



SpecIF – Was ist das?

16.11.2022 Tutorial TdSE 2022 | Dr. Oliver Alt | Paderborn

GfSE →
Gesellschaft für
Systems Engineering e.V.
GERMAN CHAPTER OF INCOSE



Block I

9:00 – 10:10 Uhr

- Begrüßung und Vorstellung
- Was ist SpecIF?
- Die SpecIF-Terminologie

60 Minuten +
10 Minuten Pause

Block II

10:10 – 10:50 Uhr

Semantische Integration I

30 Minuten +
10 Minuten Pause

Block III

10:50 – 11:30 Uhr

Semantische Integration II

30 Minuten +
10 Minuten Pause

Block IV

11:30 – 12:00 Uhr

Praktische Anwendung

30 Minuten

Vorstellungsrunde

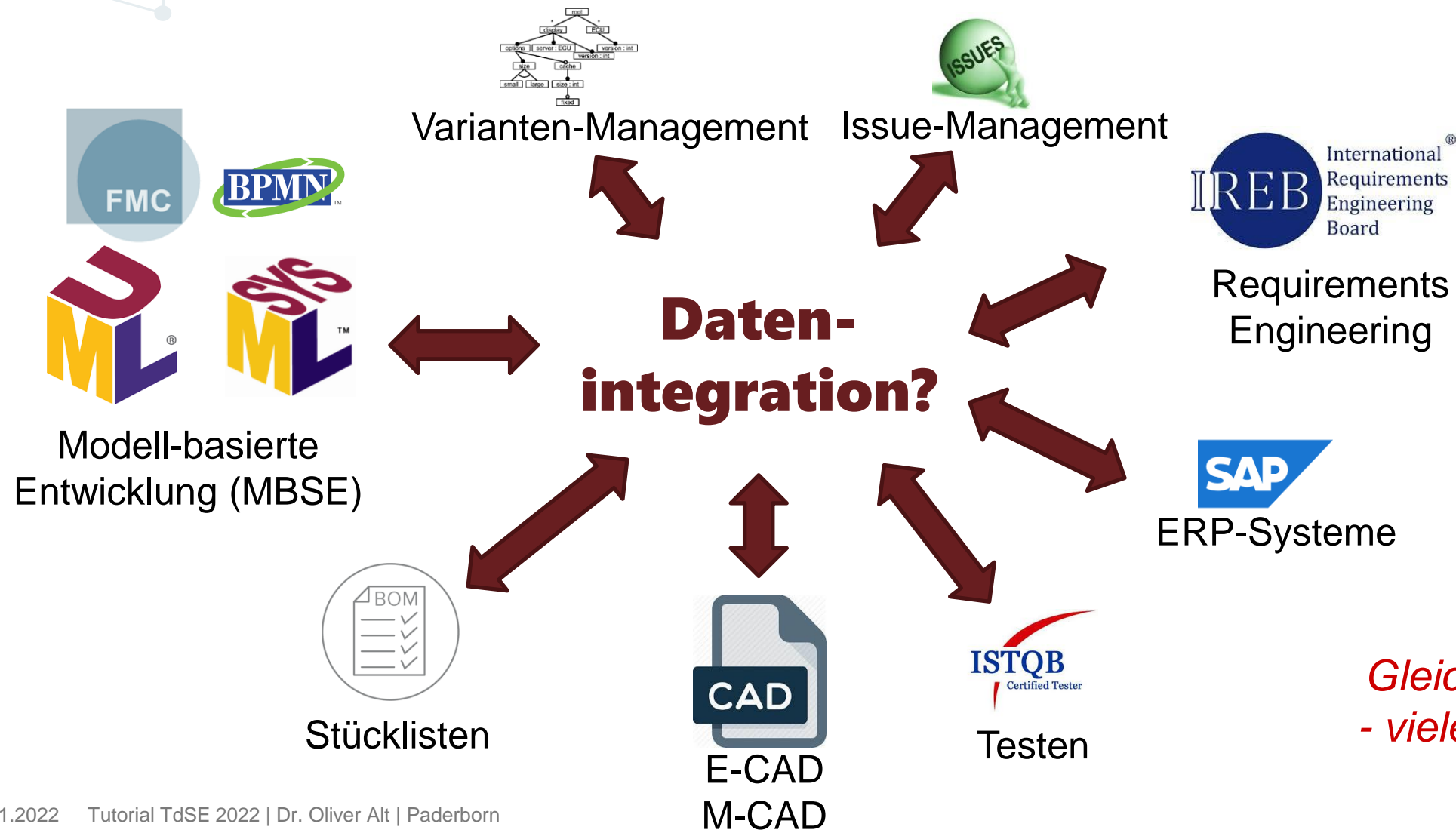


1

Was ist SpecIF?



Systems Engineering im Produktlebenszyklus



*Gleiches Produkt
- viele Werkzeuge*

Grundlegende Annahmen

- Es wird immer spezielle Werkzeuge für bestimmte Aufgaben geben, weil die eierlegende Wollmilchsau ein Fabelwesen ist.
- Dennoch gibt es ein Interesse
 - Teilergebnisse verschiedener Teams zusammen zu führen
 - übergreifend zu navigieren, zu suchen und zu prüfen
 - Modellinformationen zwischen Organisationen und Werkzeugen auszutauschen



→ Hier setzt SpecIF an ...

Was ist SpecIF?



Eine Initiative und Technologie definiert durch Mitglieder der Arbeitsgruppe PLM4MBSE der Gesellschaft für Systems Engineering (GfSE)

- SpecIF ist ein Vokabular für standardisierte Begriffe im MBSE und PLM sowie eine Aussagenlogik „Subjekt Prädikat Objekt“ *... ABER AUCH*
- eine Weiterentwicklung des Requirement Interchange Format (ReqIF), also ein Datenformat *... ABER AUCH*
- ein standardisiertes Web-API *... ABER AUCH*
- eine semantische Integrationsebene

SpecIF 1.1 ist seit einem Jahr freigegeben



- Die erste Freigabe der SpecIF erfolgte am 10.11.2021 mit Version 1.1 zum TdSE 2021
 - <https://specif.de>
 - <https://github.com/GfSE/SpecIF>
- Vorangegangen waren ca. 5 Jahre gemeinsame Entwicklungsarbeit innerhalb der Arbeitsgruppe



Grundsätze bei der Entwicklung von SpecIF



- Nicht das Rad neu erfinden, sondern auf bestehende Standards aufbauen
 - Implementierungen erleichtern
- Nur so komplex wie nötig und so einfach wie möglich
- Open Source Referenzimplementierungen
 - <https://github.com/GfSE>
 - <https://github.com/oalt>

SpecIF hat technische und Domänenaspekte



OpenAPI

Domänen:

- Generic
- Requirement-Mgmt
- Supplier Agreement (Automotive)
- Model-Integration
- Vocabulary
- Test-Management
- Variant-Management

Standardisiertes Vokabular

DublinCore

OMG Standards Development Organization

Domänen: Standardisierte Semantik

FMC

IREB International Requirements Engineering Board

ISTQB Certified Tester

ReqIF Requirements Interchange Format

Technisch:
Technischer Informations-
träger

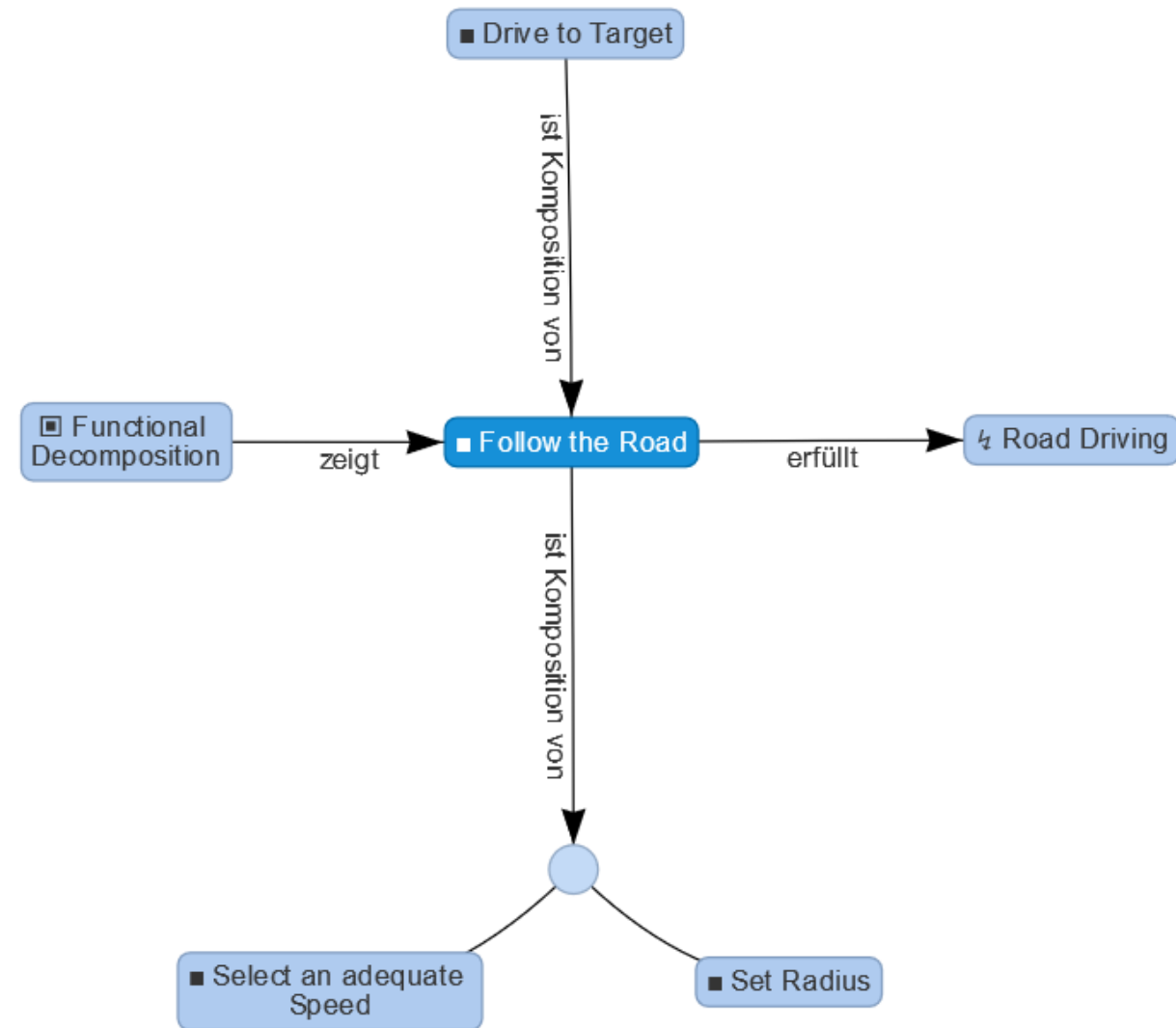
{JSON}
JavaScript Object Notation

Schema and Constraints



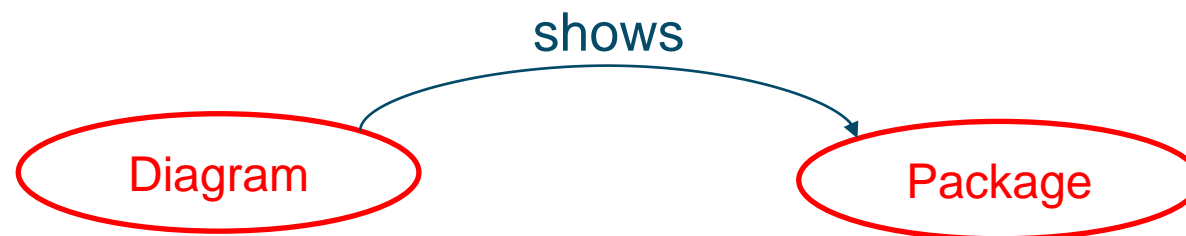
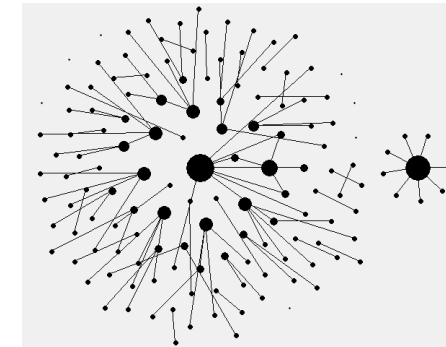
SpecIF – eine Art ReqIF 2.0

- Weiterentwicklung von ReqIF
 - ReqIF- wie SpecIF-Daten sind Graph-Daten
 - JSON mit JSON-Schema
 - Versionierung der Einzelelemente
 - Mehrsprachigkeit
 - Vererbung
 - Weitere genutzte Standards: XHTML, SVG

