

# Teil V: Werkzeuge für spezifische Zwecke

## 70. Werkzeuge zur Anforderungsanalyse

1

Prof. Dr. rer. nat. Uwe  
Aßmann

Institut für Software- und  
Multimediatechnik

Lehrstuhl Softwaretechnologie

Fakultät für Informatik

TU Dresden

<http://st.inf.tu-dresden.de>

Version 12-1.0, 23.01.13

- 1) Requirements Management
- 2) Requisite Pro
- 3) Ontology-Driven  
Requirements Engineering  
(ODRE)
- 4) Traceability to other Artefacts

# References

2

- ▶ Katja Siegemund, Edward J. Thomas, Yuting Zhao, Jeff Pan, and Uwe Assmann. Towards Ontology-driven Requirements Engineering. Semantic Web Enabled Software Engineering (SWESE) Workshop at ISWC 2011, Koblenz
  - <http://iswc2011.semanticweb.org/fileadmin/iswc/papers/workshops/swese/4.pdf>
- ▶ [Mylopoulos1999] John Mylopoulos, Lawrence Chung, and Eric Yu. From Object-oriented to Goal-oriented Requirements Analysis. Communications of the ACM, 42(1):31-37, 1999.
- ▶ [Zowghi2002] Didar Zowghi and Vincenzo Gervasi. The Three Cs of Requirements: Consistency, Completeness, and Correctness. In Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality, (REFSQ'02), 2002.
- ▶ [Lamsweerde2000] Axel van Lamsweerde. Requirements Engineering in the year 00: A Research Perspective. In International Conference on Software Engineering, pages 5-19, 2000.
- ▶ Grady, Robert; Caswell, Deborah (1987). Software Metrics: Establishing a Company-wide Program. Prentice Hall. pp. 159. ISBN 0-13-821844-7.

# Tool References

## 3 ▶ [RPro] Requisite Pro User's Guide

- [ftp://ftp.software.ibm.com/software/rational/docs/v2003/win\\_solutions/rational\\_requisitepro/req\\_pro\\_user.pdf](ftp://ftp.software.ibm.com/software/rational/docs/v2003/win_solutions/rational_requisitepro/req_pro_user.pdf)

## ▶ Dominic Tavassoli, IBM Software. Requirements Definition and Management - Ten steps to better requirements management. June 2009

- [ftp://ftp.software.ibm.com/software/emea/de/rational/neu/Ten\\_steps\\_to\\_better\\_requirements\\_management\\_EN\\_2009.pdf](ftp://ftp.software.ibm.com/software/emea/de/rational/neu/Ten_steps_to_better_requirements_management_EN_2009.pdf)

## ▶ Tools: [http://www.jiludwig.com/Requirements\\_Management\\_Tools.html](http://www.jiludwig.com/Requirements_Management_Tools.html)

## ▶ Free community-licensed tool Axiom (Windows, Linux): <http://www.iconcur-software.com/>

- [http://d60f31wukcdjk.cloudfront.net/docs/Axiom\\_4\\_User\\_Manual.pdf](http://d60f31wukcdjk.cloudfront.net/docs/Axiom_4_User_Manual.pdf)

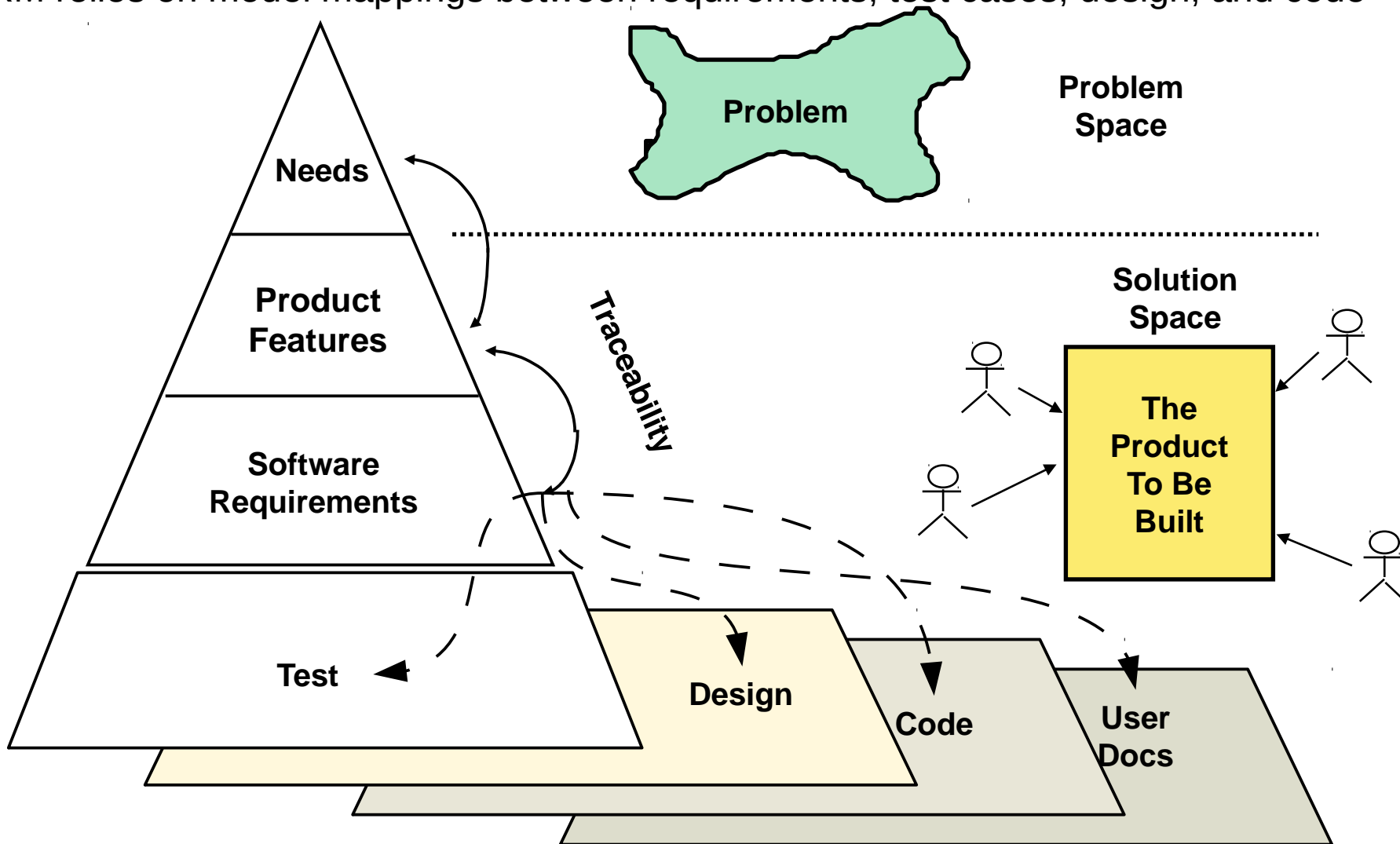
## ▶ Teach videos of Axiom

- <http://www.iconcur-software.com/resources.html>
- Video on linking matrix (traceability matrix) <http://iconcur-software.com/tutorials/matrix.htm>

# Introduction to Requirements Management (RM)

4

- ▶ RM bridges the needs of the customer to testing, design, coding, and documentation
- ▶ RM relies on model mappings between requirements, test cases, design, and code



# Tools in an Integrated Development Environment (IDE)

5

Requirements Tool

Coding Tool

Testing Tool

Model mappings

Model slicing

Model composition

Reachability analysis (traceability)

