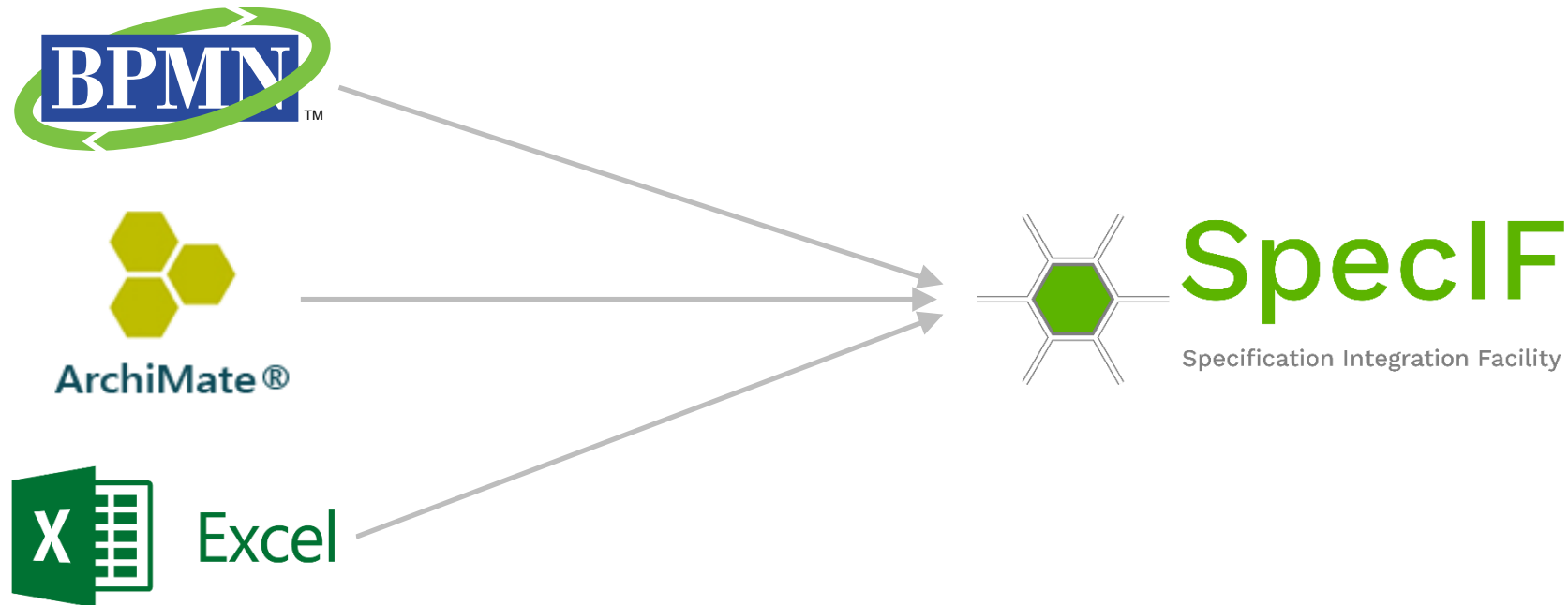


Integrate BPMN and Archimate Models using SpecIF

TdSE 2021

Oskar von Dungern, Dr.-Ing., enso managers GmbH

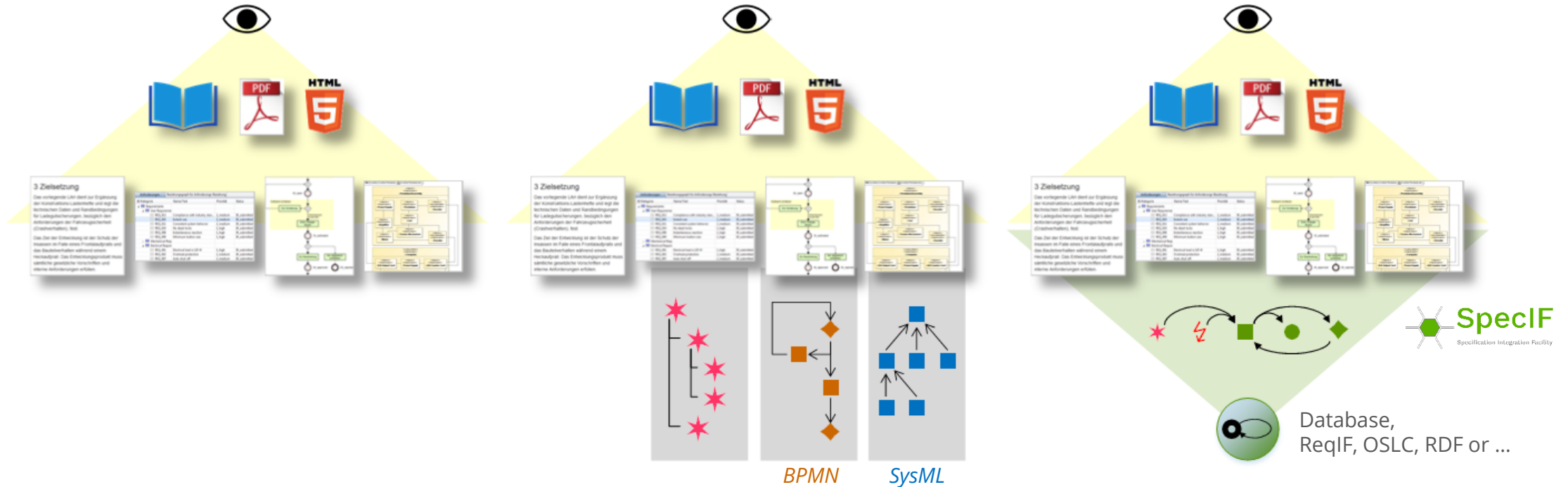


Basic Assumptions

- There will be always specialized tools for different purposes
- It is unwise to require collaborators to use certain tools or even a single tool
- Yet, there is an interest
 - to navigate, search and audit partial results in a *common context*
 - to exchange model information between organizations and tools