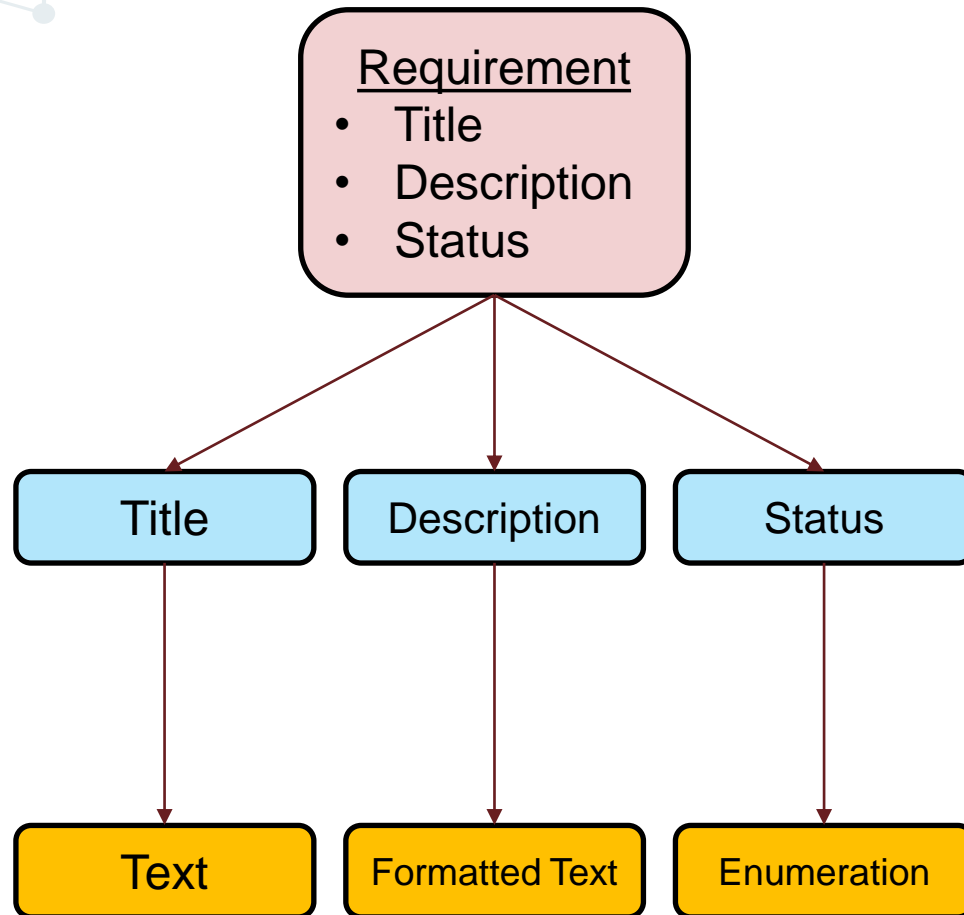


Grundprinzipien der Informationsabbildung in SpecIF

- SpecIF Daten sind Graph-Daten (Knoten und Kanten)
- Die Knoten heißen *Ressourcen*, die Kanten *Statement*
- Zwei *Ressourcen* und ein *Statement* bilden eine prädikatenlogische Aussage



Wie repräsentiere ich Daten mit SpecIF?



Resources

Properties

Data Types

Resource: Requirement

Title:
„Cloud First Approach“

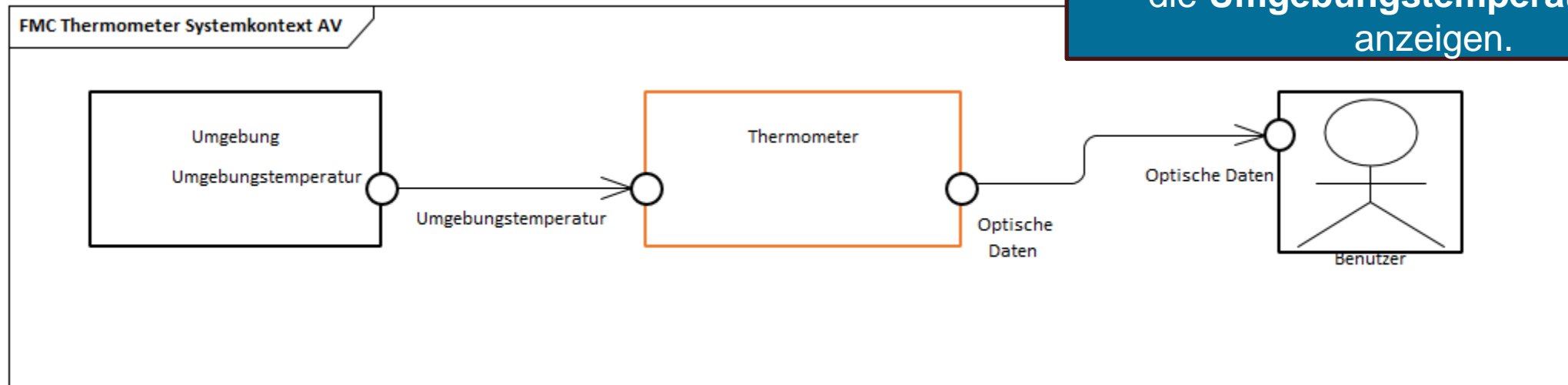
Description:
„All digitalization solutions of
the KM group **shall** follow the
cloud-first approach.“

Status:
„Approved“

Semantische Netze

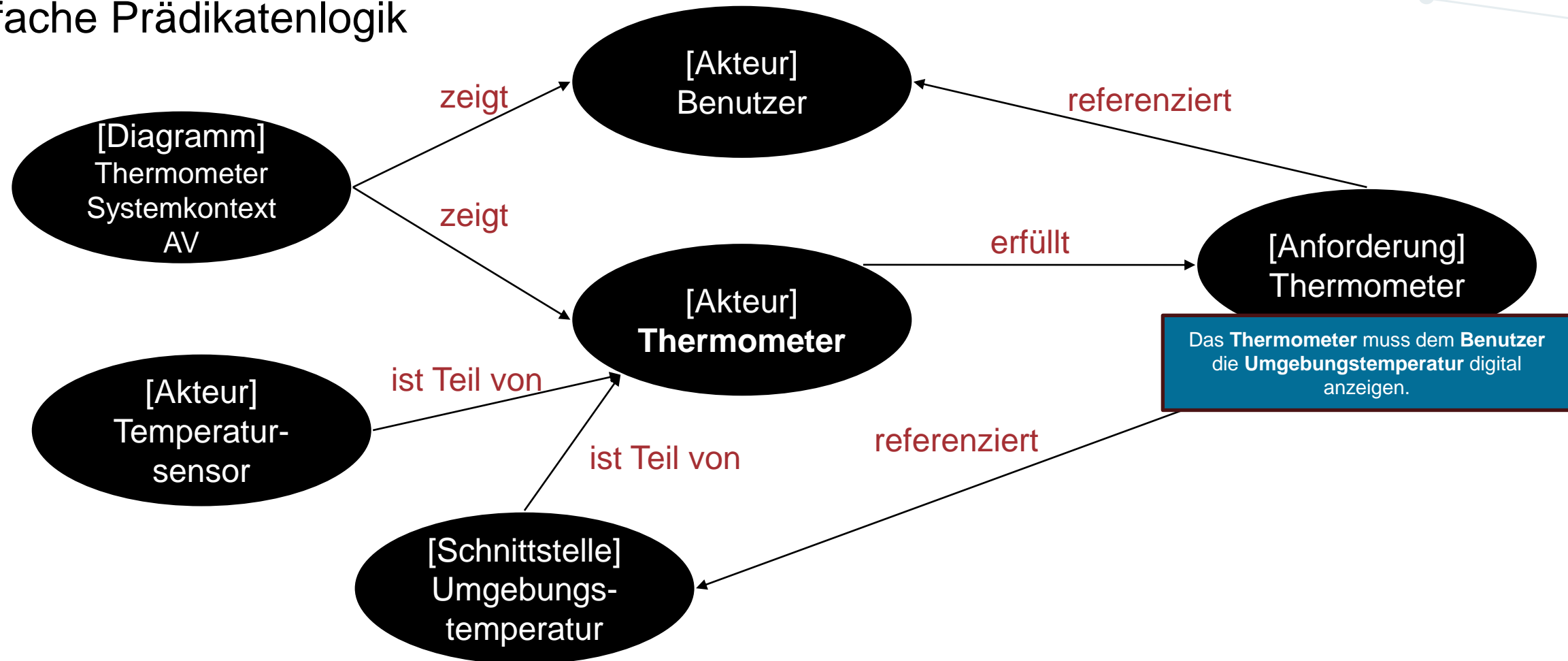
- Durch Ressourcen und Statements lassen sich Graphen, so genannte „semantische Netze“ aufbauen – analog zu Techniken des „semantic Web“ (RDF etc.)
- In solchen Netzen kann man durch die Daten navigieren und ggfs. auch logische Schlüsse daraus ziehen
- Beispiel:

Das Thermometer muss dem Benutzer die Umgebungstemperatur digital anzeigen.



Ein semantisches Netz

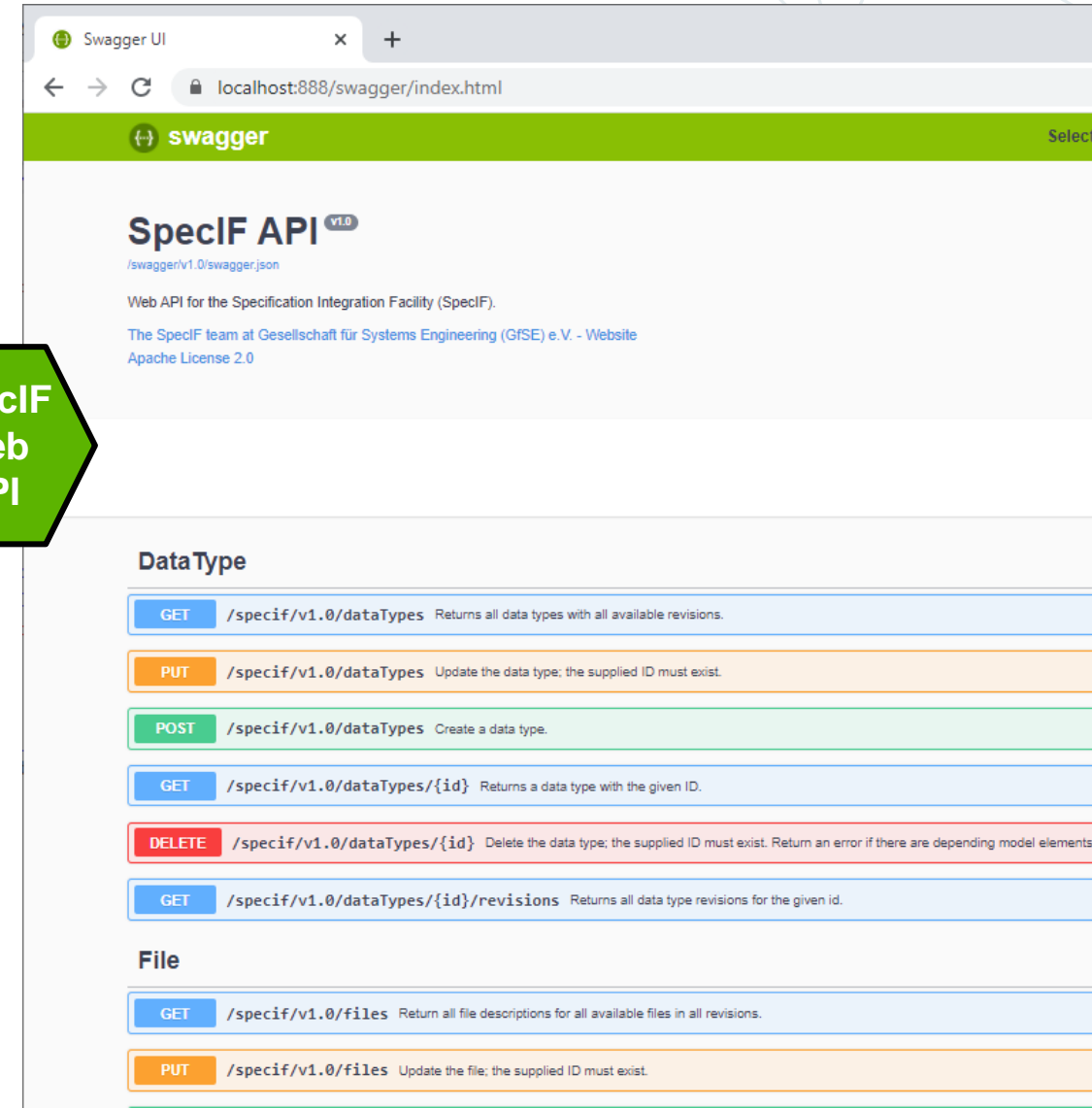
Einfache Prädikatenlogik



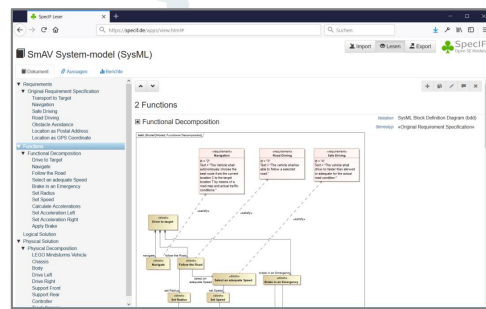
SpecIF – das Web-API

- Swagger bzw. OpenAPI Spezifikation
 - <https://github.com/GfSE/SpecIF-OpenAPI>
- CRUD-Operationen für alle SpecIF-Elemente
- Einheitliche Schnittstelle für Werkzeuge
- Prototypische Umsetzung als Open Source
 - <https://github.com/oalt/SpecIF-Backend>
 - <https://github.com/oalt/specif-mongodb-backend>