Attribute analysis

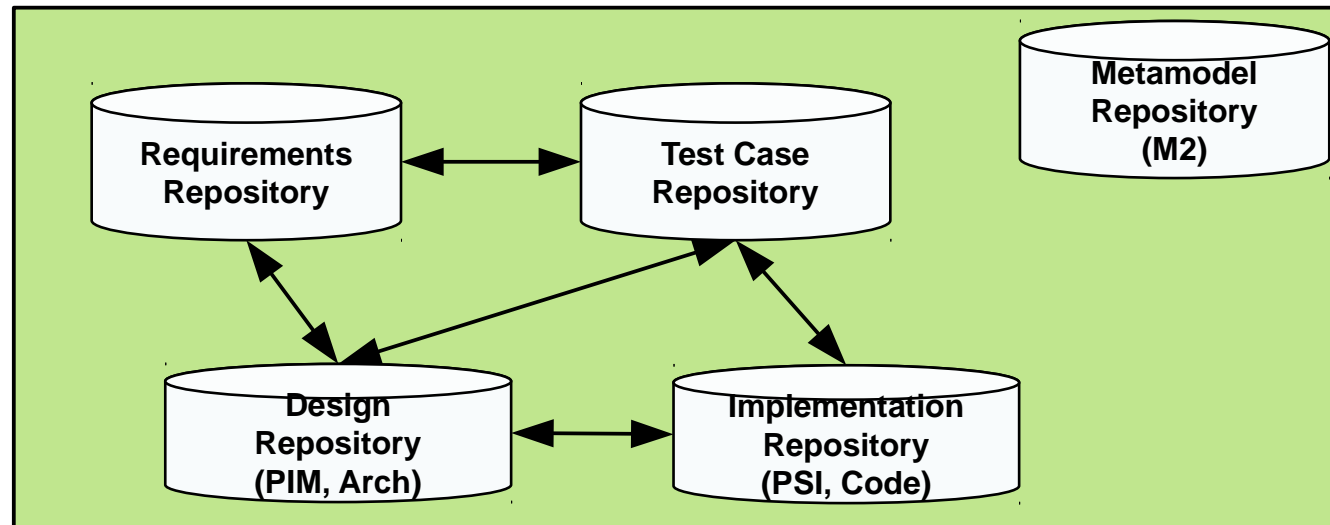
Reasoning engine

Relational engine

GRS engine

TRS engine

XML engine



# Deficiencies of Current RE Methods

6

- ▶ Relationships among requirements are inadequately captured
  - Causal relationship between consistency, completeness and correctness [Zowghi2002]
  - Completeness and consistency are not verified
- ▶ Requirement problems (e.g. conflicts, incompleteness) are detected too late or not all
- ▶ Relationships between requirements and dependent artifacts are insufficiently managed (test, documentation, design, code)
- ▶ Desirable:
  - Models for RE need richer and higher-level abstractions (goals, problems, needs) to validate that they are fulfilled [Mylopoulos1999]
  - **Model mappings (direct and indirect)** between the artifacts (design, code) and the goals, problems, needs of the customer
  - Requirements are consistently managed with design, code, and documentation

# 70.1 Tool-Based Requirements Management

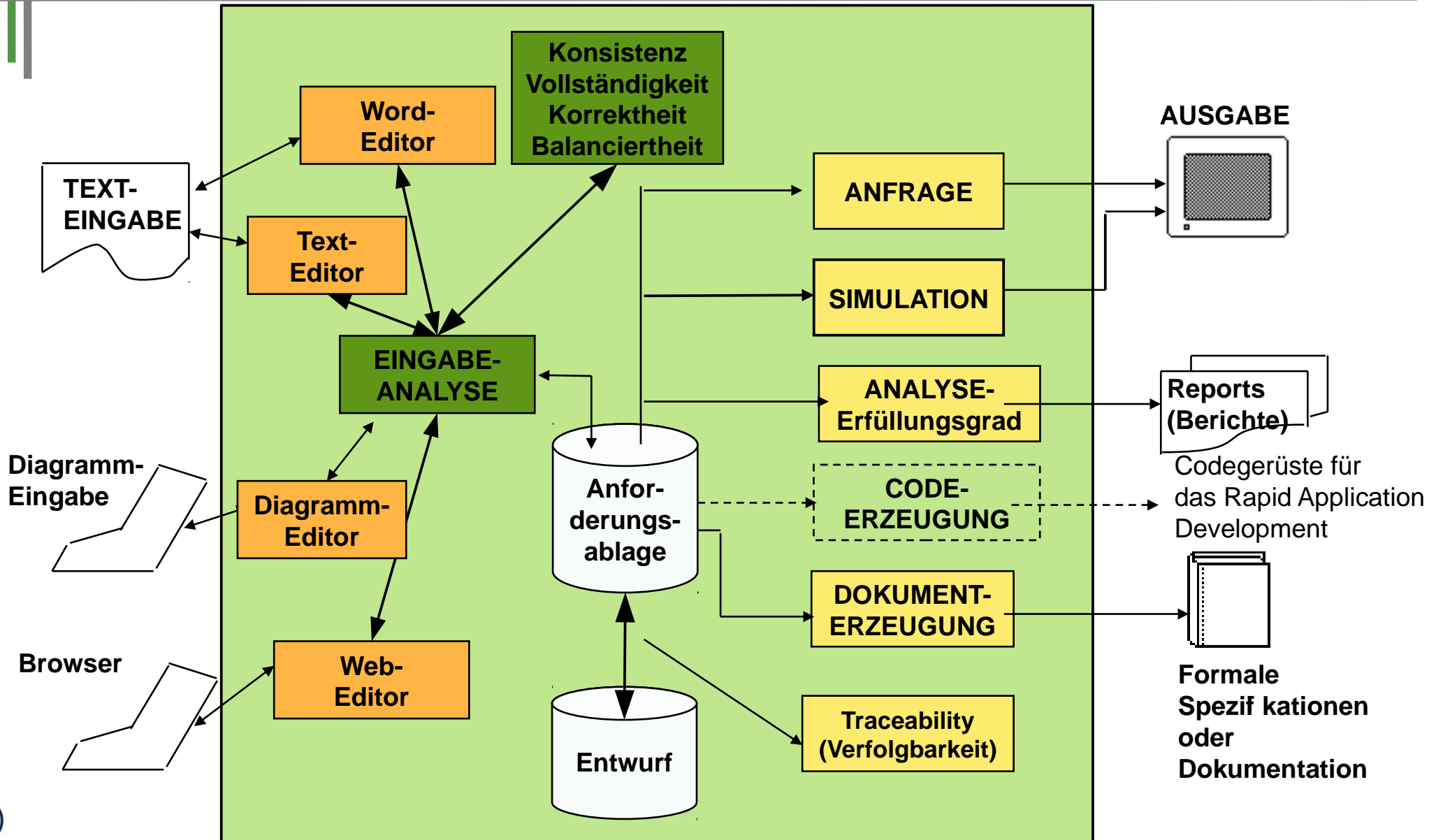


7



# Requirements Tools on the Requirement Database

8





# Metamodeling of Requirements

9

- ▶ Generell empfiehlt sich die Metamodellierung von Requirements, da
  - sich viele Domänen und Anwendungsbereiche unterscheiden
  - die Granularität der Requirements unterschiedlich ist (Balancing)
  - die Requirements dann als Modellelemente existieren und über Model mappings in den Entwurf, die Architektur und den Code verfolgt werden können (**traceability, Verfolgbarkeit**)
  - Verfolgbarkeit wird über Modellabbildungen (model mappings) hergestellt
- ▶ Many requirement tools are metamodel-controlled
  - die die Requirements typisieren
  - und Requirements verlinken

# 70.2 Requisite Pro



10

- ▶ Aufbau einer **metamodellgesteuerten Anforderungsdatenbank**:
  - Formulargesteuerte Erstellung eines Metamodells für die Anforderungen (**requirement types**) (Metasprache ERD)
    - Spezifikation von Anforderungsattributen, wie Status, Priorität, Schwierigkeit, Stabilität, Kosten
    - Anforderungsabhängigkeiten und -verknüpfungen
    - Möglichkeit der hierarchischen Verfeinerung, sowie unterschiedlicher Sichten auf Anforderungsverknüpfungen
  - Anfragen und Querying möglich
  - Konfigurationsmanagement/Änderungsverwaltung: Revisionsstände, Abhängigkeiten, Historie
  - Unterstützung gruppenorientierten Arbeitens
  - Integration in Vorgehensmodelle und SEU, z. B. Rational Unified Process mit Rational Rose, ClearCase sowie MS Project.
- ▶ **Verfolgbarkeit**: in einer “Traceability Matrix” können Requirements mit den Testfälle verknüpft werden
- ▶ Erstellen von **Anforderungsdokumenten mit Word-Vorlagen**:
  - Dokumente strukturiert nach (Standard-)Vorlagen (templates)
  - Unterschiedliche Typen von Anforderungen werden unterstützt (z.B. Produkt-, Software-, Test- und Anwendungsfall-Anforderungen).