→ That's where SpecIF kicks in: Specification Integration Facility

The eye sees the same – behind the surface it gets interesting



Creating the „Visible“

- Text editing and image „drawing“
- Needs brain and discipline to build and keep consistent

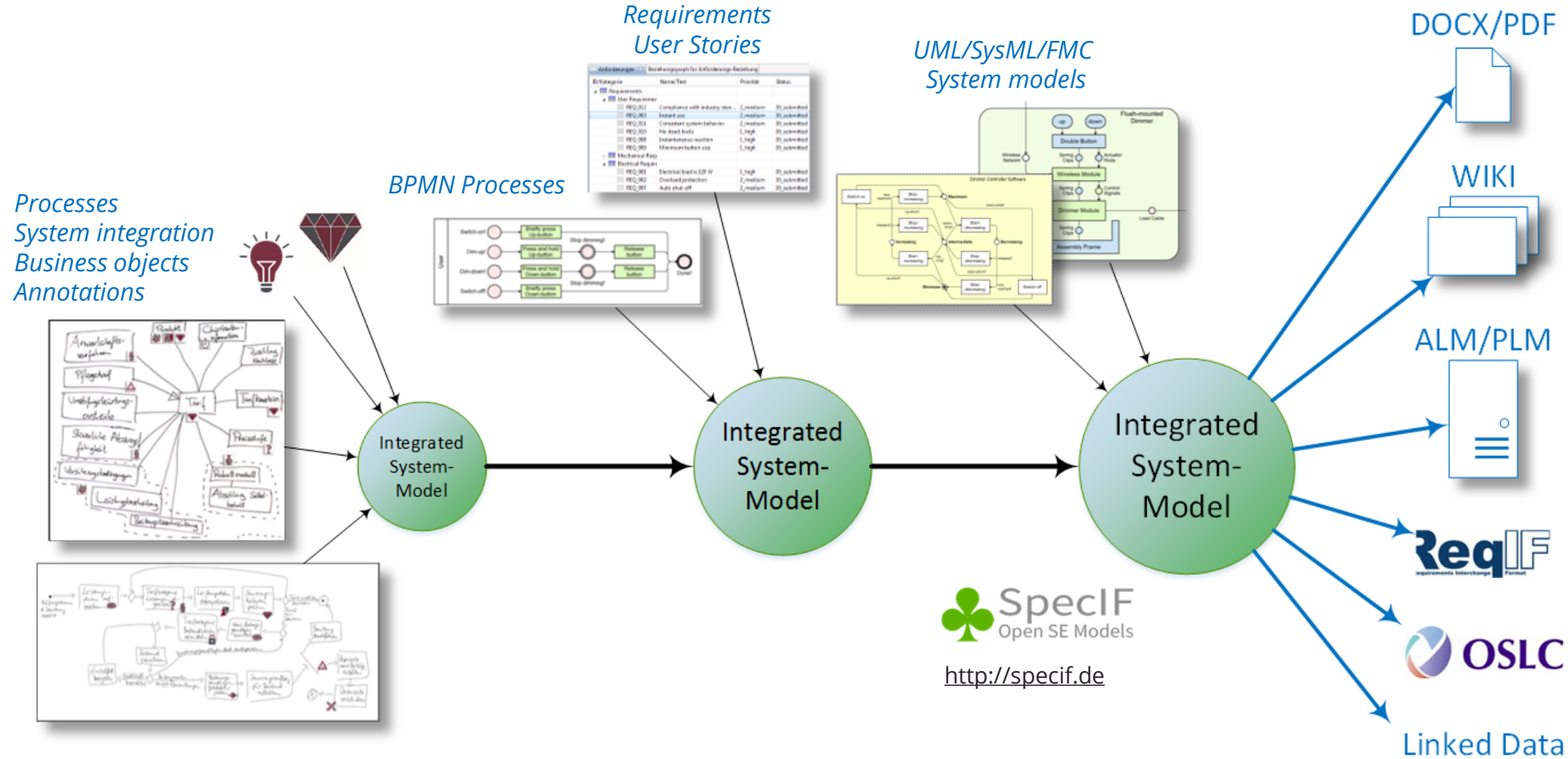
Partial Modelling

- Text editing and modelling per method
- Tool support within the methods

Model Integration

- Text editing and modelling per method
- Elements in all views are interrelated by a semantic net

Add partial models step-by-step ...