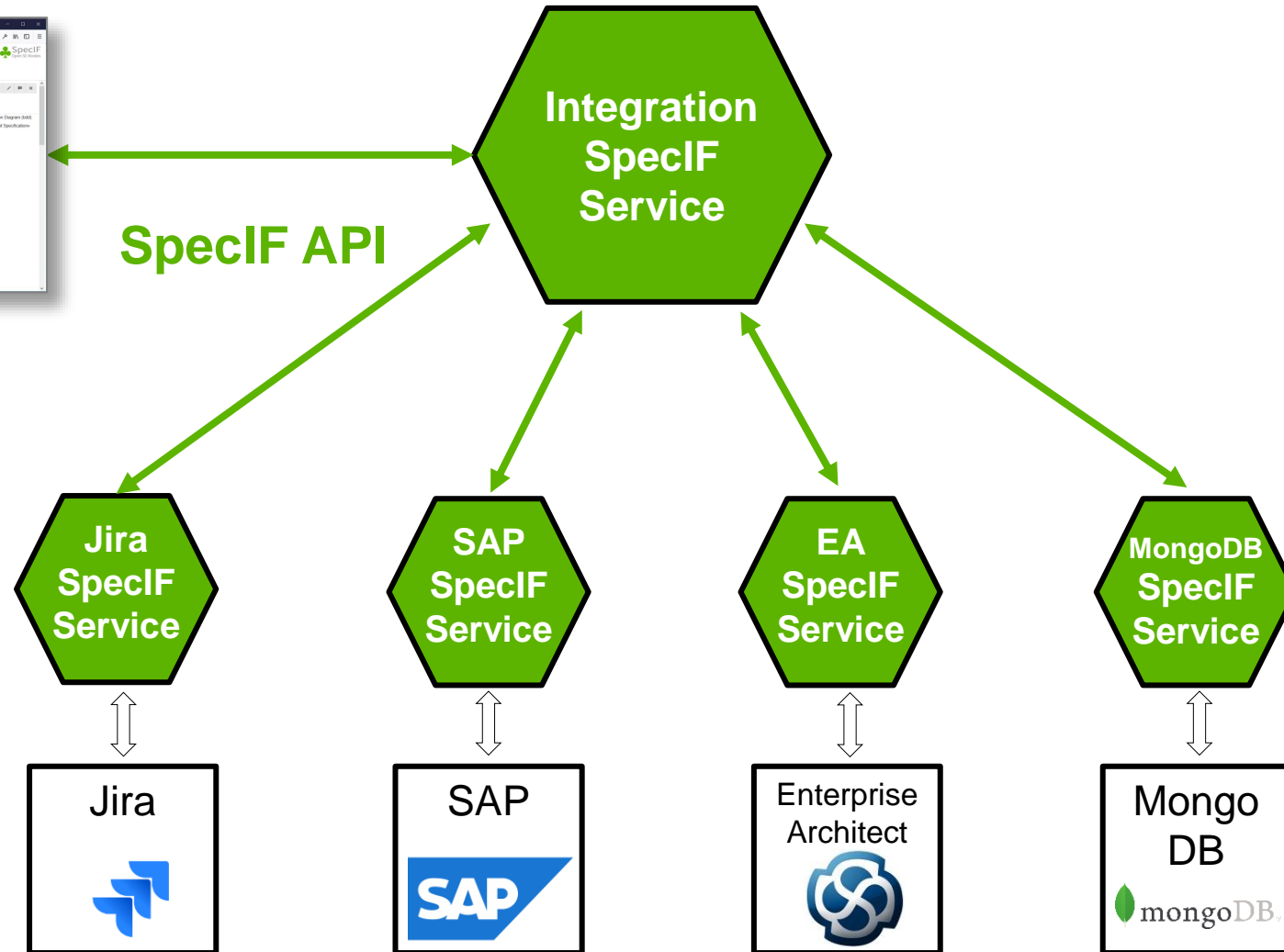
SpecIF
Web
API



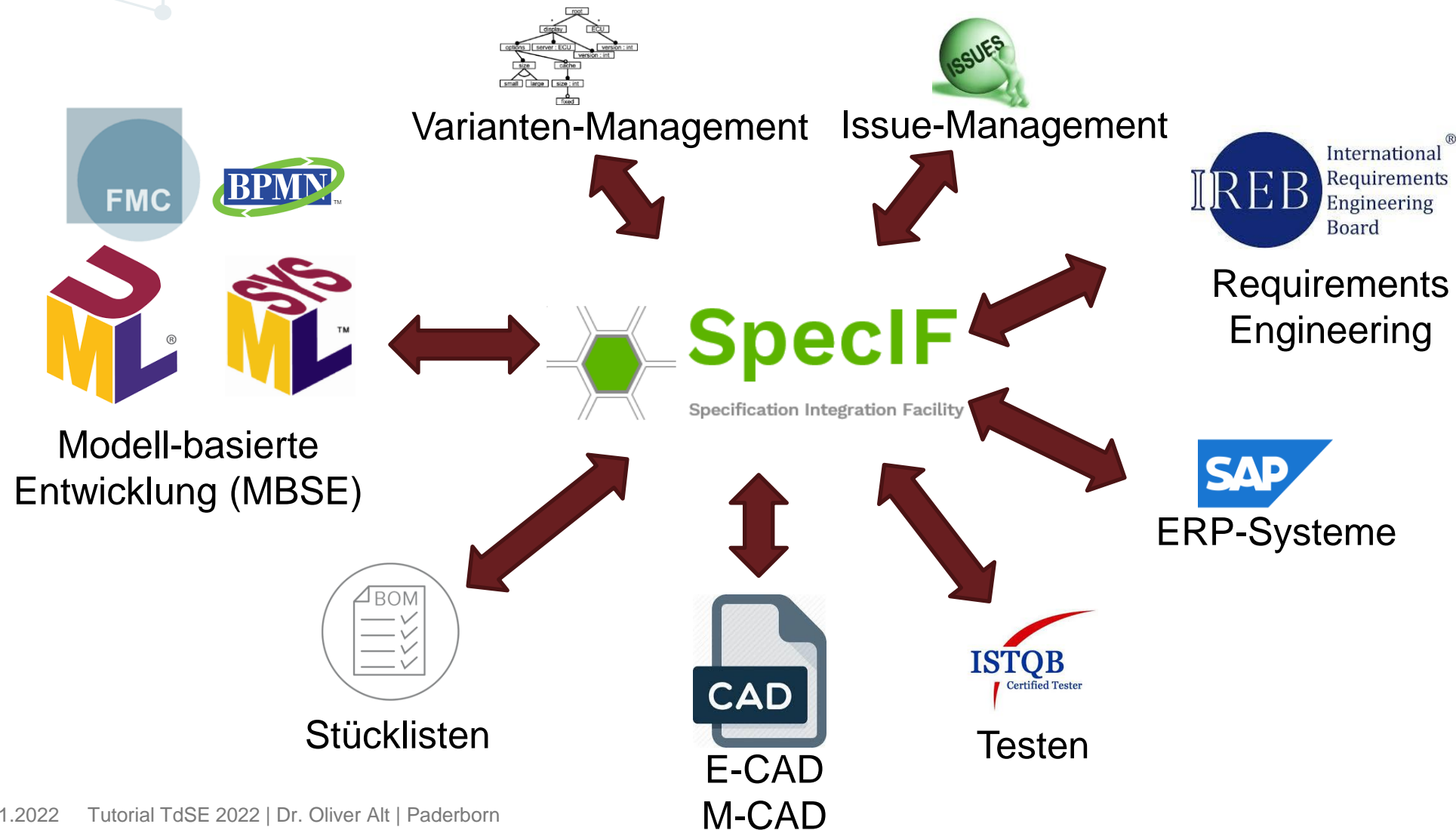
Semantische Integration durch Microservices: SpecIF und sein Web-API sind aufeinander abgestimmt



*SpecIF
User
Frontend*



Systems Engineering im Produktlebenszyklus





Fragen?

2

Die SpecIF- Terminologie

Eigenschaften - Properties

- Um konkrete Daten zu hinterlegen kann man einer Ressource oder einem Statement so genannte „Property“-Werte geben und diese mit Daten füllen
- Properties haben
 - Titel – Name der Eigenschaft
 - Datentyp – String (formatiert und unformatiert), Zahl, URL, Datum, Aufzählung, Dauer
 - Wert – Mehrsprachigkeit wird unterstützt
- Beispiel:
 - Kommentar
 - Titel [string]
 - Beschreibung [string (xhtml)]

Kommentar von O. Alt

Diese Anforderung sollte nochmal mit dem Management diskutiert werden!

Grundsätzliche Eigenschaften jedes SpecIF-Objekts

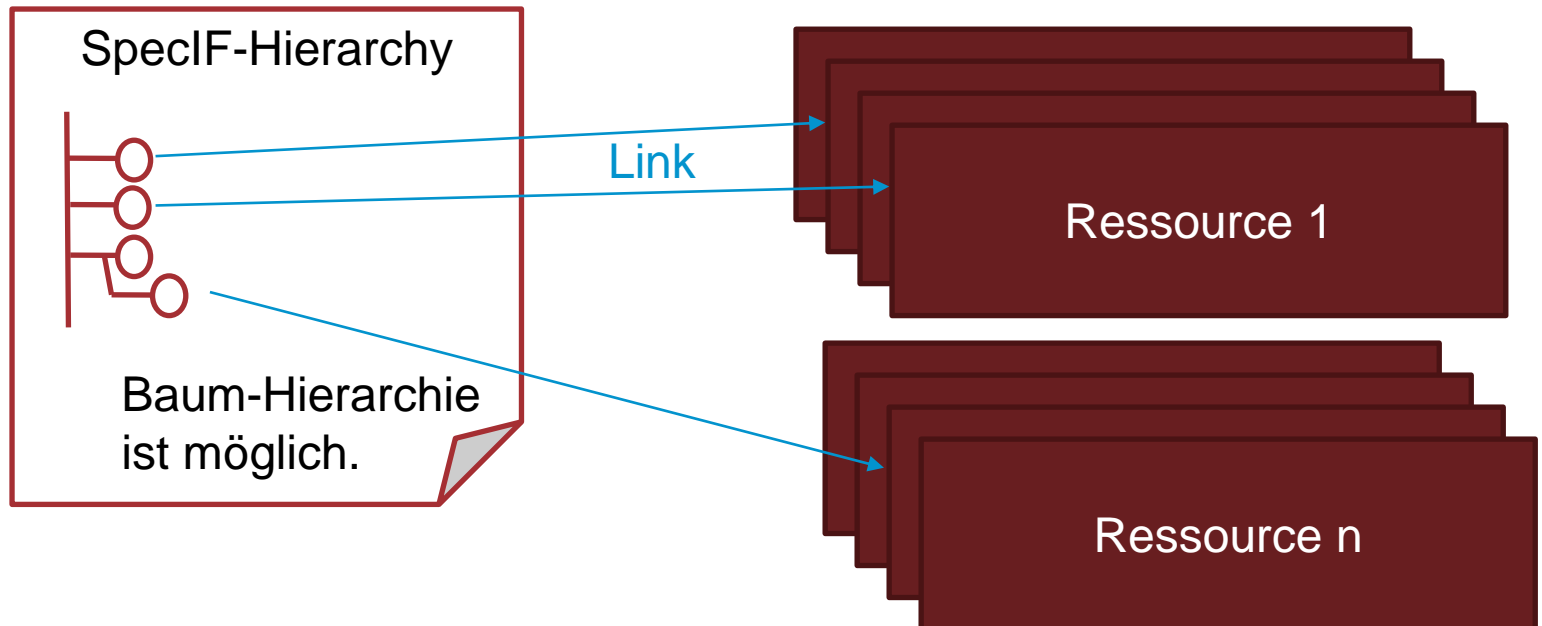
- Neben den Eigenschaftswerten (*properties*) hat jedes Element in SpecIF außerdem immer folgende implizite Eigenschaften
 - ID – Eindeutiger Identifikator
 - Typ (*class*) des Elements
 - Revisionsinformation (*revision*, *replaces*) (optional)
 - Änderungsdatum (*changedAt*)
 - Autor (*changedBy*)