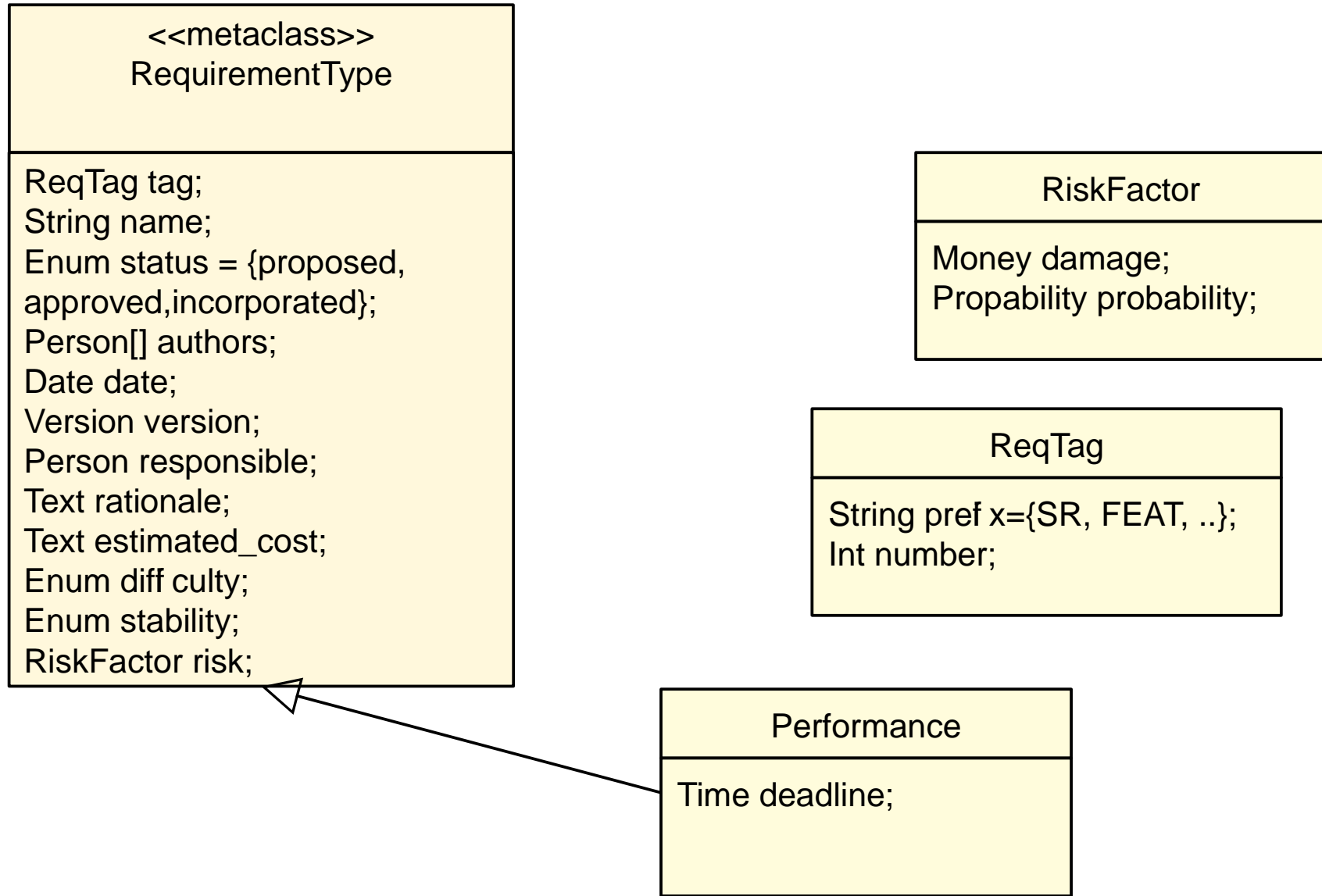
<http://www-142.ibm.com/software/products/de/de/reqpro/>

[ftp://ftp.software.ibm.com/software/rational/docs/v2003/win\\_solutions/rational\\_requisitepro/reqpro\\_user.pdf](ftp://ftp.software.ibm.com/software/rational/docs/v2003/win_solutions/rational_requisitepro/reqpro_user.pdf)

<http://public.dhe.ibm.com/common/ssi/ecm/en/rad10955usen/RAD10955USEN.PDF>

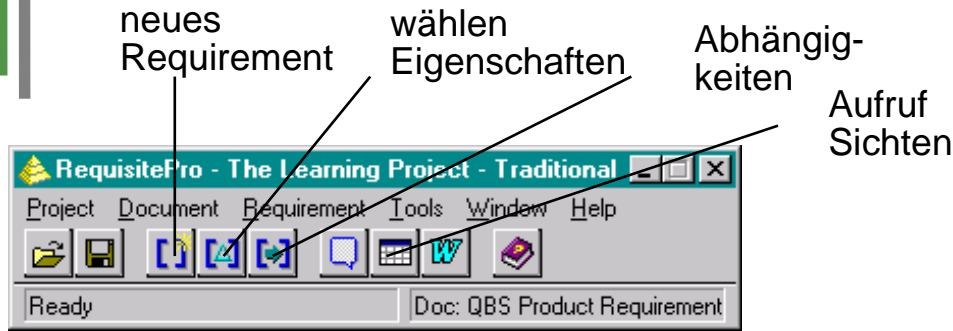
# Metaclass RequirementType (Ex.)

12



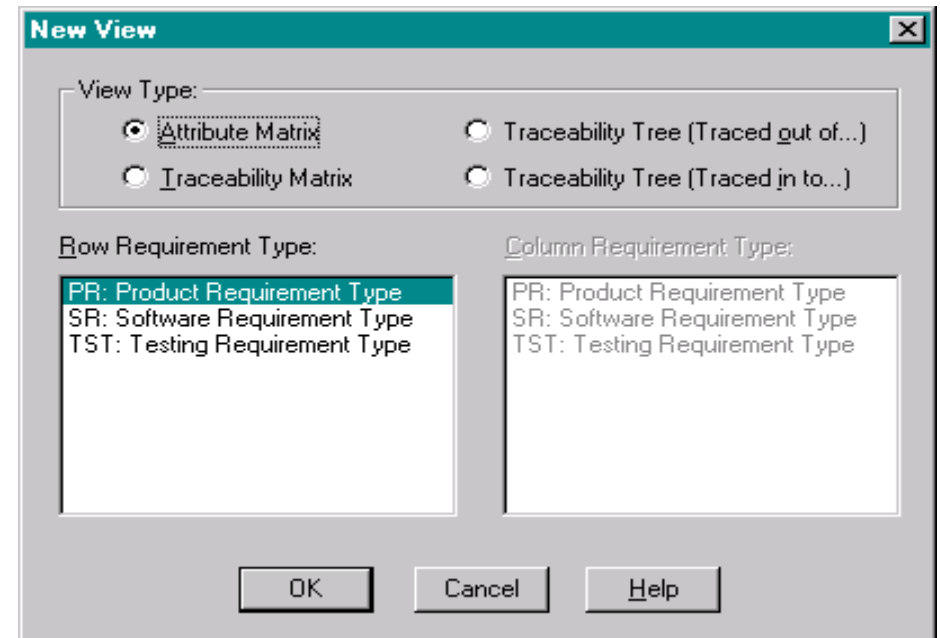
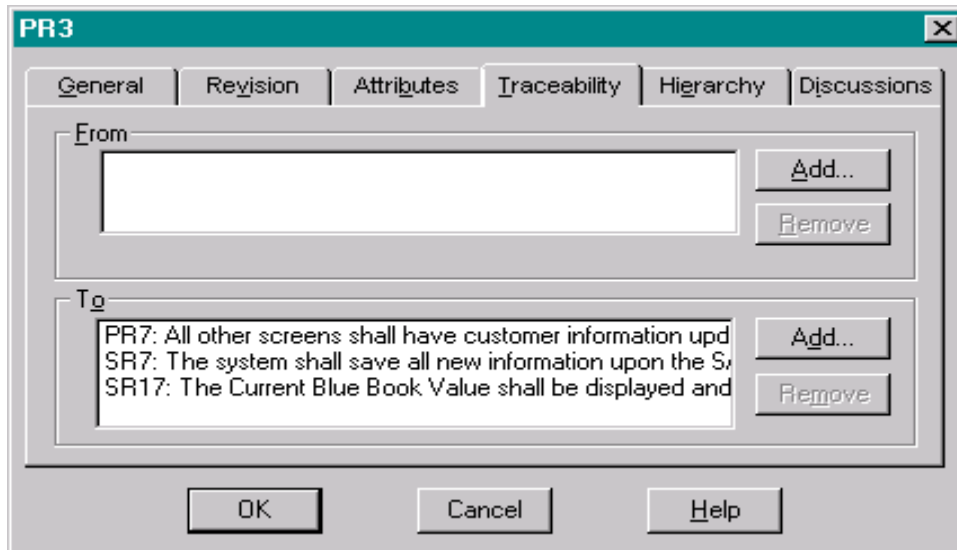
# RequisitePro - Hauptansichten

13



Auswahl unterschiedlicher Sichten und Requirementstypen

## Beschreibung des Requirements PR3



# FURPS Classification of Requirements

14

FURPS delivers RequirementTypes for RequisitePro  
[Wikipedia] [Grady/Caswell] in Hewlett-Packard

- ▶ **Functionality** - Feature set, Capabilities, Generality, Security
- ▶ **Usability** - Human factors, Aesthetics, Consistency, Documentation
- ▶ **Reliability** - Frequency/severity of failure, Recoverability, Predictability, Accuracy, Mean time to failure
- ▶ **Performance** - Speed, Efficiency, Resource consumption, Throughput, Response time
- ▶ **Supportability** - Testability, Extensibility, Adaptability, Maintainability, Compatibility, Configurability, Serviceability, Installability, Localizability, Portability

