semantik als Voraussetzung



# Herausforderungen zu Traceability

- Standard und Metamodell für Link-Semantik
- Integration in alle Entwicklungsphasen
- Erhebungstechniken für Text, Modelle und Code
- Integration in Entwicklungswerkzeuge
- Integration mit Change-Management-Methoden und Versionierung
- Methoden für Link-Nutzung : Verfolgung, Darstellung, Bewertung
- Regeln für Detaillierungsgrad von Links
- Best Practices für Anwendung in Domänen
- Akzeptanz, Kenntnisse, Motivation and Kontrolle für Anwendung in Praxis





# Offene Fragen

- Motivation und Nutzen von Traceability
- Kosten-/Nutzen-Verhältnis beim Einsatz von Traceability
- Welche Nutzung: Visualisierung, Auswertung, Impact Analyse ....
- Aufwand zur Erstellung von Pflege reduzieren
- Wie findet man Traceability-Links; sind die vorhandenen Methoden / Tools leistungsfähig genug
- Sisyphus vorstellen und vergleichen: Nutzung der Links
- Versionierung von (verbundenen) Modellen
- Wie Abgleich von Modellen durchführen
- Was gehört zum Modell, was ist außerhalb, aber Traceability
- n-näre Verbindungen als Links, z.B. Links zwischen Links



# Nächste Schritte

- Nächster Workshop zum Thema „Kosten-/ Nutzen-Verhältnis beim Einsatz von Traceability“
- Untersuchung der Anwendung von Traceability (was ist der Nutzen)
- Aspekt der Evolution von Modellen
- Versionierung von Modellen und Beziehungen