Eine Ressource (JSON)

```
{  
  "id": "_73392B3D_CF8F_4ac0_BC77_E6A2C9415EF4",  
  "revision": "F16C40BE-DFC4-46BB-85C4-FDF9433F8E73",  
  "replaces": [],  
  "changedAt": "2021-03-07T11:16:21",  
  "changedBy": "oa"  
  "class": {  
    "id": "RC-Requirement",  
    "revision": "1.1"  
  },  
  "properties": [  
    ...  
  ]  
}
```

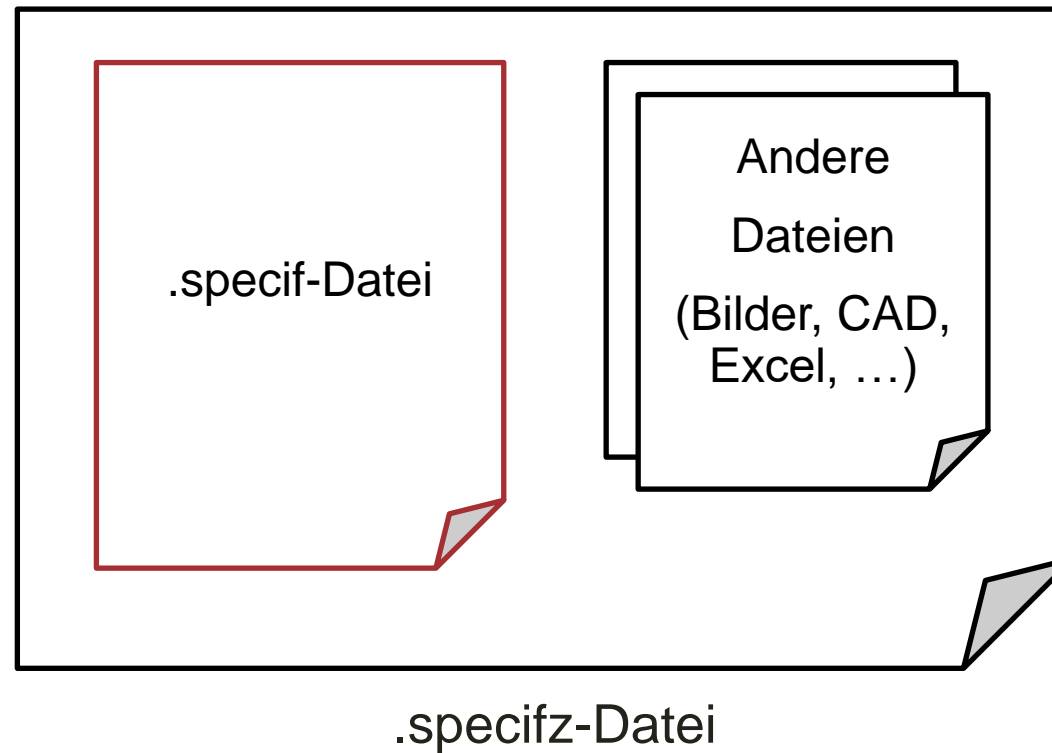
SpecIF kann Sichten und Hierarchien abbilden

- SpecIF kennt das Konzept der Trennung von Modell und Sicht
- Es lassen sich Hierarchieebenen als Sichten bilden
 - Kapitelstrukturen in Spezifikationsdokumenten
 - Navigationsbäume von Modellierungs- und CAD-Werkzeugen



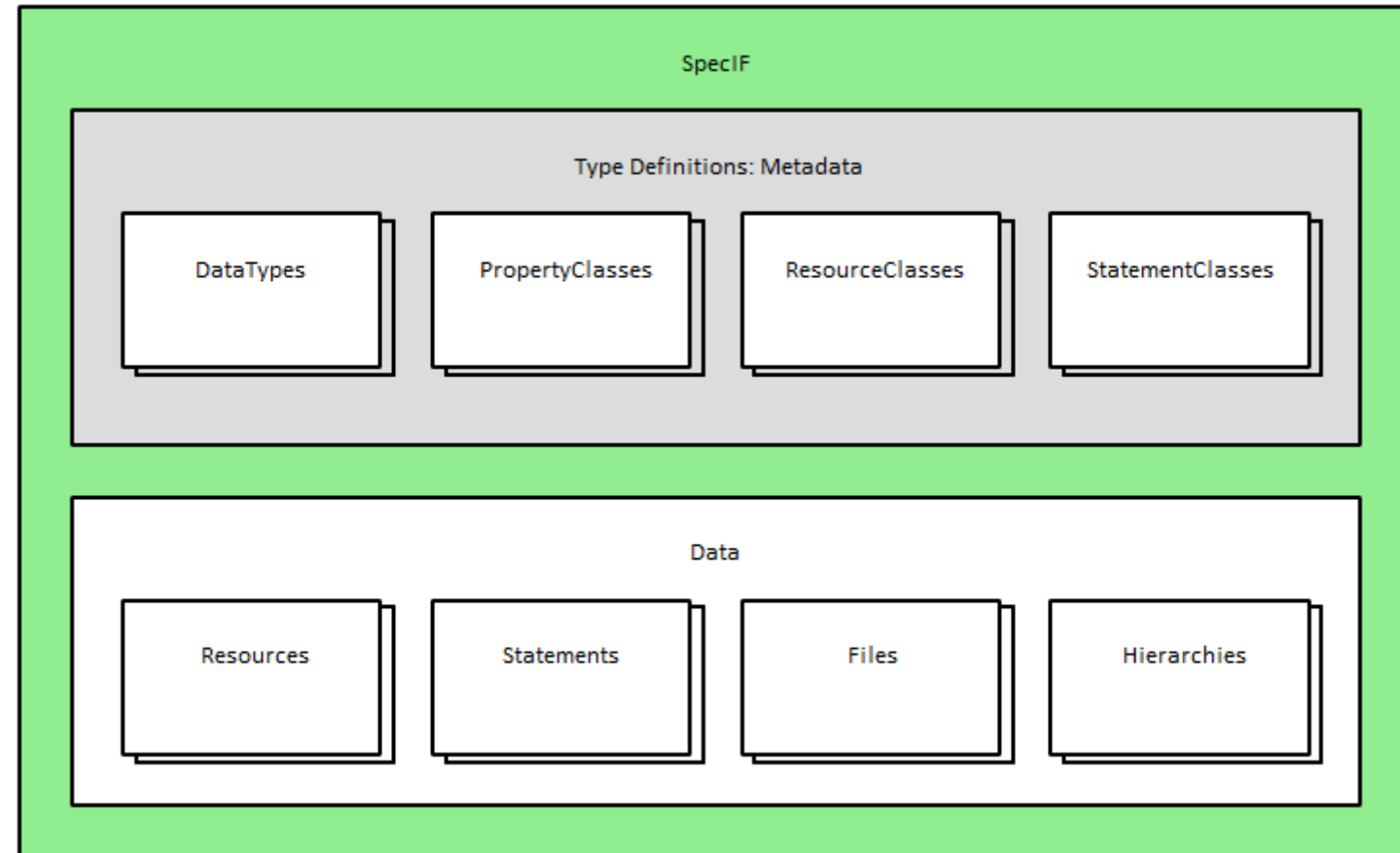
Dateien mit SpecIF transportieren

- Dateien lassen in SpecIF einbetten (ZIP-Archiv, *.specifz*) und aus SpecIF Daten verlinken



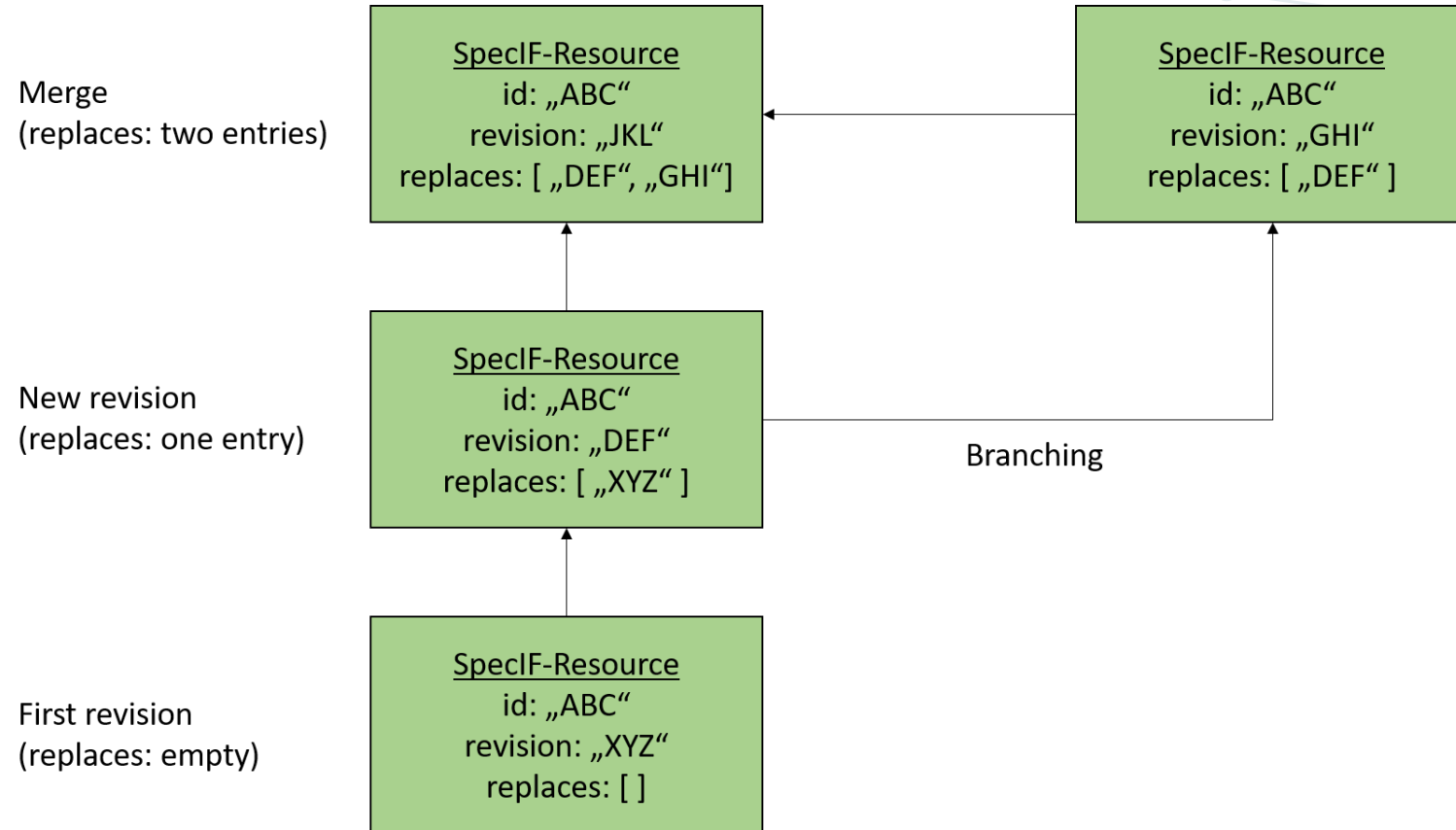
SpecIF kennt Daten und Metadaten

- Die Strukturen/Typen der mit SpecIF repräsentierten Daten lassen sich auch mit SpecIF beschreiben
 - Metadaten oder Metamodell
- Metadaten bauen aufeinander auf
- **Data Type** – primitive Datentypen
 - int, double, string, bool, URL, duration, dateTime
- **Property Class** – Definition von Properties unter Nutzung von Data Type
- **Ressource Class** – Definition eines Ressource-Typ unter Nutzung von Property Classes.
- **Statement Class** – Definition eines Statement-Typ unter Nutzung von Property Classes.