# Attribute Matrix of Requisite Pro

15

- ▶ The attribute matrix is a hierarchical table (relation) of requirement objects and their attributes
  - Super and subrequirements
  - Priority and Status, and other attributes

# Formalizing Requirement Texts

16

- ▶ If requirements are entered in Word text, they can be **formalized by text mining** with
  - Verb-noun-analysis
  - Keyword identification: MUST, MAY, SHALL, SHOULD, WILL, CUSTOMER
  - Markup information, such as section headers, emphasizing, etc.
  - Concept recognition by looking up nouns in domain models (glossaries, taxonomies, ontologies)
- ▶ Requirements can also be recognized from Word tables [RPro]



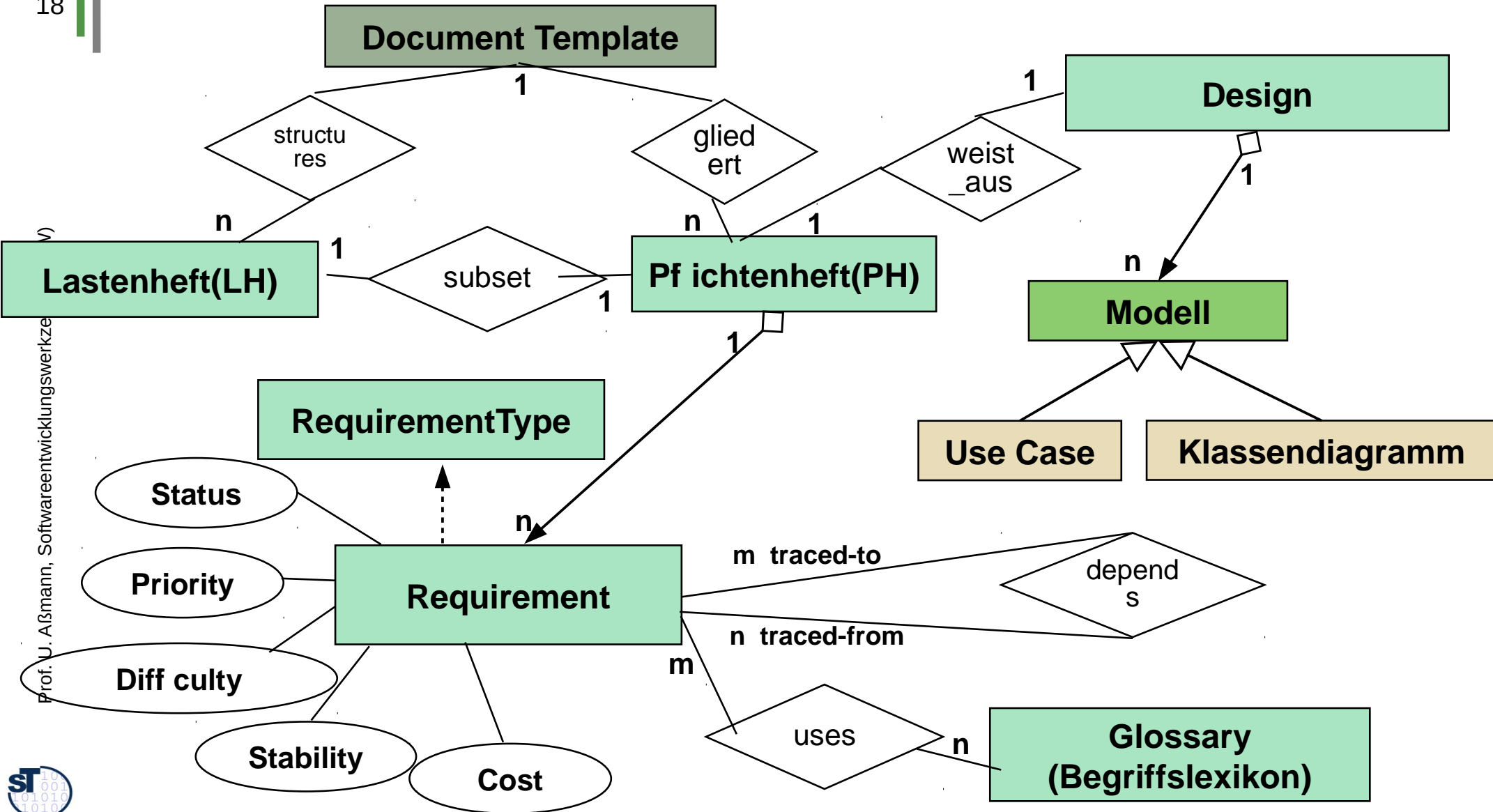
# Traceability with Model Mappings

17

- ▶ The Traceability Matrix connects and relates requirements by **direct traces** and **indirect traces** over **trace\_to** and **trace\_from** relationships
  - The trace relationship is a model mapping within the requirements model
  - External projects can be imported, and traces to their public requirements can be defined
- ▶ Direct traces are entered
  - into a form
  - into the corresponding bitfield of the traceability matrix
- ▶ If somebody changes the requirements later, the trace links become **suspect** and should be checked

# Begriffe des Requirements Managements in RequisitePro

18



# Tools

19

CaliberRM	Borland	<a href="http://www.borland.com/us/products/caliber/index.aspx">http://www.borland.com/us/products/caliber/index.aspx</a>
DOORS	IBM	<a href="http://www-01.ibm.com/software/awdtools/doors/">http://www-01.ibm.com/software/awdtools/doors/</a> <a href="http://www.docstoc.com/docs/90794258/Getting-the-most-out-of-DOORS-for-requirements---NJIT-Computer">http://www.docstoc.com/docs/90794258/Getting-the-most-out-of-DOORS-for-requirements---NJIT-Computer</a>
Siehe auch Test Tools		

# 70.3 Ontology-Driven Requirements Engineering (ODRE)

20

Uwe Aßmann<sup>1</sup>, Katja Siegemund<sup>1</sup>, Edward J. Thomas<sup>2</sup>,  
Jeff Pan<sup>2</sup>, Yuting Zhao<sup>2</sup>

<sup>1</sup> Technische Universität Dresden, Germany

<sup>2</sup> University of Aberdeen, UK

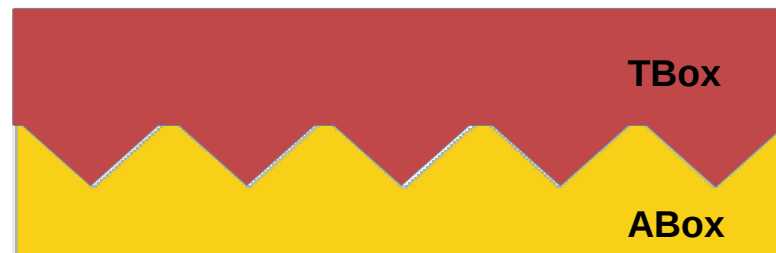
SWESE Oct 24, 2011



# Why Ontology-Driven Requirements Engineering (ODRE)?

21

- ▶ Use graph-logic isomorphism to store requirements and their requirement types in logic, more precisely, in an OWL ontology
  - Provide a metamodel (T-Box of requirements ontology) with a huge set of relevant metadata and requirement relationships
- ▶ Use reasoning services to
  - provide meaningful checks for completeness and consistency, e.g., as queries to the A-Box with SparQL
  - Make specific suggestions to repair inconsistencies and incompleteness
- ▶ Ontology consists of T- and A-Box
  - TBox (Terminological Box) provides metadata
  - ABox (Axiom Box, Fact Base) provides requirements, goals, relationships,...