Five Principles of Model Integration

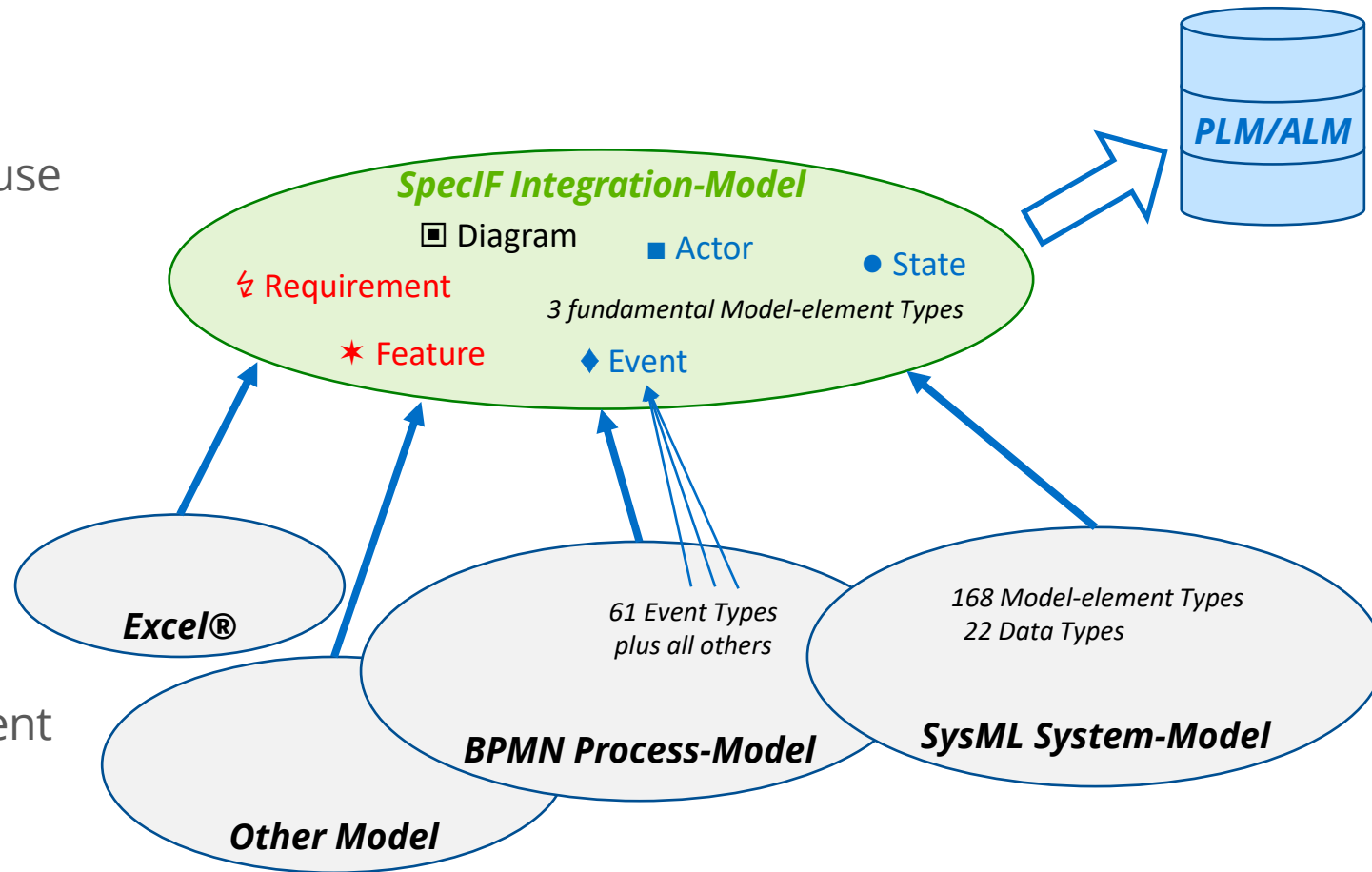
1. Separate View and Model
2. Abstract Model Element Types
3. Use a Vocabulary
4. Share Model Elements between Views
5. Interrelate Model Elements to build a semantic net

Find a Useful Abstraction Level

- **Model-Integration:** Cannot match original model-element types, because there are too many.
- **Configuration Management:** Should only handle a few artefact types.

→ Map to 3 fundamental model-element types ■ Actor, ● State and ◆ Event being common to *all* notations.

(see Fundamental Modelling Concepts by S.Wendt)



Use a Vocabulary

- Add meaning to terms
- Agree on terms and meaning
- Meaning is conveyed with the terms
- Can be translated to
 - national languages
 - special terminology in a given field