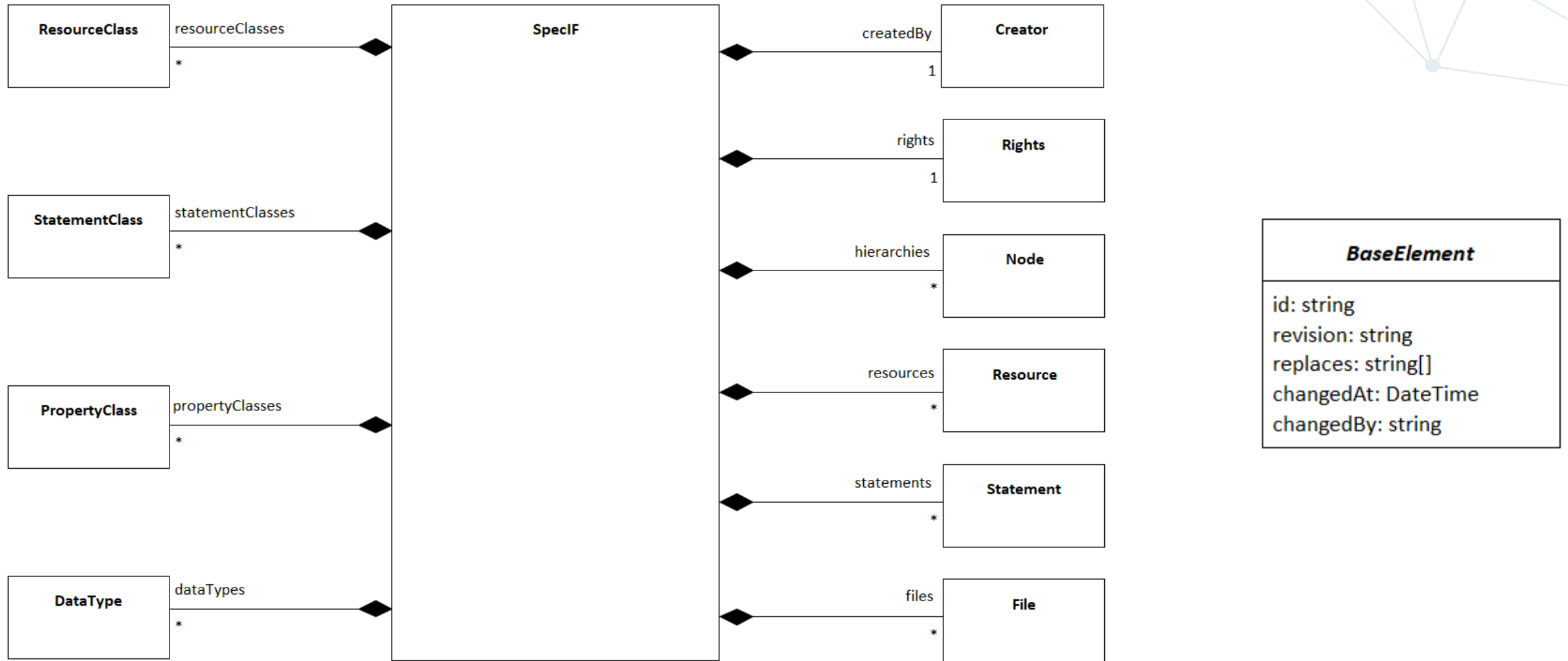


Versionierung in SpecIF

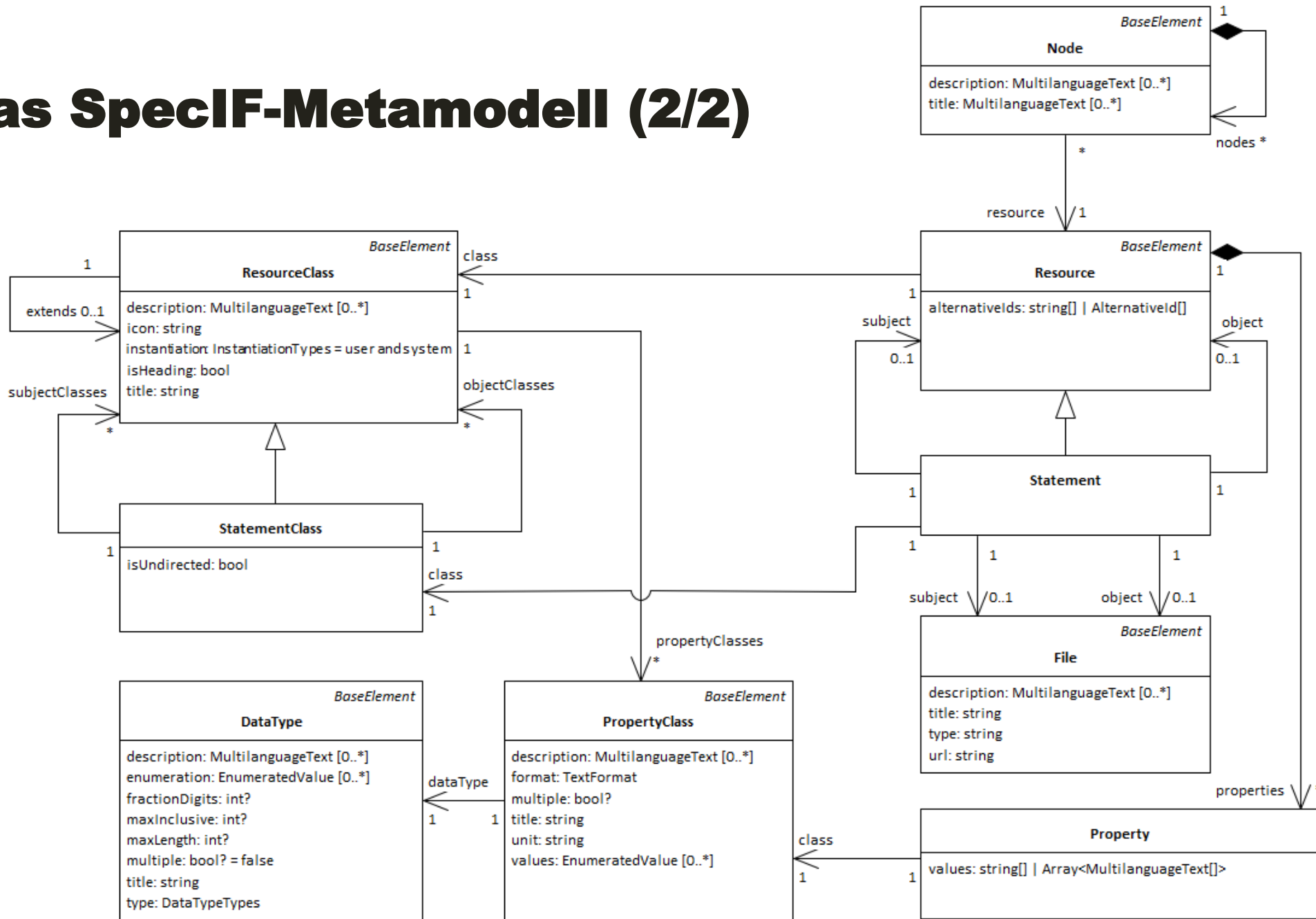
- Revisionierung basiert auf Revisions-IDs und Vorhaltung der Vorgänger und Nachfolger bei **gleichbleibender ID**
- Damit lassen sich lineare Szenarien, als auch „branch“ und „merge“ abbilden



Das SpecIF-Metamodell (1/2)



Das SpecIF-Metamodell (2/2)



Zusammenfassung – SpecIF-Terminologie



- SpecIF-Daten sind Graph-Daten
- Knoten des Graphen heißen **Ressourcen**
- Kanten des Graphen heißen **Statements** (deutsch: Aussagen)
- Eigenschaftswerte der Knoten und Kanten heißen **Property**
- Primitive Datentypen heißen **Data Type**
- Hierarchische Strukturen zur Sichtenbildung heißen **Hierarchy** (deutsch: Hierarchien)

Übung 1

Anwendung der SpecIF- Terminologiekonzepte



Übungsbeispiel: Dokumentation von Entscheidungen

- Definieren Sie eine SpecIF-Ressource und gegebenenfalls auch SpecIF-Statements mit der man (Management-)Entscheidungen innerhalb eines Projektes dokumentieren kann.
- Ziel soll es sein, dass die Entscheidung sauber und nachvollziehbar dokumentiert wird und mit relevanten anderen Arbeitsprodukten verknüpft werden kann (z.B. mit Anforderungen)

3

Semantische Integration I

SpecIF – das Vokabular

- Damit Daten austauschbar und ohne manuelle Zuordnung interpretiert werden können, definiert die SpecIF eine Sammlung von Vokabularbegriffen mit feststehender Bedeutung (Semantik)
- Diese Semantik ist fester Teil einer SpecIF-Freigabe
- Die Begriffe des Vokabulars verwenden bereits gebräuchliche Begriffe – aus bestehenden Standards - wo immer dies möglich ist
- Namensräume (*namespaces*) erlauben die Zuordnung von Begriffen zu deren Herkunft
 - Z.B. „dcterms:title“ - Ein Begriff aus der Dublin Core Terminologie



SpecIF 1.1 – Primitive Datentypen

Boolean	The Boolean data type.
Byte	A byte is an integer value between 0 and 255.
Integer	A numerical integer value from -32768 to 32767.
Real	A floating point number (double).
Real with 2 Decimals	A floating point number (double) with two fraction digits.
Date or Timestamp	Date or timestamp in ISO-format
String[256]	String with max. length 256
Plain or formatted Text	Descriptive text represented in plain or rich text using XHTML.
URL	A uniform resource locator.
E-mail	Data type to represent an e-mail address.

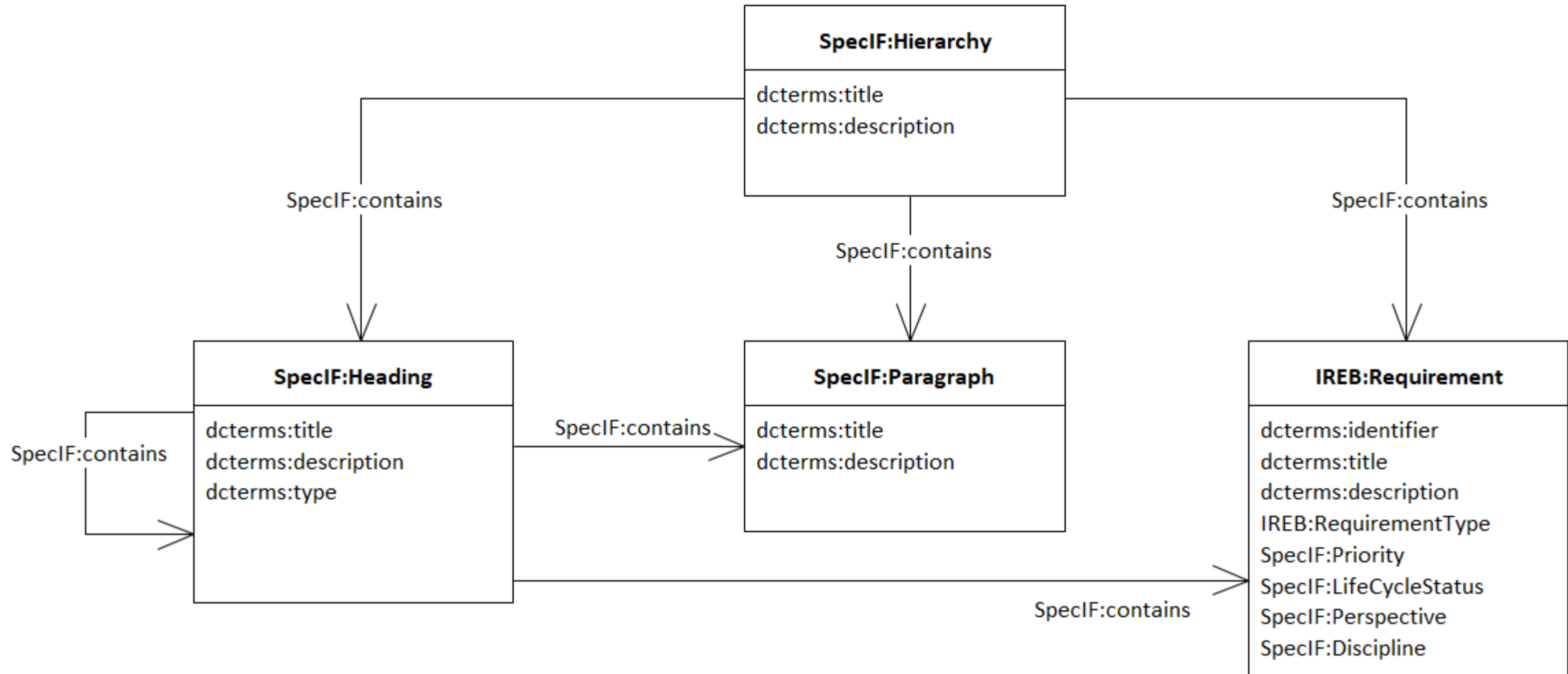
SpecIF 1.1 - Enumerationsdatentypen

SpecIF:LifeCycleStatus	Enumerated values for status
SpecIF:Priority	Enumerated values for priority
SpecIF:Discipline	Enumerated values for engineering discipline
IREB:RequirementType	Enumerated values for the requirement type according to IREB.
Perspective	Enumerated values for the perspective (of a requirement)
SpecIF:VisibilityKind	Enumerated values for visibility.