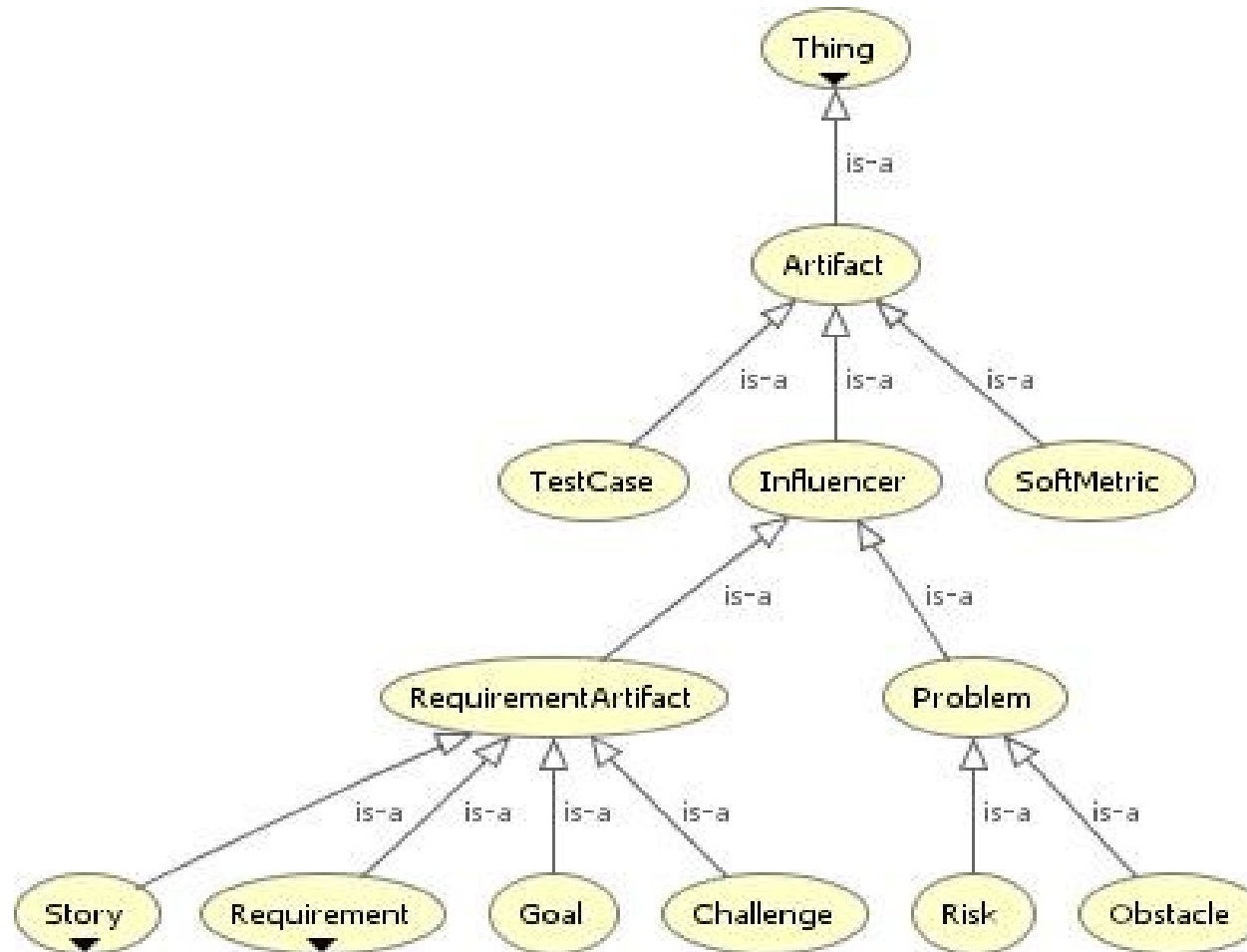
# ODRE Needs Goal-Oriented RE (GORE)

22

- ▶ Lamsweerde defines **goals** as "declarative statements of intent to be achieved by the system under consideration" [Lamsweerde2000]
- ▶ Benefits of explicit specification of goals in GORE:
  - Goals drive the identification of requirements
  - Goals provide a criterion for sufficient completeness of a requirement specification
    - Specification of pertinent requirements
    - Relationships between goals and requirements can help to choose the best one
  - Concrete requirements may change over time whereas goals pertain stable

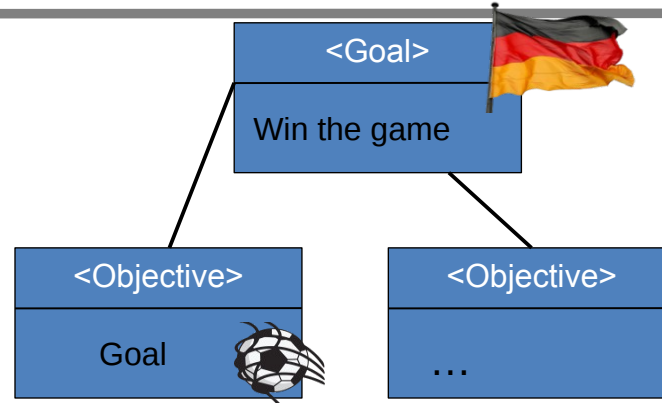
# Goal-Oriented Requirements Engineering (GORE) – TBox of GORE Ontology