→ Use and contribute to the SpecIF System Engineering Vocabulary

Resource Class Names

- FMC:Actor
- IREB:Requirement

Statement Class Names

- IREB:satisfiedBy
- oslc_rm:refinedBy

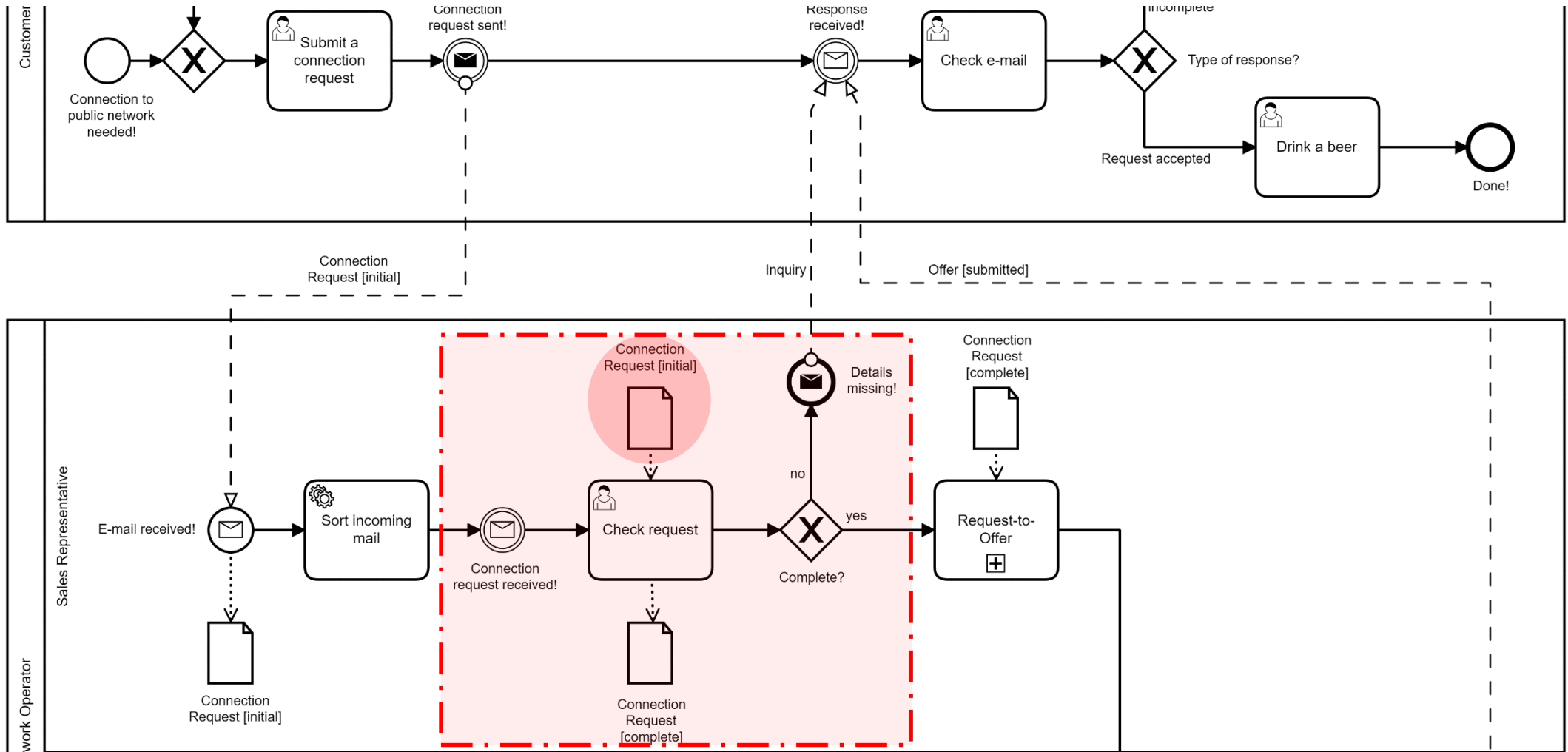
