

# Artefakte, Linktypen und Besonderheiten von OOSE/RUP

---

Matthias Riebisch

TU Ilmenau

Workshop AK Traceability – 07.12.2007 Darmstadt

# Eigenschaften von Traceability Links

---

## Obligatorisch:

- Identifier
- Startelement (mit Typ und Kontext)
- Endelement (mit Typ und Kontext)
- Type des Links
- Entwurfsentscheidung (mit Zielen, Alternativen, deren Bewertung)

## Optional:

- Sicherheit bezüglich Korrektheit (für Reverse Engineering)
- Eigentümer / Ersteller (für Auflösung von Widersprüchen)
- Priorität: Bedeutung (für Ermittlung der Auswirkungen)

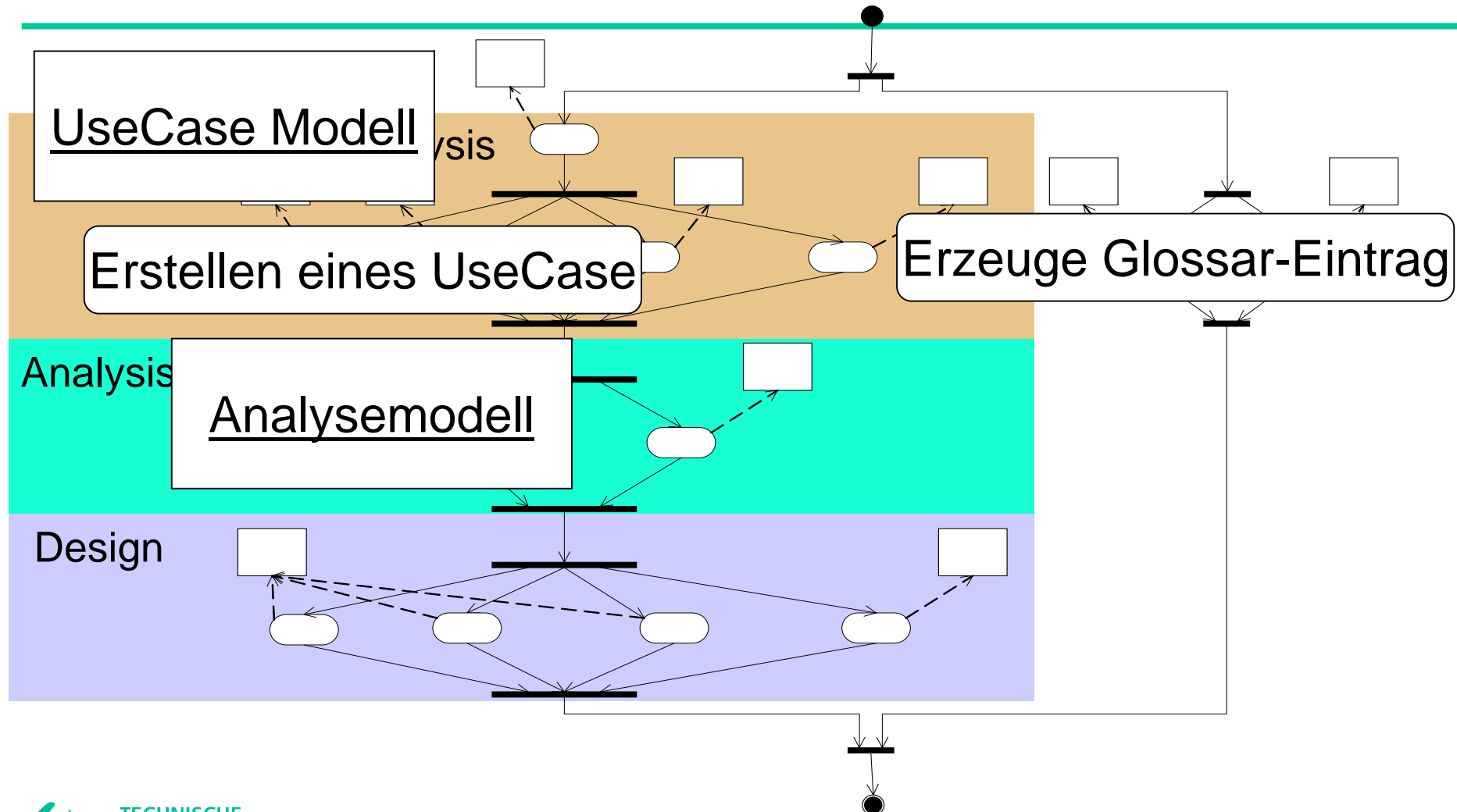
# Entwicklungsmethodik in OOSE und RUP

---

- Jacobson, I.: Object-Oriented Software Engineering: A Use Case Driven Approach. Addison Wesley, 1992
- Kern des Rational Unified Process, 2000 .. 2005
- Schwerpunkt auf (objektorientierter) Analyse
- Weniger Unterstützung für Architekturentwicklung, Integration und Deployment