23



# Goal-Oriented RE (Motivation Example)

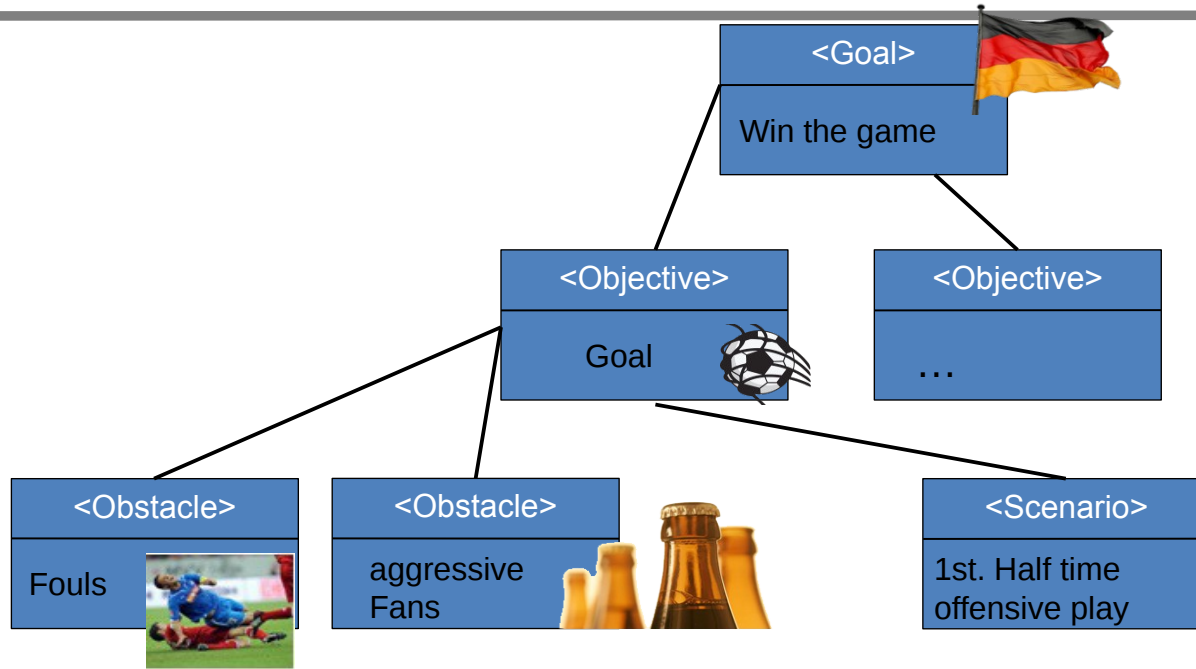
24



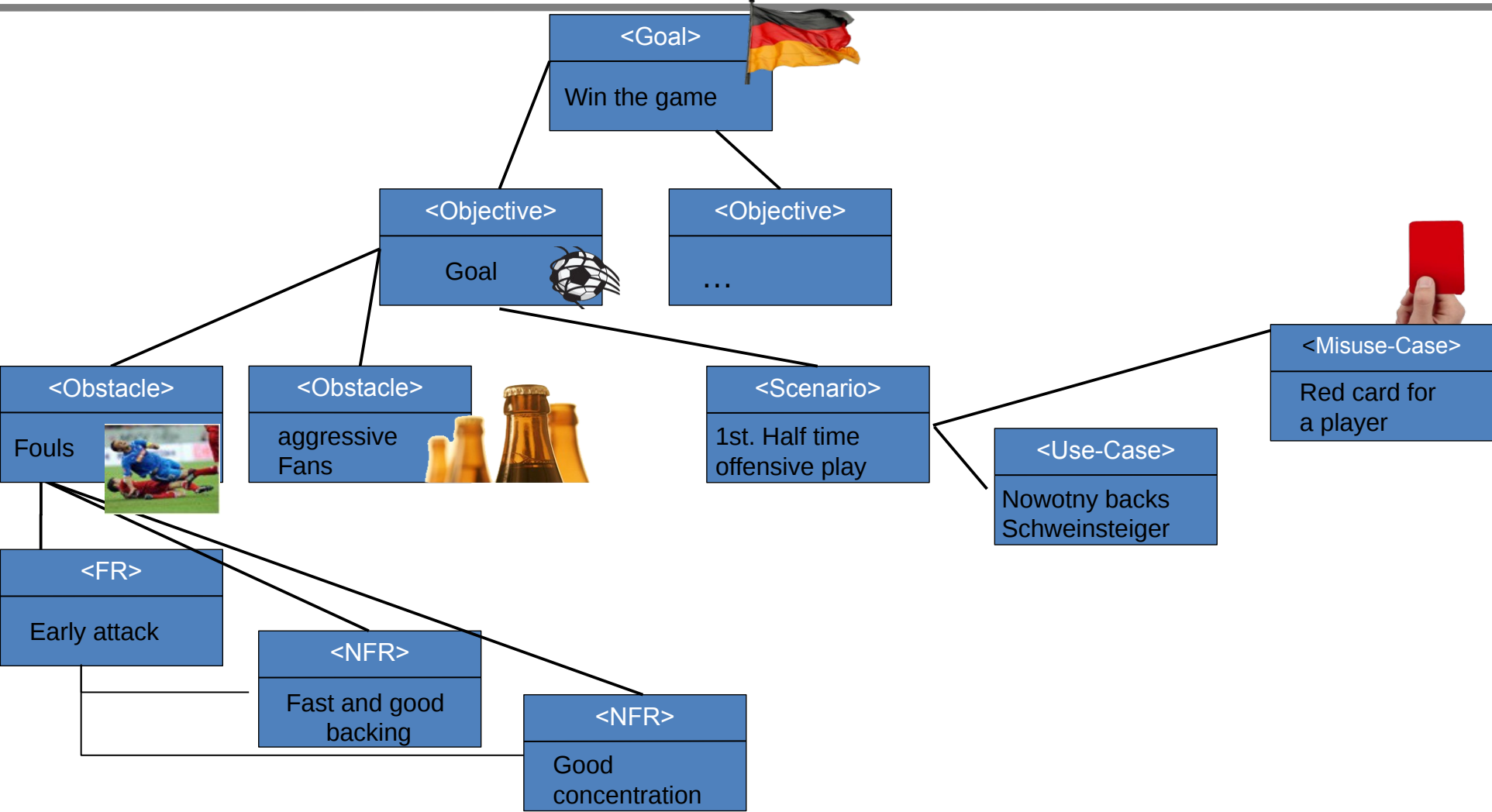


# Goal-Oriented RE (Motivation Example)

25

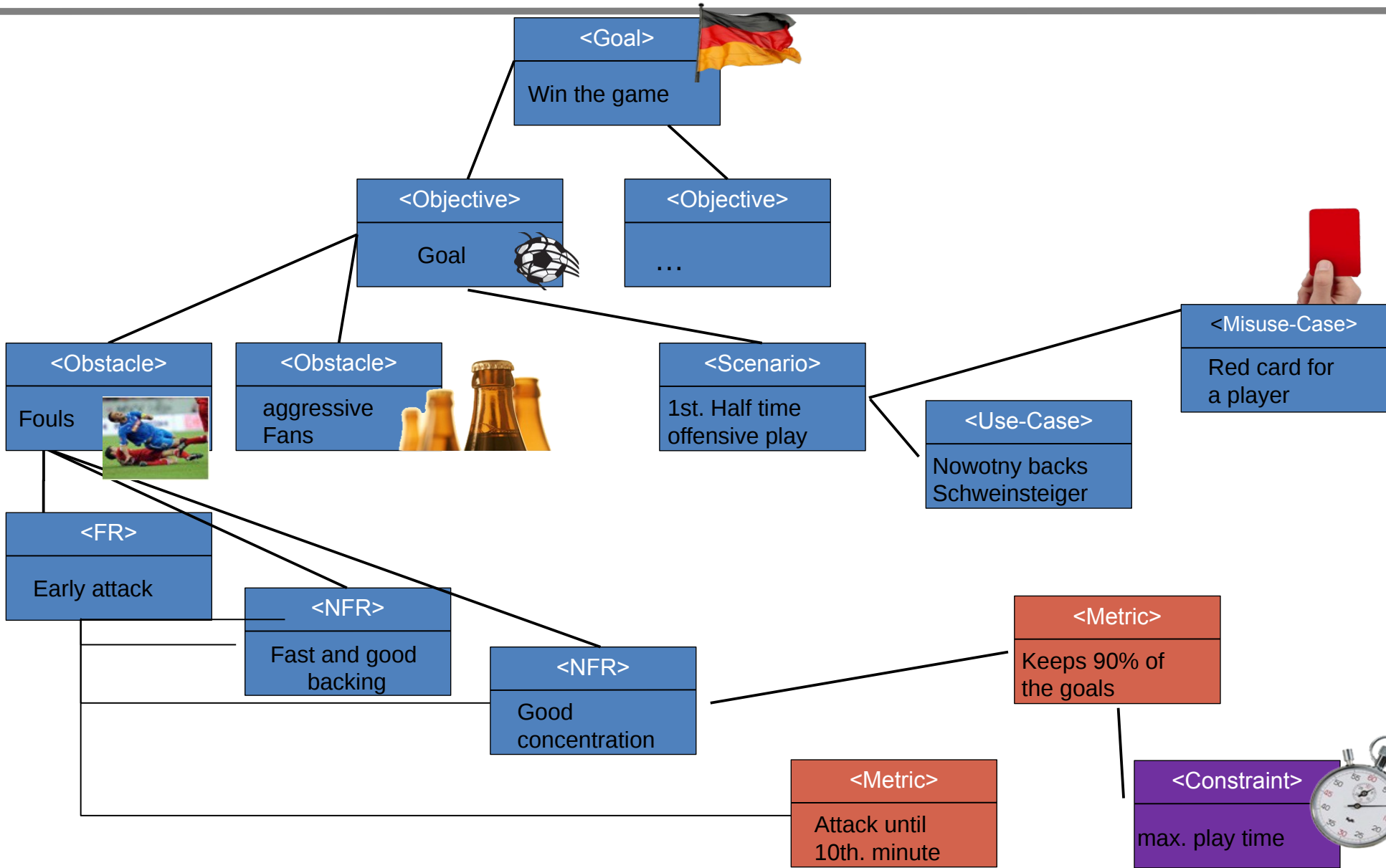


# Goal-Oriented RE (Motivation Example)



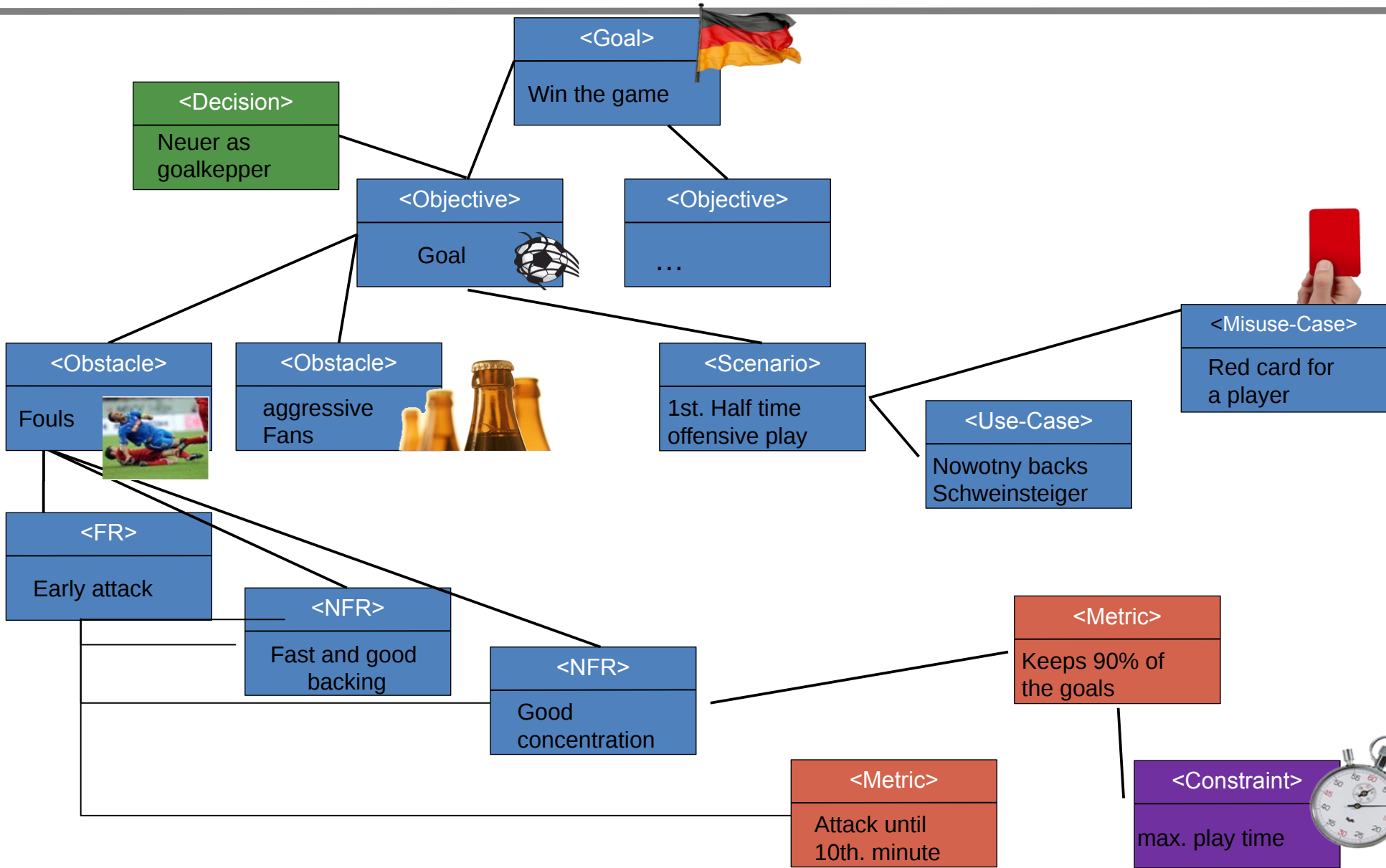
# Goal-Oriented RE (Motivation Example)