Property Names

- dcterms:title
- SpecIF:Priority

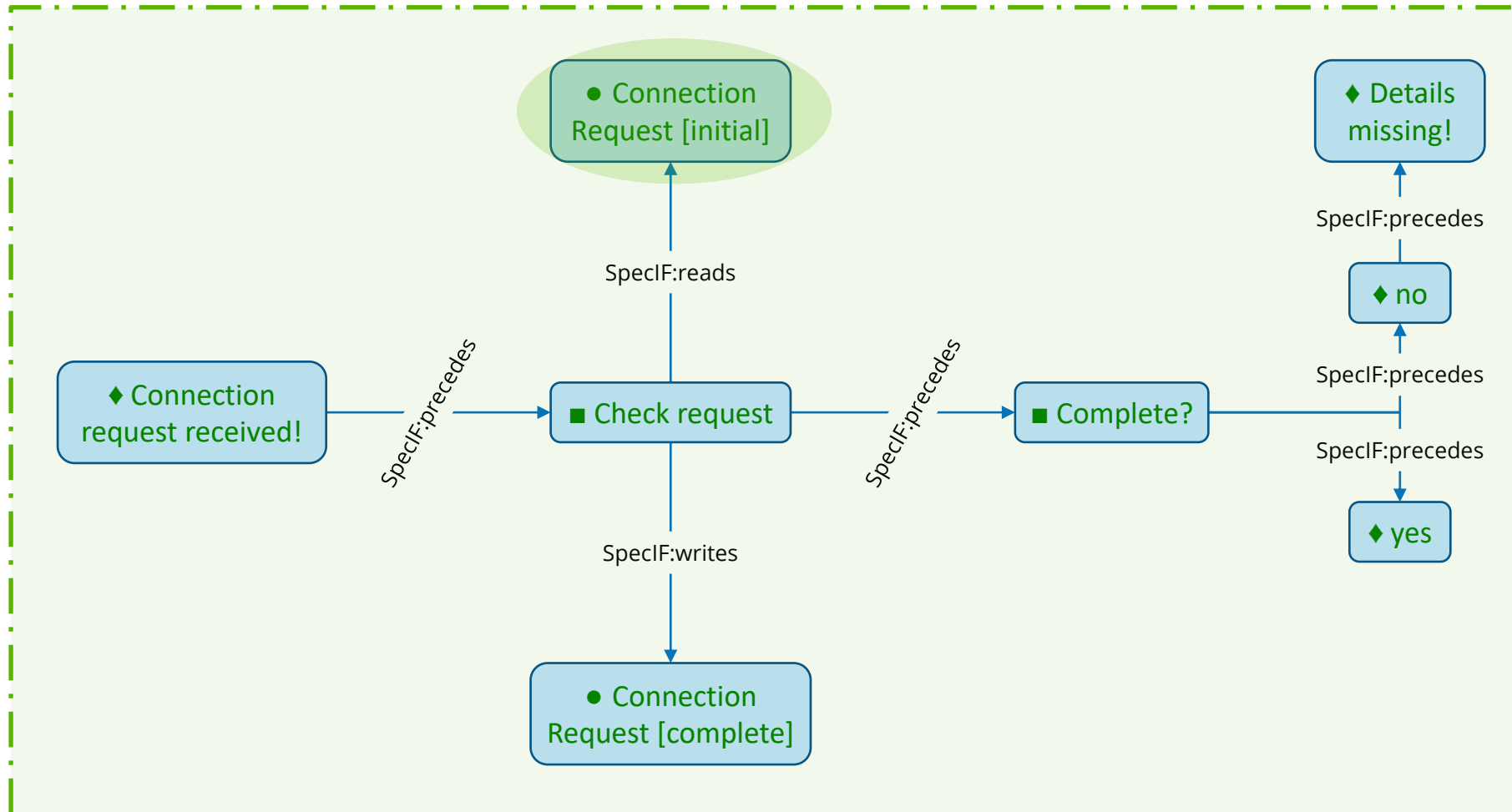
Property Values

- SpecIF:priorityHigh

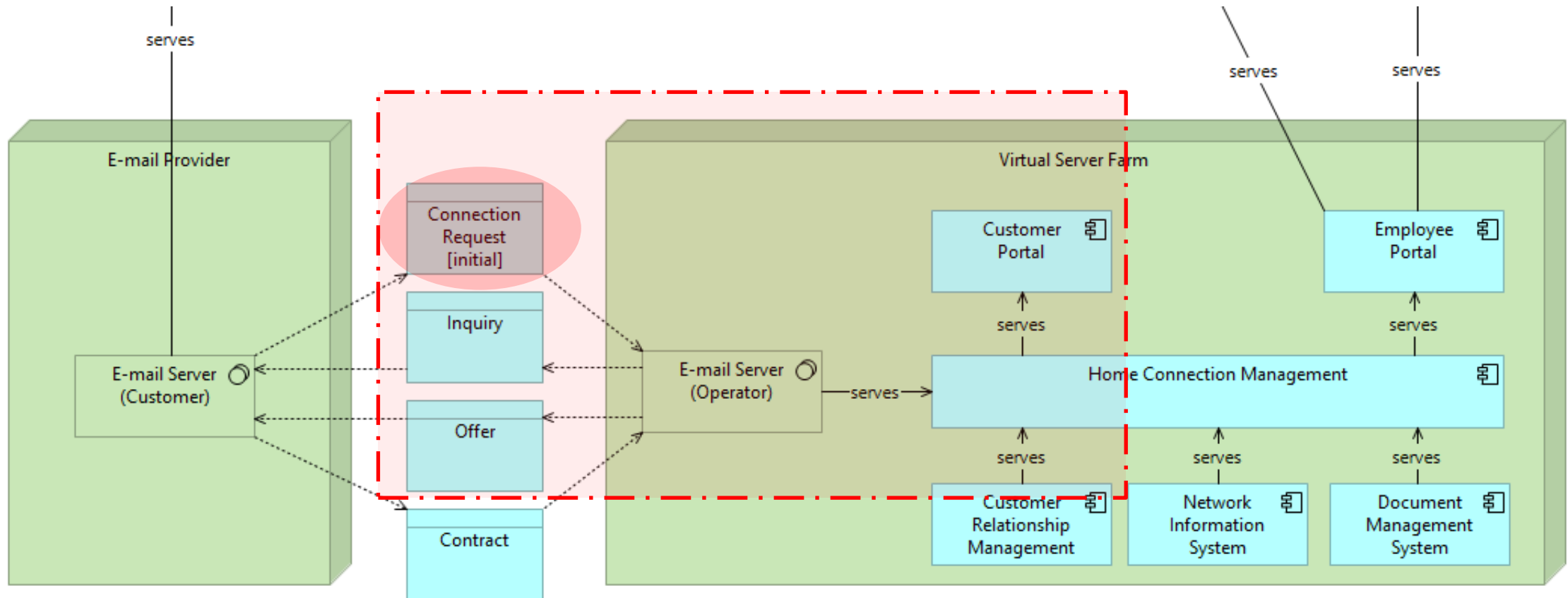
Telephone Connection Request – Business Process (BPMN)