# OOSE/UP

## Requirements Analysis



# Artefakte in der Anforderungsanalyse

---

- Vision Document
  - *realize* zwischen Need und Feature
- Software Requirements Specification (SRS)
  - Funktionale und nichtfunktionale Requirements
- Glossar
  - Begriffe und Erläuterungen
- Domain Object Model (DOM)
  - Akteur, Objekt, Prozess
- Interface Description
  - *refine*

# Artefakte in der objekt-orientierten Analyse

---

- Analyse-Klasse
  - Entitäten-, Schnittstellen-, Steuerungs-Klasse
- Paket
- Use-Case Realisierungs-Analyse
- Relation zwischen Analyse-Klassen
- Analyse-Modell
- Architekturbeschreibung

# Artefakte im objekt-orientierten Entwurf

---

- Entwurfs-Klasse
- Use-Case Realisierung -> Entwurf
- Subsystem, Komponente
- Entwurfsmodell

# Link-Typen

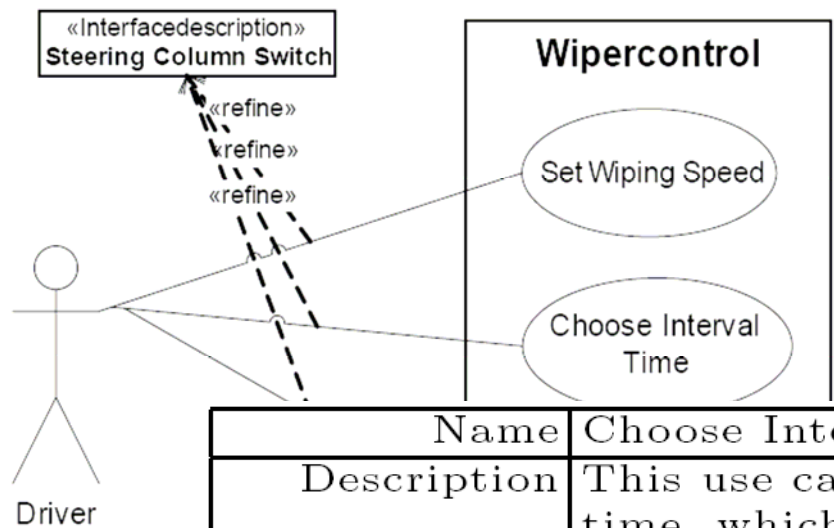
---

- *refine* – Verfeinerung eines Elements
- *realize* – Umsetzung von Problem zu Lösung
- *verify* – zwischen Definition und Prüfung
- *define* – zwischen Definition (z.B. Glossar) und Benutzung / Bezug

Weitere Link-Kategorien: implizit / explizit, ...



# Use Case and Description



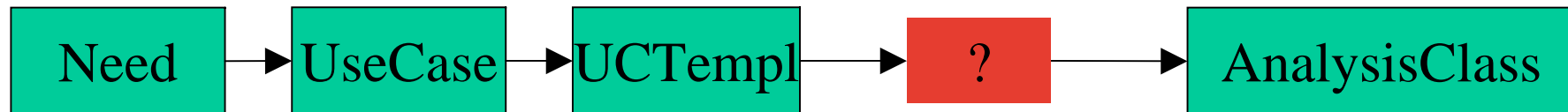
Use case description by natural language

Name	Choose Interval Time
Description	This use case allows the driver to set a new interval time, which is waited between two wipes.
Actors	Driver
Rationale	Steering column switch has been set to position interval.
Precondition	Clamp 15 is active and the steering column switch has position OFF.
Normal Flow	<ol style="list-style-type: none"> <li>1 The driver switches the steering column switch to position INTERVALL.</li> <li>2 The driver switches the steering column switch to position OFF.</li> <li>3 The driver switches the steering column switch after not more than 30s to position INTERVALL.</li> <li>4 The system has to set the new interval time as the time the steering column switch has been in position OFF.</li> </ol>
Altern. Flow	no
Postconditions	no