27



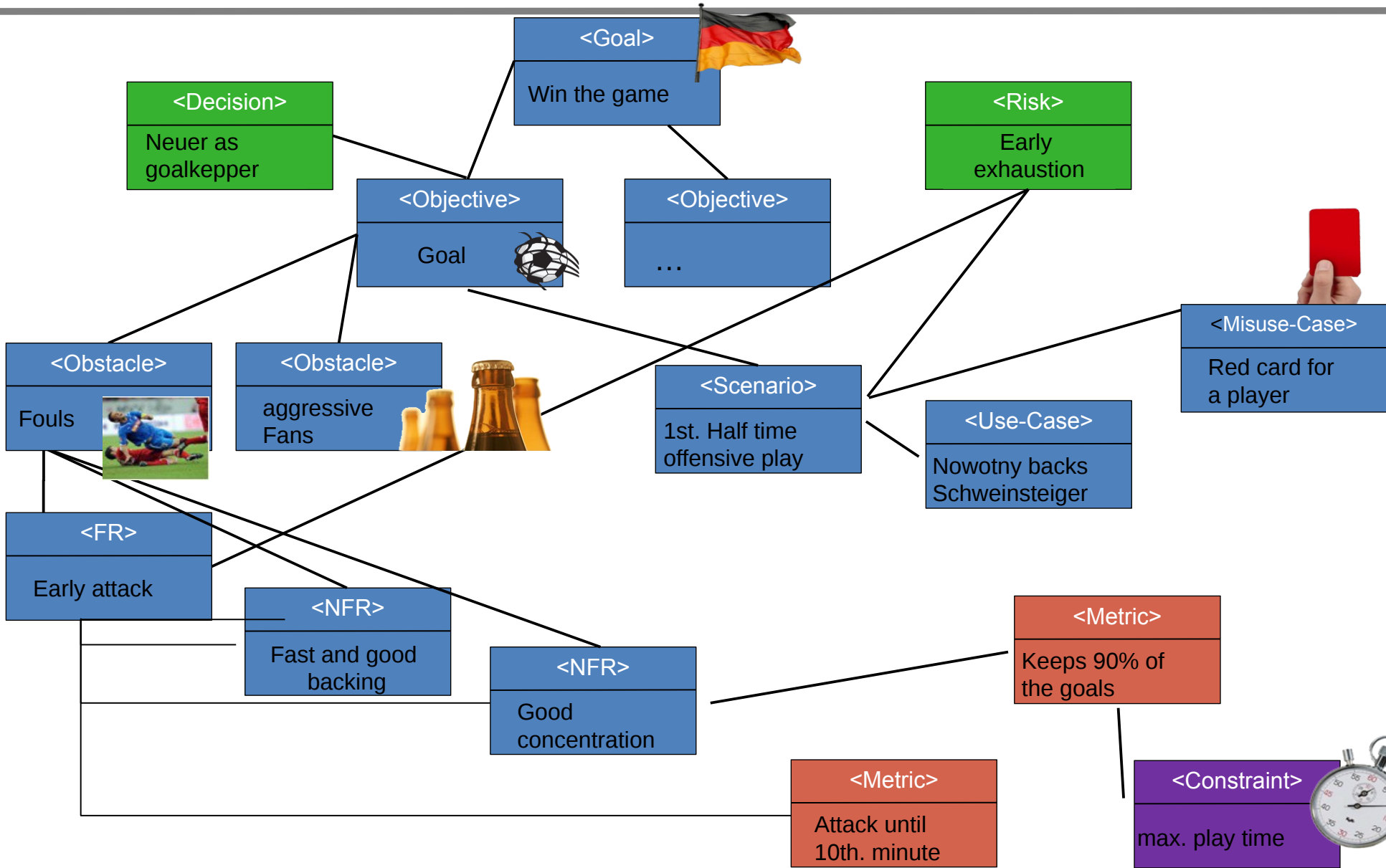
# Goal-Oriented RE (Motivation Example)

28



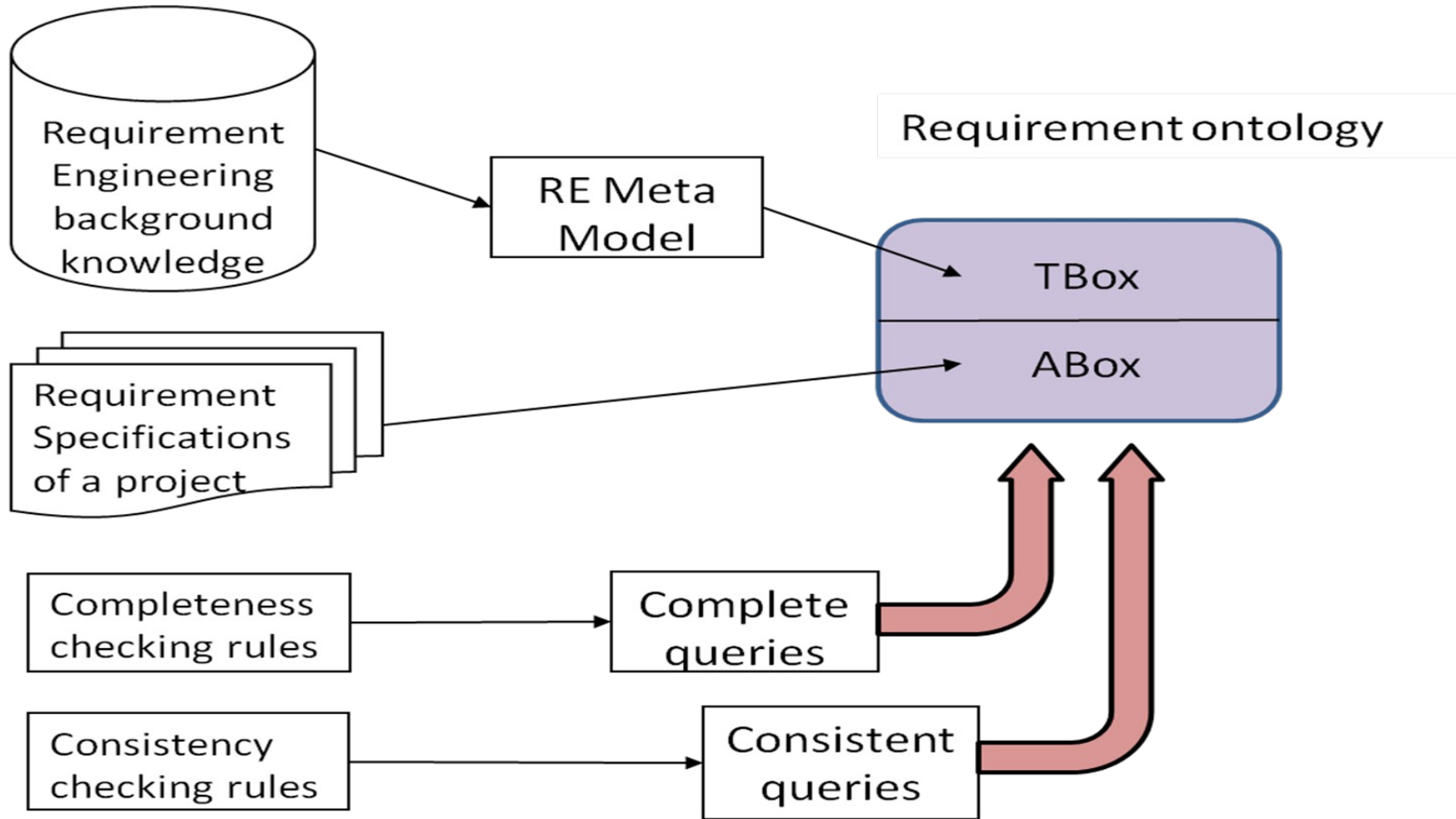
# Goal-Oriented RE (Motivation Example)

29



# Architecture for ODRE Tool

30



# Reasoning for RE – Completeness Check

31

- ▶ Example of Completeness Rule:

“Every Functional Requirement (FR) must define whether it is mandatory or optional.”