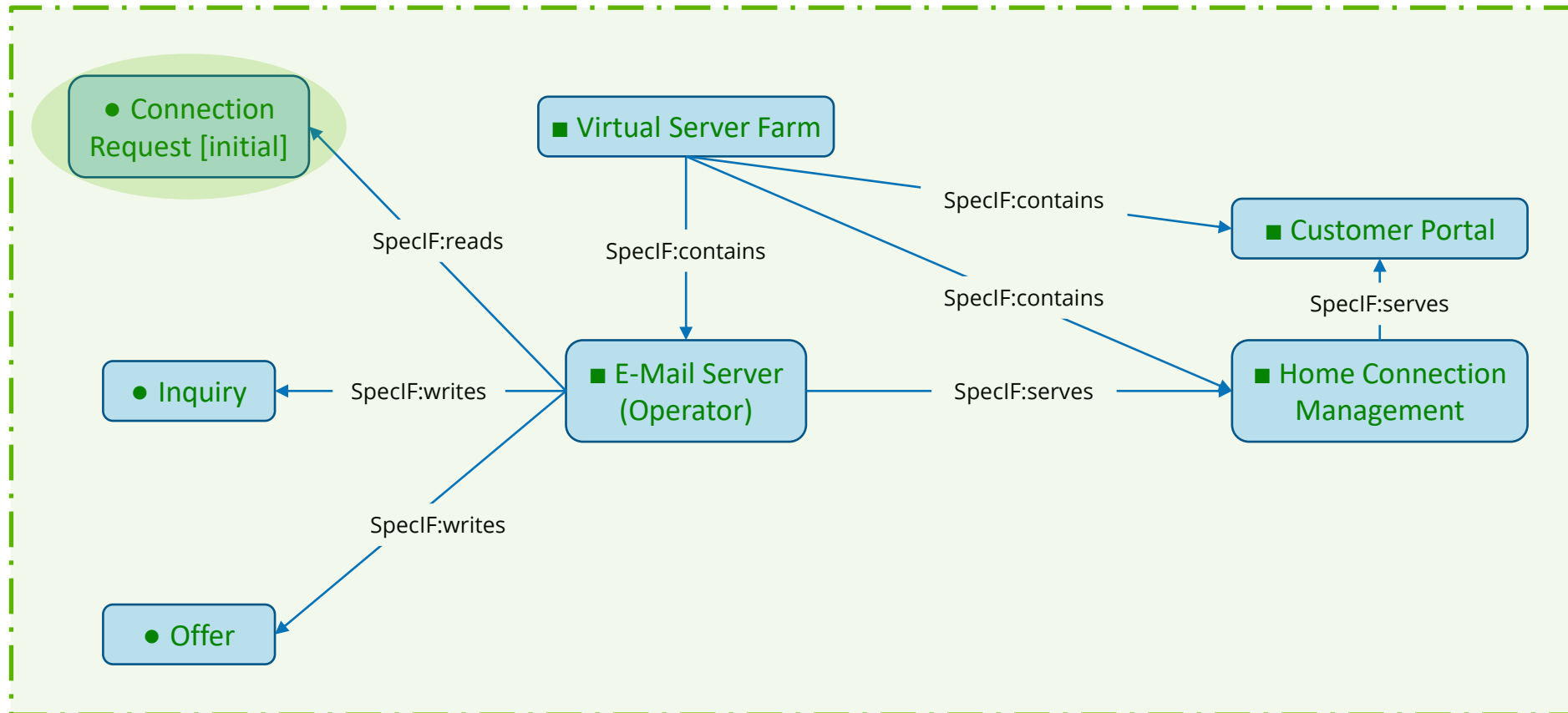
BPMN → SpecIF Transformation



Telephone Connection Request - Application Landscape (Archimate)