# Transform Use Case to Analysis Model

---

- Object-oriented analysis based on the experience of the developer, steps not precisely defined
- Natural language exploitation with ambiguity



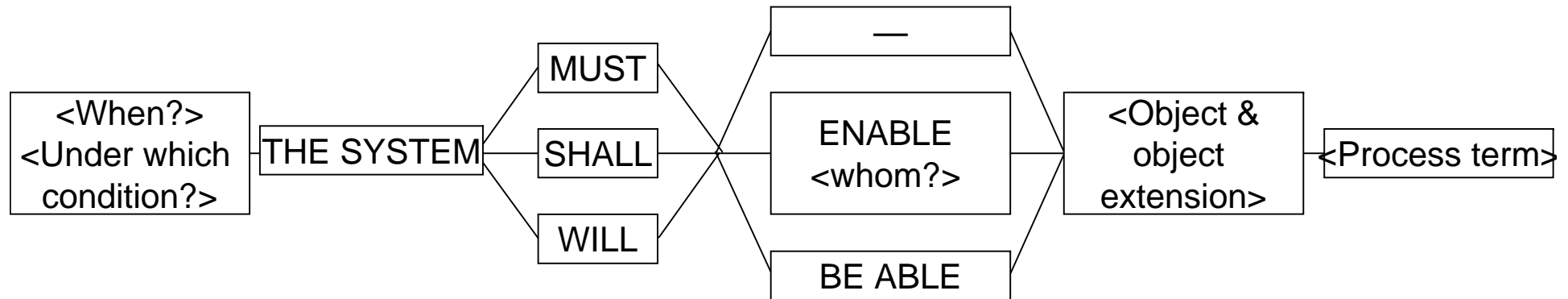
- Links from use case to analysis model cannot be established automatically
- Only one rule for traceability links:  
Each analysis class is related to at least one use case

# Refinement of OOSE/UP #1

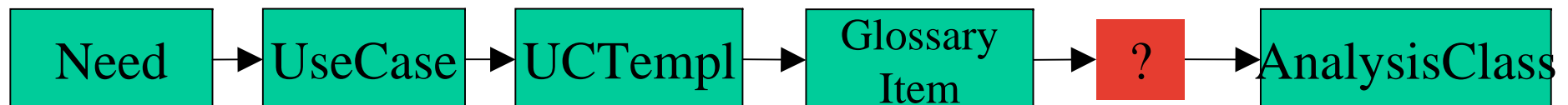
## Using a Text Template Method

---

- Formalization of natural language
- SOPHIST method for requirements engineering



- Adaptation for Formalizing description of sequences in use case descriptions
- Relation to glossary items and identifiers of model elements facilitate links
- Rules for glossary items used to maintain link correctness



# Refinement of OOSE/UP #2

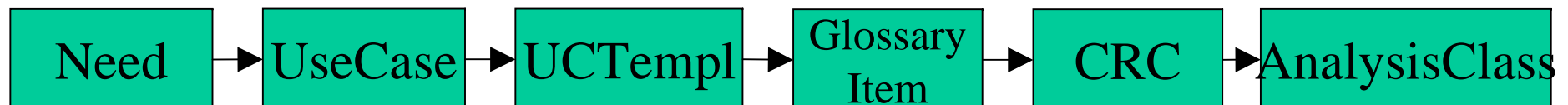
## Using the CRC Card Method

---

- Verbs and nouns are not enough for defining classes  
...