SpecIF 1.1 – Property-Klassen

dcterms:identifier	A unique reference to the resource within a given context.
dcterms:title	A name given to the resource.
dcterms:description	An account of the resource. Descriptive text about resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML <div> element. (source: OSLC)
SpecIF:Origin	The origin (source, reference) of an information or requirement.
SpecIF:Diagram	A partial graphical representation (diagram) of a model.
SpecIF:Notation	The notation used by a model diagram, e.g. 'BPMN:2.0', 'OMG:SysML:1.3:Activity Diagram' or 'FMC:Block Diagram'.
SpecIF:LifeCycleStatus	The 'status', e.g. lifecycle state, of the resource.
SpecIF:Priority	The 'priority' of the resource.
SpecIF:Discipline	The engineering discipline (system, electronics, mechanics, software, safety).
SpecIF:Responsible	The 'person' being responsible for the resource.
SpecIF:DueDate	A 'due date' for the resource.
UML:Stereotype	A stereotype gives an element an additional/different meaning.
SpecIF:Abbreviation	An abbreviation for the resource or statement.
dcterms:type	The element type resp. the metamodel element (e.g. OMG:UML:2.5.1:Class)
SpecIF:Alias	An alias name for the resource.
rdf:value	A value of different meaning, depending on the element type (attribute default value, a taggedValue value etc.).
IREB:RequirementType	Enumerated value for the requirement type according to IREB.
SpecIF:Perspective	Enumerated values for the perspective (of a requirement).
SpecIF:Visibility	The visibility of a resource (e.g. public, private, protected,...) as known from object orientation.
SpecIF:ImplementationLanguage	The name of an used implementation language (e.g. C, C++, C#, Java, ADA, OCL, ALF etc.).

SpecIF 1.1 – Resource-Typen für Dokumente



SpecIF 1.1 – Resource-Typen für grafische Modelle



UML:Package
dcterms:description dcterms:title dcterms:type rdf:value SpecIF:Alias SpecIF:LifeCycleStatus UML:Stereotype

SpecIF:Diagram
dcterms:description dcterms:title dcterms:type SpecIF:Diagram SpecIF:LifeCycleStatus SpecIF:Notation UML:Stereotype

FMC:Event
dcterms:description dcterms:title dcterms:type rdf:value SpecIF:Alias SpecIF:LifeCycleStatus UML:Stereotype

FMC:Actor
dcterms:description dcterms:title dcterms:type rdf:value SpecIF:Alias SpecIF:LifeCycleStatus UML:Stereotype

FMC:State
dcterms:description dcterms:title dcterms:type rdf:value SpecIF:Alias SpecIF:LifeCycleStatus UML:Stereotype

SpecIF:Feature
dcterms:description dcterms:identifier dcterms:title SpecIF:Discipline SpecIF:LifeCycleStatus SpecIF:Perspective SpecIF:Priority

IREB:Requirement
dcterms:description dcterms:identifier dcterms:title IREB:RequirementType SpecIF:Discipline SpecIF:LifeCycleStatus SpecIF:Perspective SpecIF:Priority

SpecIF:Collection
dcterms:description dcterms:title dcterms:type rdf:value SpecIF:Alias SpecIF:LifeCycleStatus UML:Stereotype

SpecIF 1.1 – Statement-Klassen (1/2)

Titel	Beschreibung
rdf:type	States that the relation subject is an instance of the relation object.
SpecIF:refersTo	A resource 'refers to' any other resource.
SpecIF:dependsOn	Requirement/feature depends on requirement/feature.
SpecIF:duplicates	The subject requirement duplicates the object requirement.
SpecIF:contradicts	The subject requirement contradicts the object requirement.
IREB:refines	The subject requirement refines the object requirement.

SpecIF 1.1 – Statement-Klassen (2/2)

Titel	Beschreibung
SpecIF:shows	Statement: Plan resp. diagram shows model element.
SpecIF:contains	Statement: Model element contains model element.
SpecIF:serves	Statement: An actor serves an actor.
SpecIF:stores	Statement: Actor (role, function) writes and reads state (information).
SpecIF:writes	Statement: Actor (role, function) writes state (information).
SpecIF:reads	Statement: Actor (role, function) reads state (information).
SpecIF:influences	Statement: A state (information) influences a state (information).
SpecIF:precedes	An FMC:Actor 'precedes' an FMC:Actor or an FMC:Actor 'precedes' an FMC:Event or an FMC:Event 'precedes' an FMC:Actor. The rdf:type property specifies if it is a simple precedes, a SpecIF:signals or a SpecIF:triggers.
oslc_rm:satisfies	Statement: Model element satisfies requirement.
SpecIF:allocates	Statement: Model element is allocated to model element. The semantics are equal to allocation in SysML or deployment relation in UML.
SpecIF:implements	A FMC:Actor or FMC:State 'implements' a SpecIF:Feature.
SpecIF:isAssociatedWith	The subject is associated with the object.
SpecIF:isSpecializationOf	A term is a specialization of another, such as 'Passenger Car' and 'Vehicle'.

Weitere Klassen sind in Diskussion

- Für neue SpecIF-Freigaben werden aktuell weitere Klassen diskutiert und evaluiert
 - Testen
 - Featuremodellierung/Variantenmanagement
 - Vokabularbeschreibung
 - Issue-Management
 - Bill of Materials (BOM)
 - Echtzeitintegration

Übung 2

Anwendung der Vokabularbegriffe



Anwendung der Vokabularbegriffe

- Wenden Sie so gut wie möglich die gezeigten Vokabularbegriffe auf die Definitionen zur Entscheidungsdokumentation aus Übung 1 an.
- Insbesondere sollten sich die Property-Klassen verwenden lassen.

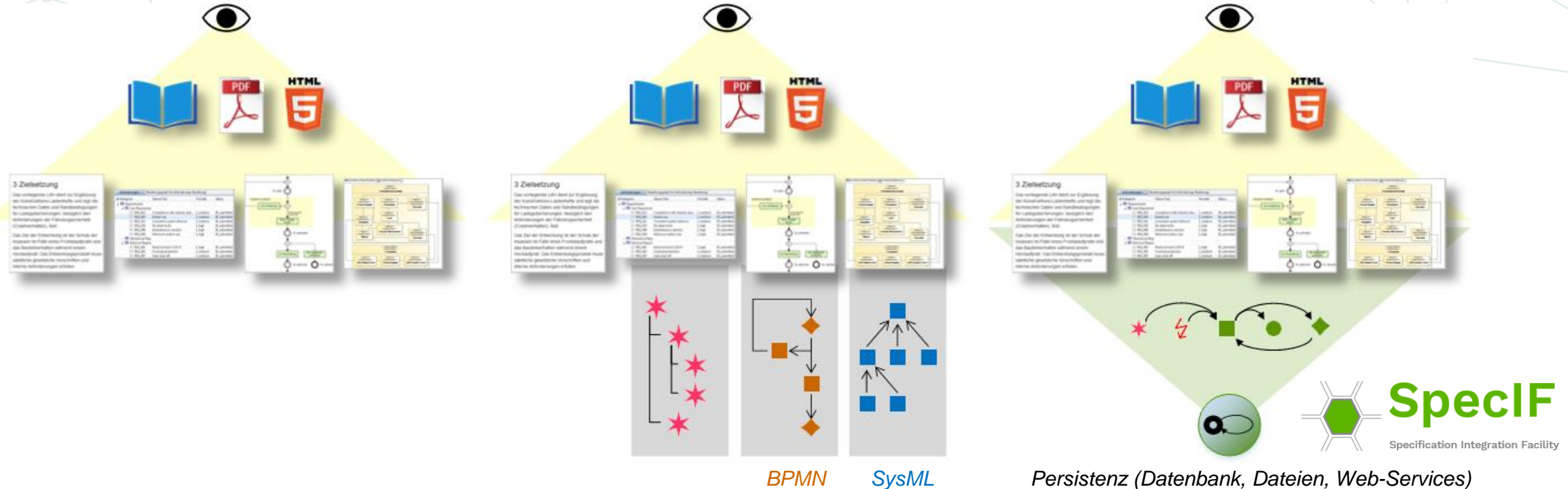
SpecIF 1.1 – Property-Klassen

dcterms:identifier	A unique reference to the resource within a given context.
dcterms:title	A name given to the resource.
dcterms:description	An account of the resource. Descriptive text about resource represented as rich text in XHTML content. SHOULD include only content that is valid and suitable inside an XHTML <div> element. (source: OSLC)
SpecIF:Origin	The origin (source, reference) of an information or requirement.
SpecIF:Diagram	A partial graphical representation (diagram) of a model.
SpecIF:Notation	The notation used by a model diagram, e.g. 'BPMN:2.0', 'OMG:SysML:1.3:Activity Diagram' or 'FMC:Block Diagram'.
SpecIF:LifeCycleStatus	The 'status', e.g. lifecycle state, of the resource.
SpecIF:Priority	The 'priority' of the resource.
SpecIF:Discipline	The engineering discipline (system, electronics, mechanics, software, safety).
SpecIF:Responsible	The 'person' being responsible for the resource.
SpecIF:DueDate	A 'due date' for the resource.
UML:Stereotype	A stereotype gives an element an additional/different meaning.
SpecIF:Abbreviation	An abbreviation for the resource or statement.
dcterms:type	The element type resp. the metamodel element (e.g. OMG:UML:2.5.1:Class)
SpecIF:Alias	An alias name for the resource.
rdf:value	A value of different meaning, depending on the element type (attribute default value, a taggedValue value etc.).
IREB:RequirementType	Enumerated value for the requirement type according to IREB.
SpecIF:Perspective	Enumerated values for the perspective (of a requirement).
SpecIF:Visibility	The visibility of a resource (e.g. public, private, protected,...) as known from object orientation.
SpecIF:ImplementationLanguage	The name of an used implementation language (e.g. C, C++, C#, Java, ADA, OCL, ALF etc.).

4

Semantische Integration II

Das Auge sieht zwar das gleiche ... doch auf den Unterbau kommt es an



Das „Sichtbare“ erstellen

- Texte editieren und Diagramme zeichnen
- Erfordert Hirn und Disziplin, um Konsistenz zu erhalten

Partial-Modellierung

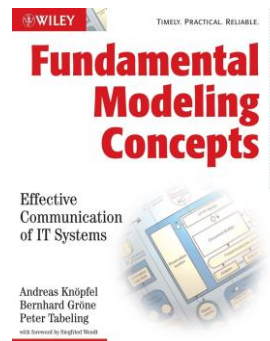
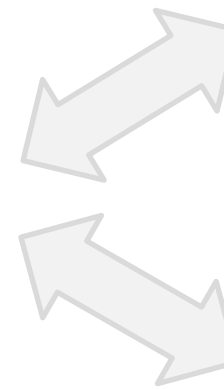
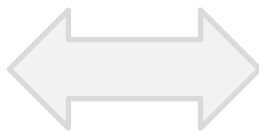
- Werkzeugunterstützung für Teilmodelle
- Teilmodelle sind leichter in sich konsistent zu halten

Modell-Integration

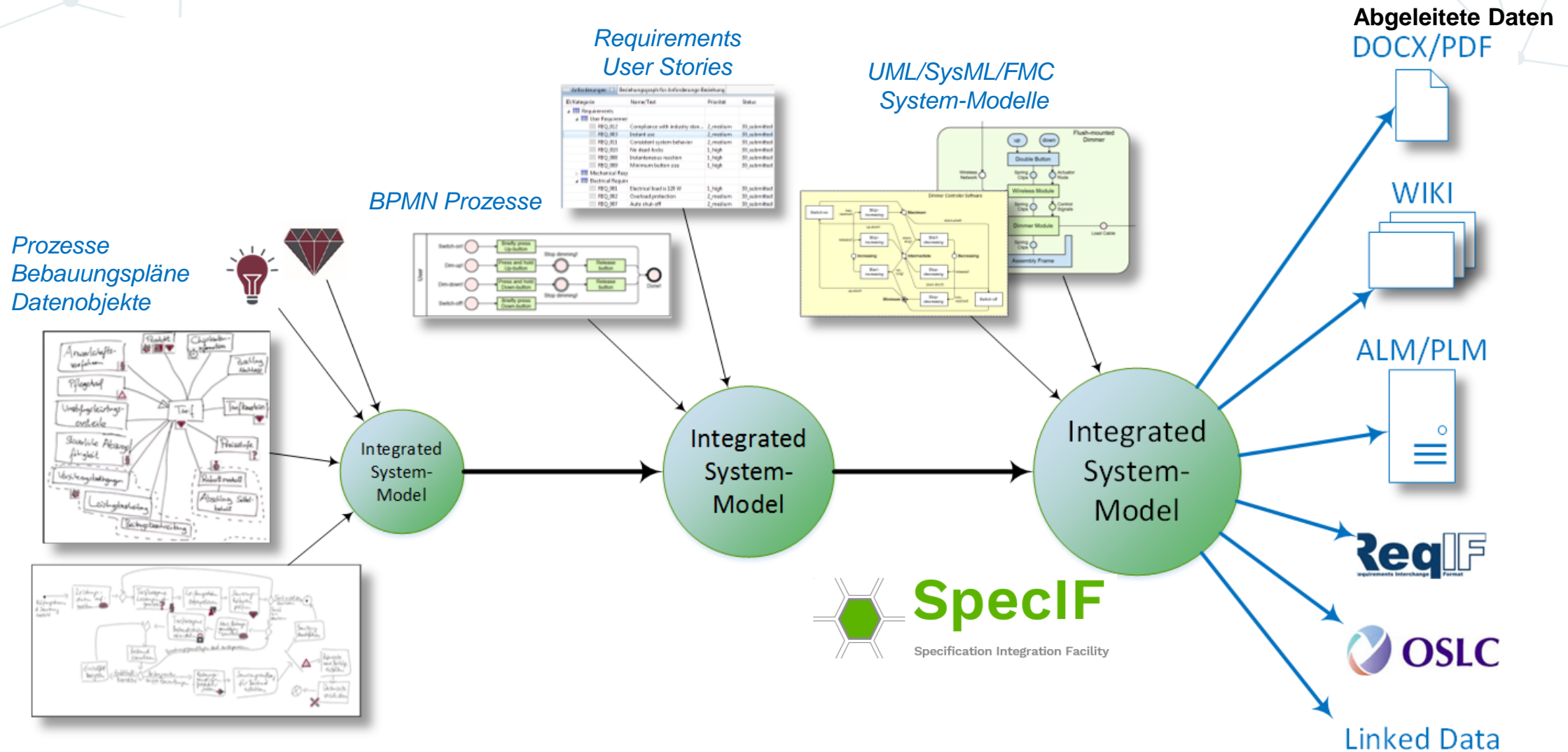
- Werkzeugunterstützung für Teilmodelle
- Elemente aller Sichten sind semantisch vernetzt