- ▶ The GORE ontology of Lambsweerde needs 46 completeness rules
  - Implemented as SPARQL queries on the A-Box
  - The requirements model is deemed incomplete if a specific rule fails
  - Reasoning Strategy: Closed World Reasoning (for negation as failure)
    - supported by SPARQL 1.1 and TrOWL reasoner

## Reasoning for RE – Completeness Check (Example)

32

“Every Functional Requirement (FR) must define whether it is mandatory or optional.”

► SPARQL rule:

```
IF FR is NOT mandatory AND NOT optional THEN
  Print error: "You did not specify whether
    the following FRs are mandatory or optional:
    [FR_n].“
  "Please specify whether these FRs are mandatory
  or optional."
```



# Reasoning for RE – Completeness Check (Example)

33

- ▶ Extract of individuals and relationships of the A-Box from the SPARQL analysis :

*isRelatedTo(Goal2;UseCase7)*

*NonFunctionalRequirement (NonFunctionalRequirement1)*

*IsOptional(NonFunctionalRequirement1; true)*

*FunctionalRequirement(FunctionalRequirement1)*

## **Error.**

You did not specify whether the following FR are mandatory or optional:

[FunctionalRequirement1](#). Please specify this attribute for the FR:

[FunctionalRequirement1](#). Every FR must specify AT LEAST ONE requirement relationship.

# Reasoning for RE – Consistency Check

34

- ▶ GORE needs 6 consistency rules among requirement artefacts (valid relations between requirement artefacts)
  - Based on a chosen subset of requirement artefacts
  - Consistency rules are encoded as DL axioms in the A-Box
- ▶ Instance specific error messages resulting from validation displayed by Guidance Engine

# Reasoning for RE – Consistency Check (Example)

35

- ▶ Extract of individuals and relationships of the A-Box from the SPARQL analysis :

*isExclusionOf (FunctionalRequirement5; FunctionalRequirement7)*

*ChosenRequirement (FunctionalRequirement5)*

*ChosenRequirement (FunctionalRequirement7)*

## Error.

The following requirements exclude others:

[FunctionalRequirement5](#).

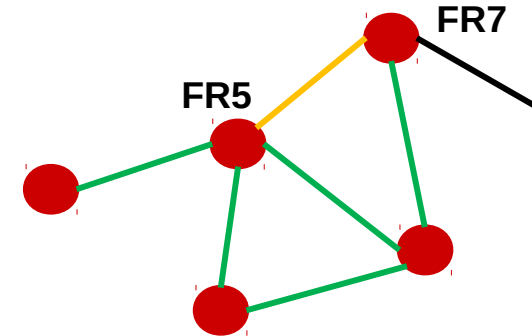
Please choose one of the following options:

## Suggestion.

Exclude the following requirements from the chosen requirement set: [FunctionalRequirement5](#). **OR**

Find alternatives for: [FunctionalRequirement5](#) or

Revise the requirement relationships of ([FunctionalRequirement5](#), [FunctionalRequirement7](#)).



# Reasoning for RE – Verification Methods (Example)

36

## ► Consistency check of requirement selection (6 rules)

Excluding requirements must not be included in one set.

```
IF excluding requirements are included in one set
THEN print error: "The following requirements exclude
    others: [R_n]."  
"Please choose one of the following options:  
Exclude the following requirements: [R_n],  
Find alternatives for [R_n] or  
Revise the requirement relationships of [[R x, R y], ... ]."
```

- ▶ All Requirement artefacts and meaningful relationships can be captured within an Ontology Metamodel
- ▶ ODRE Approach detects **inconsistent** and **incomplete** requirements
- ▶ Standard tooling (reasoners) are useful
  - Specification of requirements uses OWA
  - Verification needs CWA
- ▶ First evaluation proves applicability for medium requirement specifications
  - Problem: available requirement specifications do not provide sufficient information (much less than could be captured by ODRE)
  - Primary evaluation within MOST Project
    - Capture all requirement artefacts
    - Detect all inconsistencies and incomplete metadata
  - Main evaluation planned for PhD Thesis of Katja Siegemund (2012)

# 70.4 Traceability to other Artefacts



38

# Direct Traceability

39

- ▶ With a **direct model mapping**, a requirements model can be linked
  - to a test case specification
  - to a documentation
  - to an architectural specification
  - via the architectural specification, to the classes and procedures in the code

# Example: imbus TestBench

40





# Requirements get “red-yellow-green” Test Status Attribute

41

The screenshot shows a software application window titled "Anforderungsverwaltung von Car Konfigurator (Version 2.1, Abnahmetest)". The interface is divided into two main sections:

- Anforderungsbaum (Left):** A tree view showing the hierarchy of requirements. The root is "CarKonfigurator - Version 1.1 (caliber)". It is divided into four main categories:
  - 1. Business Requirements:** Includes "Konfiguration zusammenstellen" (yellow), "Rabatt gewähren" (green), "automatische Rabatte" (green), and "Händler gewährt Rabatt" (green, highlighted).
  - 2. User Requirements:** Includes "ständige Preisanzeige" (green) and "keine erzwungene Bedienerfolge" (yellow).
  - 3. Functional Requirements:** Includes "sofortige Preisberechnung" (red), "Quelle der Basisdaten" (green), "Import einer Datei" (green), and "Import vom OEM-Host" (green).
  - 4. Design Requirements:** Includes "gültige Konfiguration" (grey) and "Eingabe der Basisdaten" (red).
- Details Panel (Right):** A tabbed interface with tabs for "Details", "Benutzerdefinierte Felder", "Erweitert", "Wird verwendet in", and "Alle Versionen". The "Details" tab is active, showing the following information for the selected requirement "Händler gewährt Rabatt":
  - Name:** Händler gewährt Rabatt
  - ID:** WHY162
  - Version:** 1.1
  - Eigentümer:** (empty field)
  - Status:** Review Complete
  - Priorität:** Essential
  - Test-Status:** ■ Getestet PASS

Testf[...] : endpreis-berechnen-mit-rabatten\_log.xml
Aktuelle Ansicht : Endpreis berechnen mit Rabatten : [...]gurieren : Fahrzeug wählen CBR

**2.3.2 Endpreis berechnen mit Rabatten**

- 1. einfach
  - CarConfig Starten
  - Preis prüfen
  - CarConfig Beenden
- 2. Testfall
  - CarConfig Starten
  - Fahrzeug konfigurieren
  - Fahrzeug wählen CBR**
  - Sondermodell wählen
  - Zubehör wählen
  - Preis prüfen
- Fahrzeug konfigurieren
  - Fahrzeug wählen CBR
  - Sondermodell wählen
  - Zubehör wählen
  - Preis prüfen
- Fahrzeug konfigurieren
  - Fahrzeug wählen CBR
  - Sondermodell wählen
  - Zubehör wählen
  - Preis prüfen
- Endpreis berechnen "ohne" Rabatt
  - CarConfig Starten
  - Fahrzeug konfigurieren
  - Fahrzeug wählen CBR
  - Sondermodell wählen

Interaktion

**Fahrzeug wählen CBR**

Parameter	Wert
Fahrzeug	15

Fehler  Fehler hinzufügen

---

**Interaktion: Fahrzeug wählen CBR** Bemerkungen

-Beschreibung

Fahrzeug aus der Liste der Fahrzeuge wählen

-Bemerkungen zur Durchführung

-Bemerkungen zur Spezifikation

---

**Benutzerdefinierte Felder der Durchführung**

<für diesen Knotentyp können Benutzerdefinierte Felder nicht definiert werden>

---

**Liste der Anforderungen**

Name	ID	Version	Eigentümer	Status	Priorität
sofortige Preisberechnung	WHAT303	3.1	Dierk	Accepted	Essential
keine erzwungene Bedienerfolge	USER302	1.0	Dierk	Submitted	Essential
ständige Preisanzeige	USER301	1.0	Dierk	Submitted	Essential

---

**Aufgezeichnete Attribute**

**Tester**

Aktueller Benutzer

Tester

**Letzte Änderung des Ergebnisses**

Aktuelles Ergebnis  Zu prüfen

Ergebnis-Datum (DD.MM.YYYY)

Ergebnis-Zeit (HH:MM:SS)

**Zeitmessung**

Geplante Durchführungszeit (DD:HH:MM:SS.SSS)

Aktuelle Durchführungszeit (DD:HH:MM:SS.SSS)



**The End**