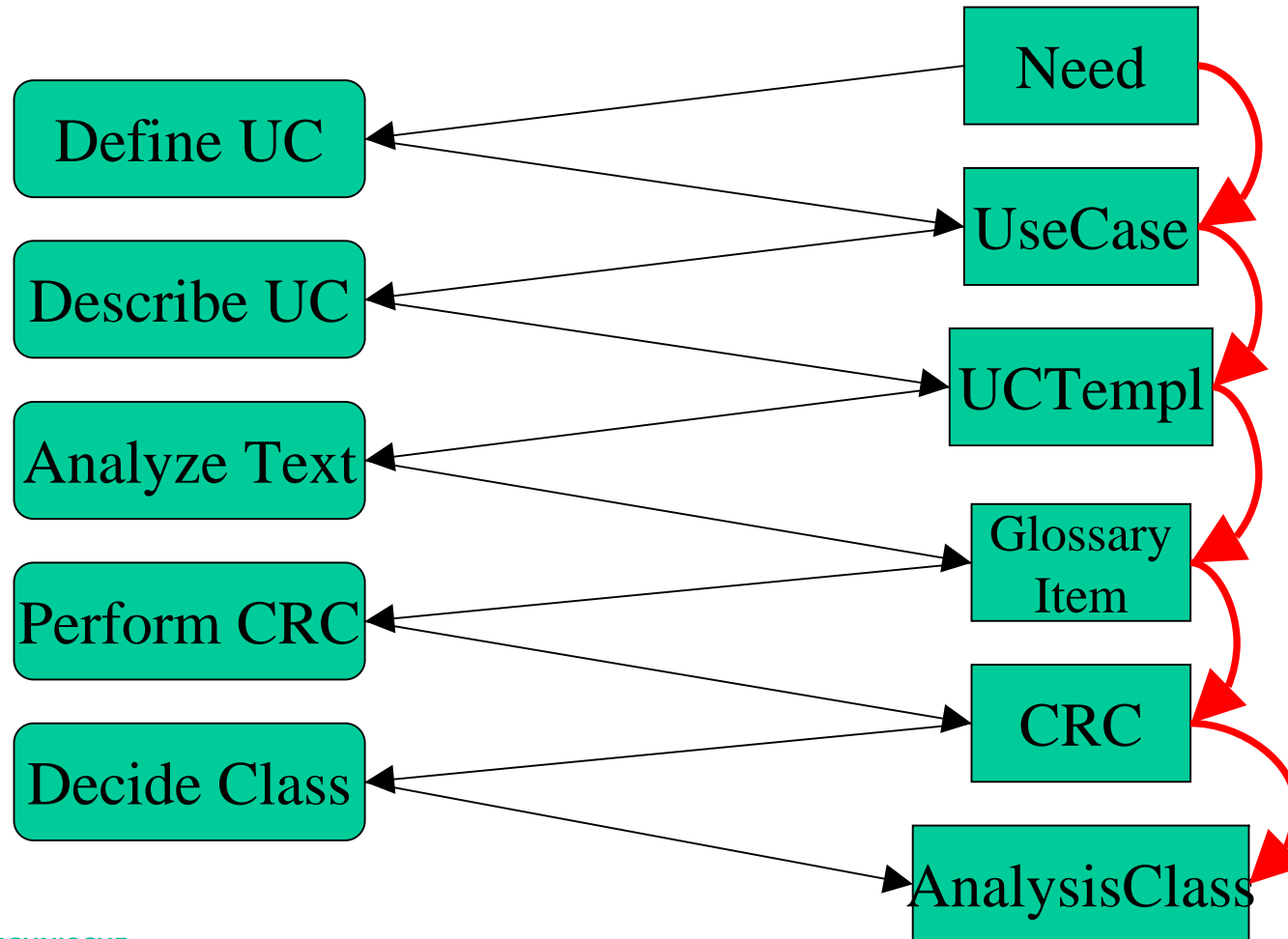
CRC: Class Responsibility Collaboration card Method  
(Cunningham 89)

- Structure responsibilities on cards
- Arrange clusters, perform scenarios, refine and combine ...
- Each card represents a candidate for a class



# Establishment of a Link Originating From a Use Case

---



# Rules and Heuristics

---

- Syntax-related rules restricting possible target elements by type
- Semantics-based rules
- Heuristics
- Use case flow activity to glossary terms of types ACTOR, INTERFACE, ACTION, STATE
- Glossary terms directly related via actor
- Decision for “good” terms



# Diskussion ....

---

**Quelle:** Patrick Mäder, Ilka Philippow, Matthias Riebisch: Customizing Traceability Links for the Unified Process. In Proceedings Third International Conference on the Quality of Software-Architectures (**QOSA2007**), Medford MA, USA, July 12-13, 2007. Springer: LNCS, 2007 (post-conference proceedings in press)

<http://www.theoinf.tu-ilmenau.de/~riebisch>