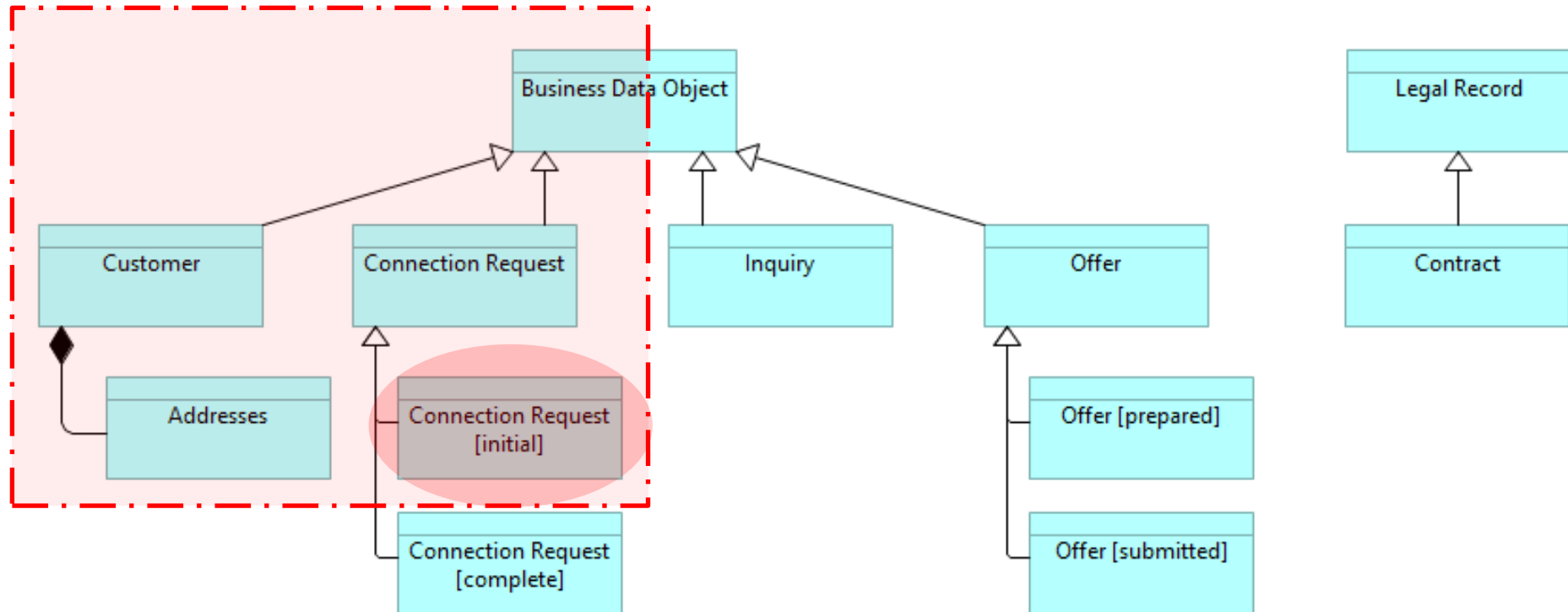


Archimate → SpecIF Transformation

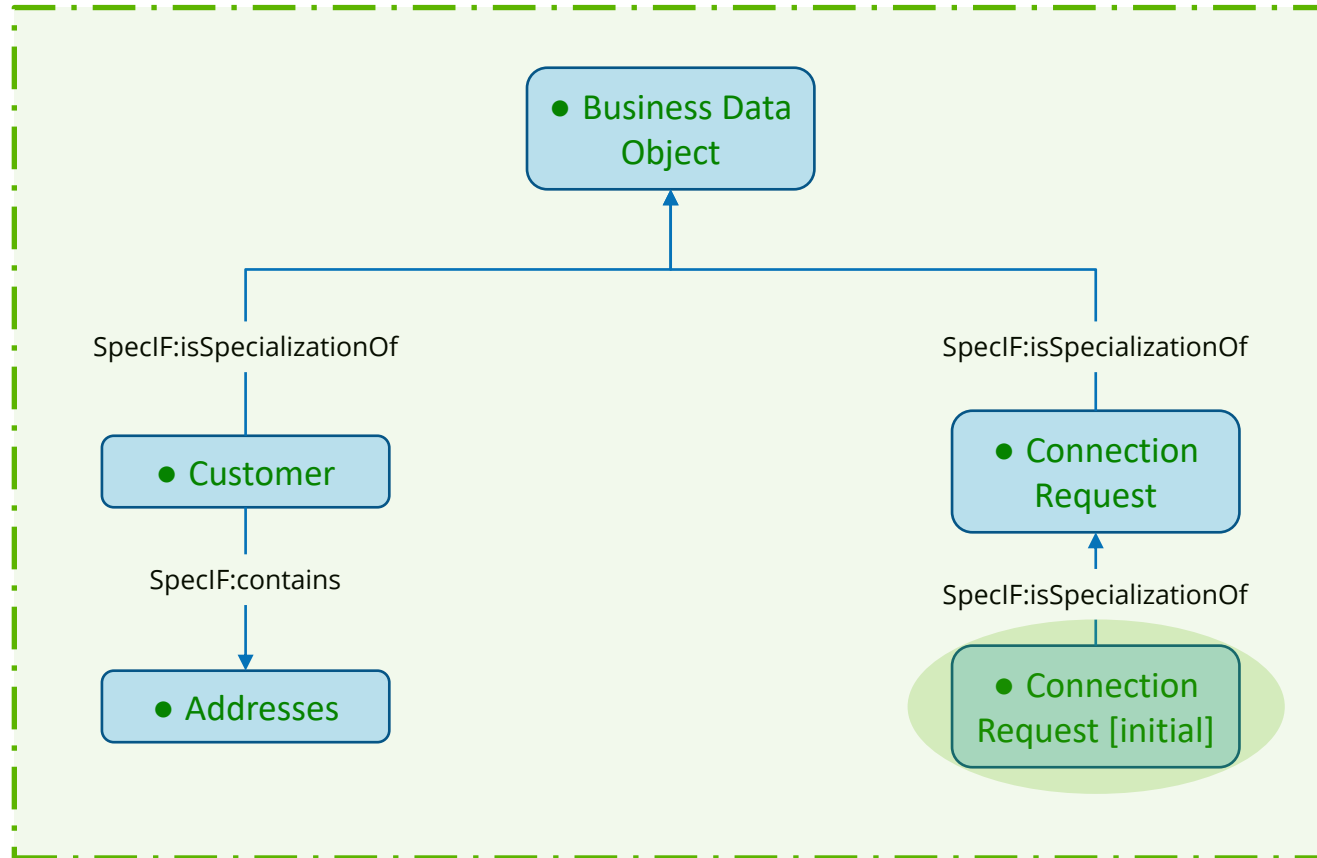


Telephone Connection Request

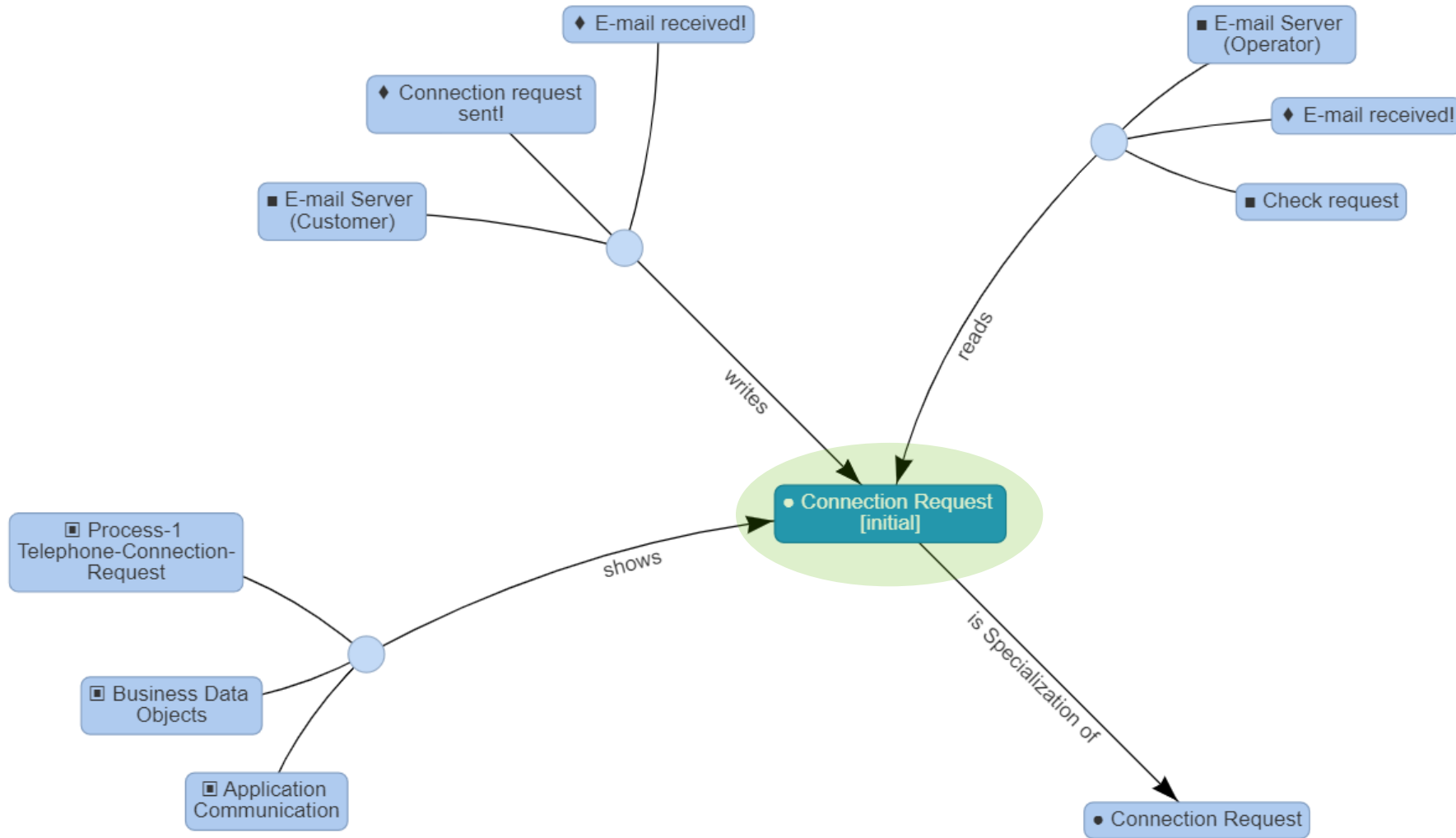
- Class Diagram (Archimate, UML, ..)