Das SpecIF-Integrationsmodell

- Für Modellierungen im Rahmen des Systems Engineering definiert die SpecIF ein Integrationsmodell, bestehend aus wenigen Elementen
- **Alle spezifischen Modelle werden auf dieses Integrationsmodell semantisch abgebildet**
- Das Integrationsmodell basiert auf dem Konzept der **Fundamental Modeling Concepts (FMC)**, definiert durch Prof. S. Wendt in den 1970er Jahren

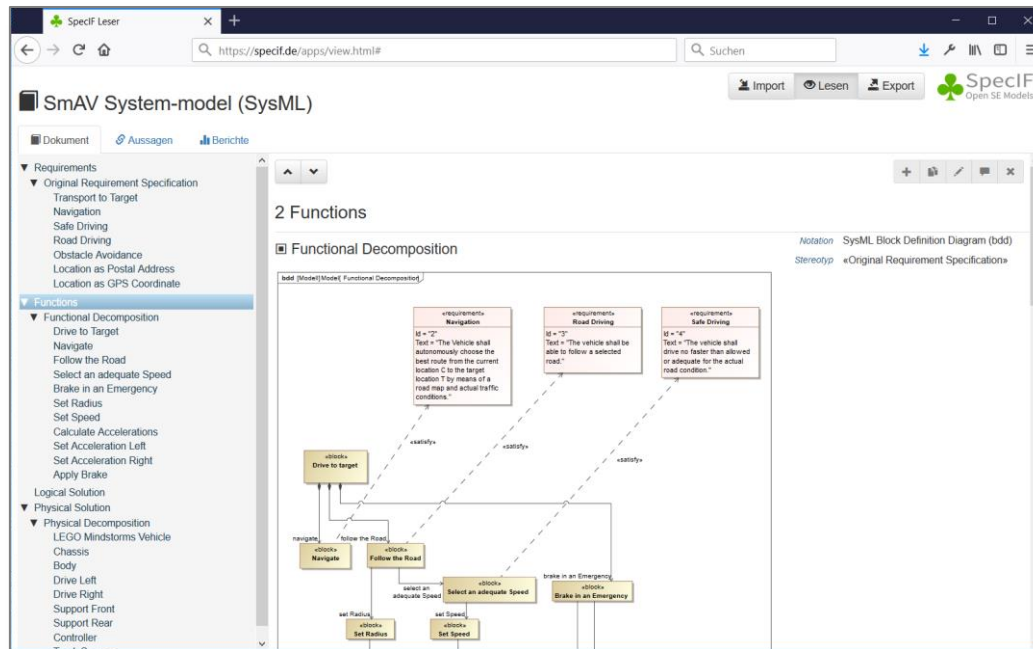


Teilmodelle schrittweise zusammen führen ...



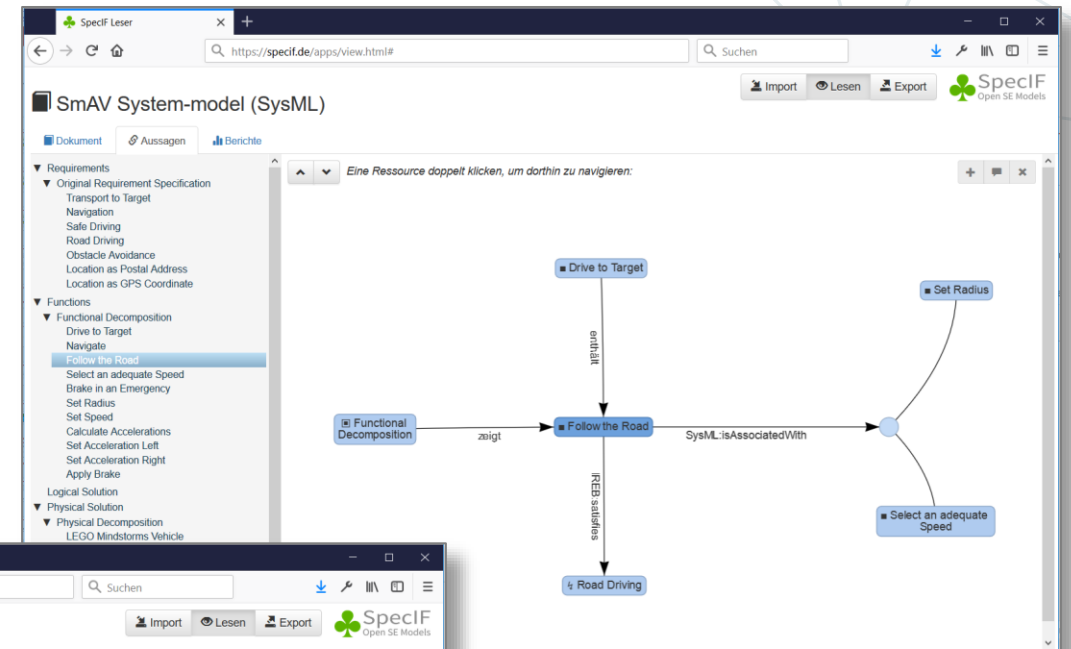
...und übergreifend navigieren, suchen und prüfen

Beispiel: Small Autonomous Vehicle (FH Esslingen)




Alle Modell-Diagramme

Alle Modell-Elemente



Alle Beziehungen



Transport to Target	Stereotyp	«Function»
Navigation		
Safe Driving		
Road Driving		
Obstacle Avoidance		
Location as Postal Address		
Location as GPS Coordinate		
▼ Functions		
▼ Functional Decomposition		
Drive to Target		
Navigate		
Follow the Road		
Select an adequate Speed		
Brake in an Emergency		
Set Radius		
Calculate Accelerations		
Set Acceleration Left		
Set Acceleration Right		
Apply Brake		
Logical Solution		
Physical Solution		
LEGO Mindstorms Vehicle		
Chassis		
Body		
Drive Left		
Drive Right		
Support Front		
Support Rear		
Controller		
Speed Controller		

■ Drive to Target	Stereotyp	«Function»
Manoeuvre the vehicle from the current position to the selected target.		
■ Navigate	Stereotyp	«Function»
Select a path to get from the current position to the target.		
■ Follow the Road	Stereotyp	«Function»
Keep the vehicle on the selected road.		
■ Select an adequate Speed	Stereotyp	«Function»
Taking into account road condition and weather, select a safe speed.		
■ Brake in an Emergency	Stereotyp	«Function»
Take the vehicle to a full stop in the shortest time possible without losing control over the trajectory.		
■ Set Radius	Stereotyp	«Function»

Resourcetypen des SpecIF-Integrationsmodells

Fundamentale Modellierungselemente (FMC)

● State

Ein Zustand (State) ist ein **passives** Modellelement, das einen aktuellen Systemzustand repräsentieren kann. Es kann selbst am Modellzustand nichts verändern.

- Zustände eines Statecharts
- Objektvariablen
- Klassendefinitionen

■ Actor

Ein Akteur (Actor) ist ein **aktives** Modellelement, das ausgeführt werden und einen Systemzustand verändern kann.

- Aktivitäten und Aktionen eines Aktivitätsdiagramms
- Zustandsmaschinen
- Komponenten
- Anwendungsfälle
- Operationen

◆ Event

Ein Ereignis (Event) ist ein Modellelement zur Modellierung von **Zeitereignissen**, bedingter Ablaufsteuerung oder genereller ein Synchronisationselement für Verhaltensabläufe.

- Zeitereignisse
- Signale
- Initiale und finale Ablaufknoten in Zustandsmaschinen und Aktivitätsdiagrammen

Resourcetypen des SpecIF-Integrationsmodells

Sichtenbildung und Strukturierung

Diagram

Ein Diagramm (Diagram) bildet eine **Sicht auf** eine Auswahl von **Elementen** des Modells in dem es diese und deren Beziehungen grafisch darstellt.

- UML/SysML-Diagramme
- BPMN-Diagramme
- FMC-Diagramme
- Schaltpläne

Package

Ein Paket (Package) bringt eine **hierarchische Struktur** in die **Elemente eines Modells** und dient zumeist der Vereinfachung der Navigation der Modellbenutzer durch die Modellelemente.

- UML/SysML-Pakete
- Ordner in Dateisystemen

Collection

Eine Sammlung (Collection) fasst innerhalb eines Diagramms Elemente zusammen und bringt diese in einen logischen Zusammenhang.

- UML/SysML-Boundaries
- BPMN-Collections

Resourcetypen des SpecIF-Integrationsmodells

Anforderung und Feature

⚡ Requirement

Eine Anforderung (Requirement) ist eine **textuelle Beschreibung** einer im Rahmen der Systementwicklung zu realisierenden Eigenschaft.

- SysML-Requirements
- Anforderungen aus allen möglichen Anforderungsmanagementsystemen (DOORS, Polarion, Jira etc.)

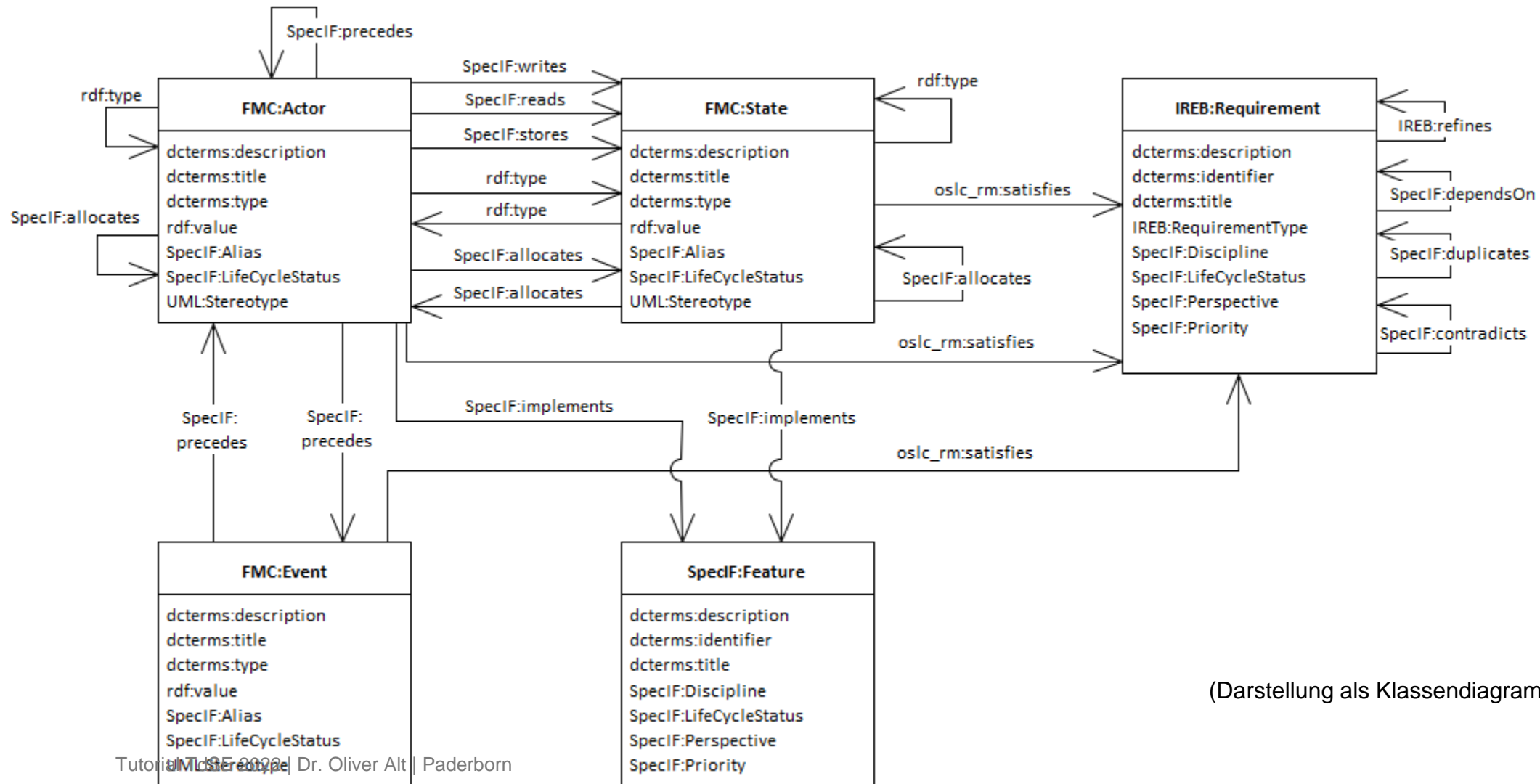
★ Feature

Ein Merkmal (Feature) ist eine charakteristische Eigenschaft eines Systems oder Produkts, welche das System oder Produkt von anderen ähnlichen unterscheidet.

- Features aus Featurelisten im Vertrieb
- Features aus Featurebäumen im Variantenmanagement

Statements des SpecIF-Integrationsmodells

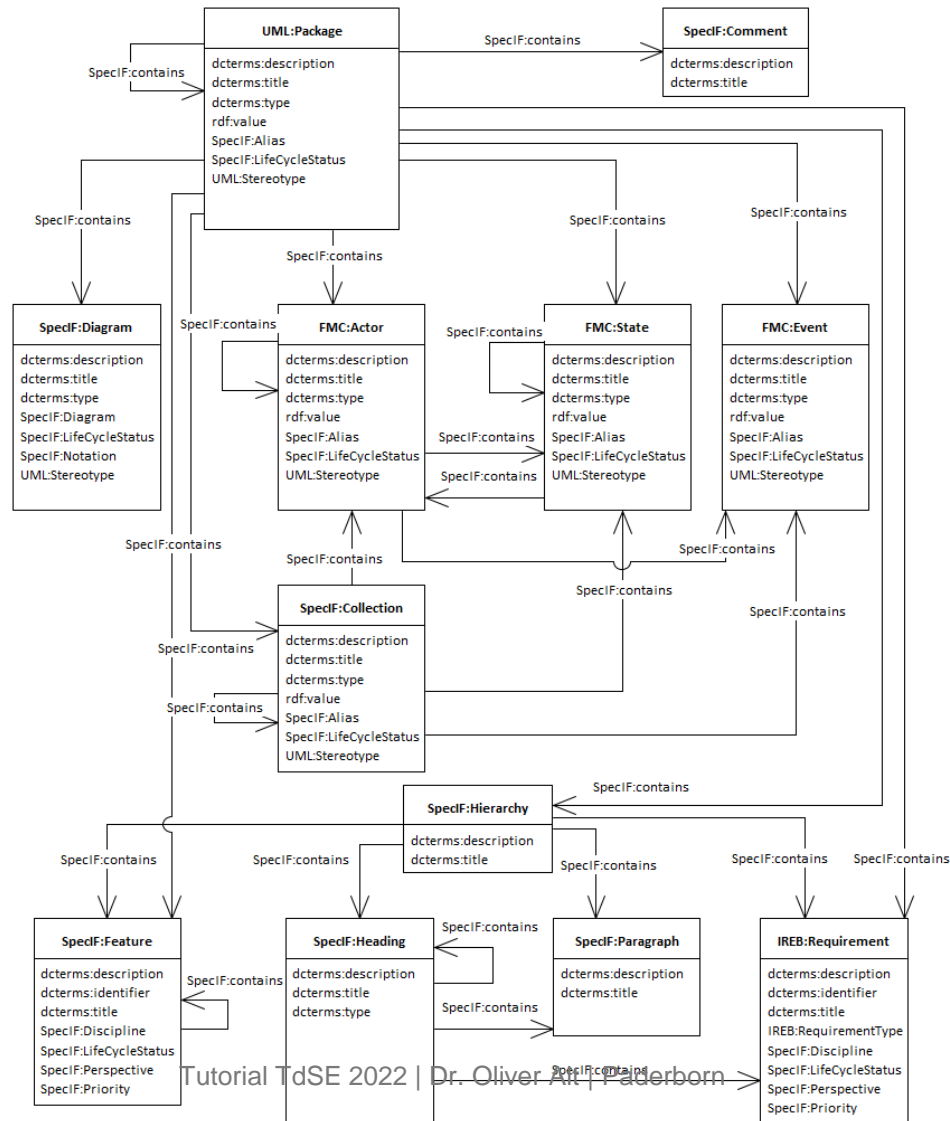
Modellierung von semantischen Zusammenhängen



(Darstellung als Klassendiagramm)

Statements des SpecIF-Integrationsmodells

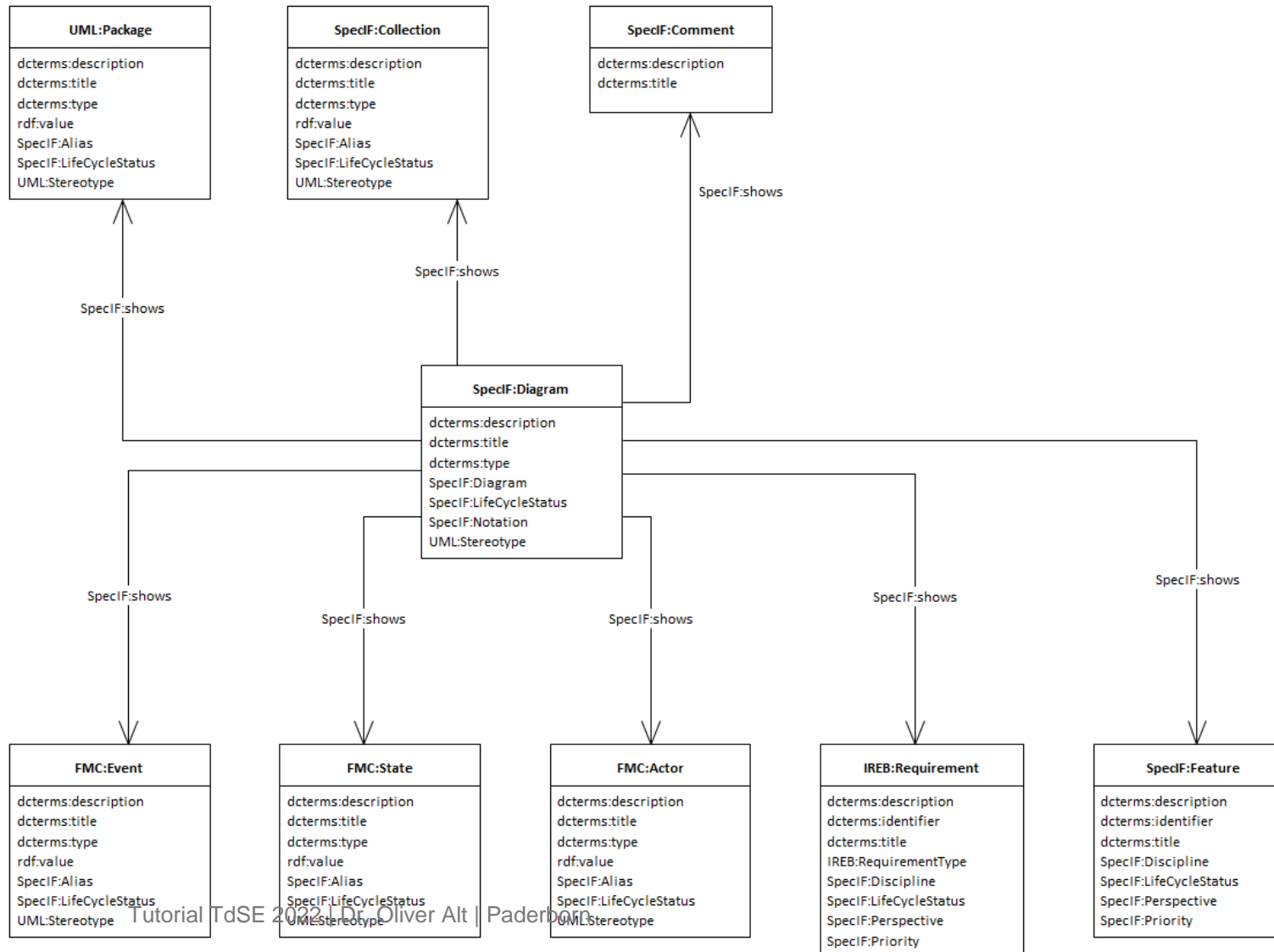
Modellierung von strukturellen Zusammenhängen



Nutzung des Contains-Statement

Statements des SpecIF-Integrationsmodells

Modellierung von Sichtbarkeit auf Diagrammen



Nutzung des
Shows-Statement

SpecIF-Daten bilden ein semantisches Netz

- Die SpecIF-Daten bilden einen großen Graphen bestehend aus Ressourcen und Statements, die miteinander in prädikatenlogischen Beziehungen stehen
- Es entsteht das **semantische Netz**
- Man kann nun durch ein solches Netz navigieren und logische Schlüsse daraus herleiten

Wie werden Diagramme repräsentiert?

- SpecIF nutzt den bestehenden SVG (Scalable Vector Graphics) Standard zum Speichern und Austauschen von 2D Diagrammdateien
 - Kann mit jedem Web-Browser dargestellt werden
- Die SVG-Daten können durch semantische Metadaten angereichert werden
 - Dadurch können Diagramme verlustfrei ausgetauscht werden (basiert in Teilen auf dem OMG Diagram Exchange Standard)

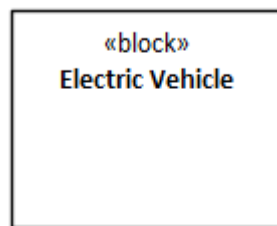
```
<g class="specif-resource-diagram-element">
  <metadata>
    <specif:shape>
      <specif:resourceReference id="_65624FBC_FE08_4dad_971E_5D26EF996570"
                             revision="5d3656420cdeb3e4a88dd9927236dbdc23a9b946" />
      <dc:Bounds x="100" y="210" width="100" height="80" />
    </specif:shape>
  </metadata>
  ... <!-- graphical SVG data -->
```

Modelldatenaustausch und -datenintegration mit SpecIF

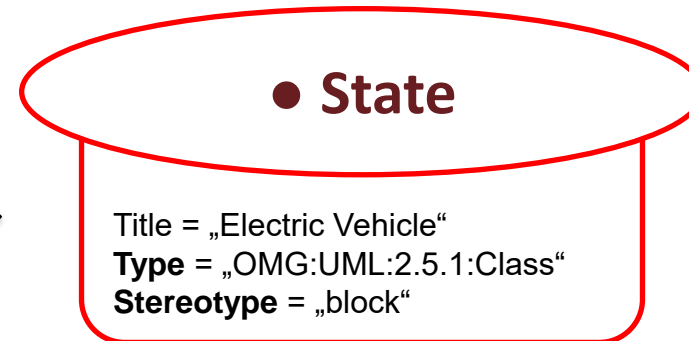
- Für den Austausch von Modelldaten mit SpecIF werden immer die gleichen Prinzipien angewandt:
 - Transformation der nativen Modelldaten in das SpecIF-Integrationsmodell
 - Transformation aller Diagramme nach SVG mit SpecIF Metadaten
- Es entsteht das semantische Netz des Modells – auch gefüttert aus mehreren Quellen
- Diese Daten lassen sich jederzeit semantisch korrekt interpretieren und in die nativen Werkzeugformate zurück verwandeln

Wie kann bei Abbildung auf die fundamentalen Modellelemente eine Rückintegration gelingen?

- Alle nativen Modelldaten werden in das SpecIF-Integrationsmodell gewandelt
 - Hier werden nur wenige Elementtypen verwendet
- Um eine Rückintegration zu ermöglichen, werden notwendige native Informationen mit übergeben
 - Properties *Type* und *Stereotype*
- Beispiel – der SysML Block:



SysML



SpecIF





Fragen?

5

Praktische Anwendung



Zusammenfassung



- Wir glauben, dass mit SpecIF ein übergreifender Datenaustausch und Datenintegration gelingen kann
 - Einfachheit
 - Verwendung bestehender Technologien und Standards wo immer möglich
 - Open Source Ansatz
 - Semantische Integration
 - Referenzimplementierungen
 - Specif.de
 - Github GfSE und oalt

Werden Sie Teil der SpecIF-Open Source Initiative durch Mitarbeit oder Finanzierung



– Program Management

- Definieren Sie Features mit und bestimmen Sie deren Priorität
- Werden Sie Teil des SpecIF Technologieboard oder „Product Owner“
- Mitglieder: Tragen durch „man-power“ oder Finanzierung bei

– Agile Development Teams

- Bis zu 7 Teammitglieder pro Team für Entwicklung und Test

– Repository

- GitHub open-source Software mit Apache oder MIT Lizenz

– Resultat

- Community Edition = Modulare Referenzimplementierung
- ... als Basis kommerzieller Softwareprodukte

SpecIF
Program Management