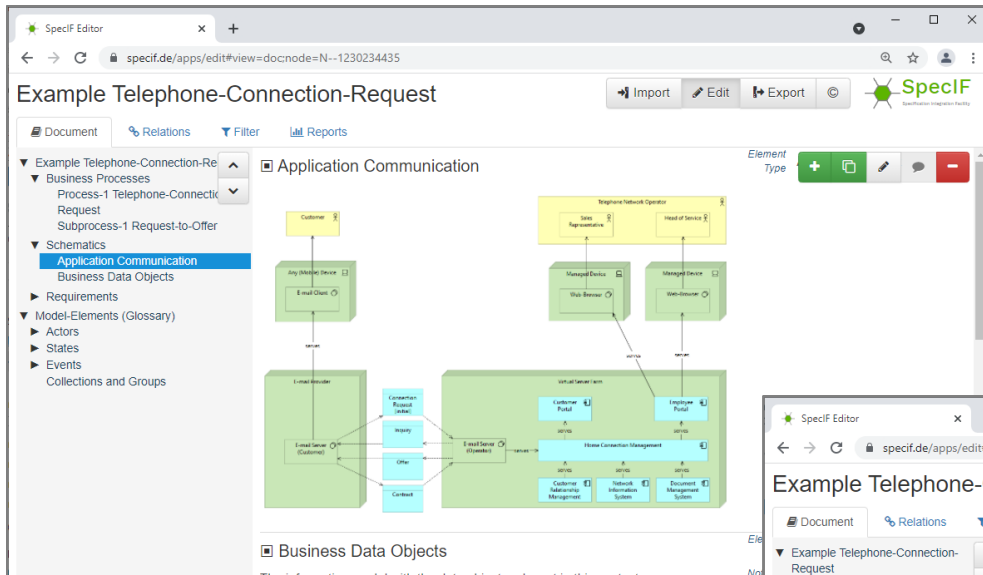
Class Diagram → SpecIF Transformation



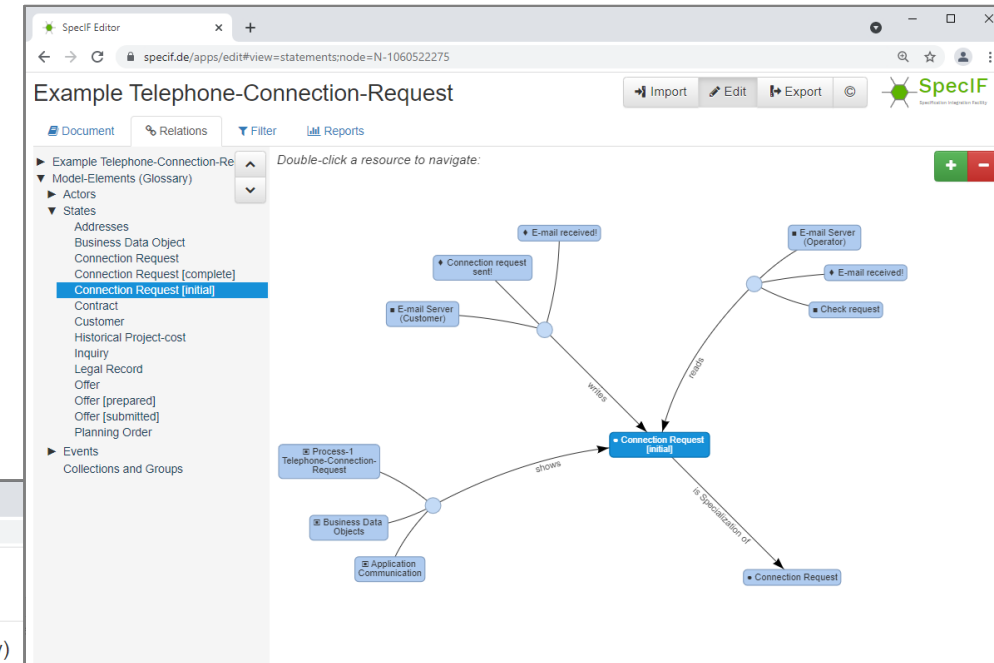
All relations combined → Semantics of a Model-Element



Navigate, search and audit in a common context



All Model-diagrams



All Relations

All Model-elements

This screenshot shows the SpecIF Editor interface with a glossary of model elements. The glossary lists various elements and their types, such as 'Any (Mobile) Device' (Archimate:Device), 'Approve offer' (bpmn:userTask), 'Check e-mail' (bpmn:userTask), 'Check request' (bpmn:userTask), and 'Create planning order' (bpmn:userTask). A blue hand icon points to the 'Check request' entry.

Element Name	Element Type
Any (Mobile) Device	Archimate:Device
Approve offer	bpmn:userTask
Check e-mail	bpmn:userTask
Check request	bpmn:userTask
Create planning order	bpmn:userTask

SpecIF Goals and Benefits

- Exchange model-based specifications between organizations and tools.
- Combine texts and models from different tools.
- Navigate, search and audit in a common context.
- Manage the product lifecycle from birth to death („end-to-end“):
 - Reference for all engineering-disciplines
 - Combining methods
 - Technology-neutral
 - Vendor-neutral
 - Schema-conforming
 - Standard-conforming
 - Open and cooperative

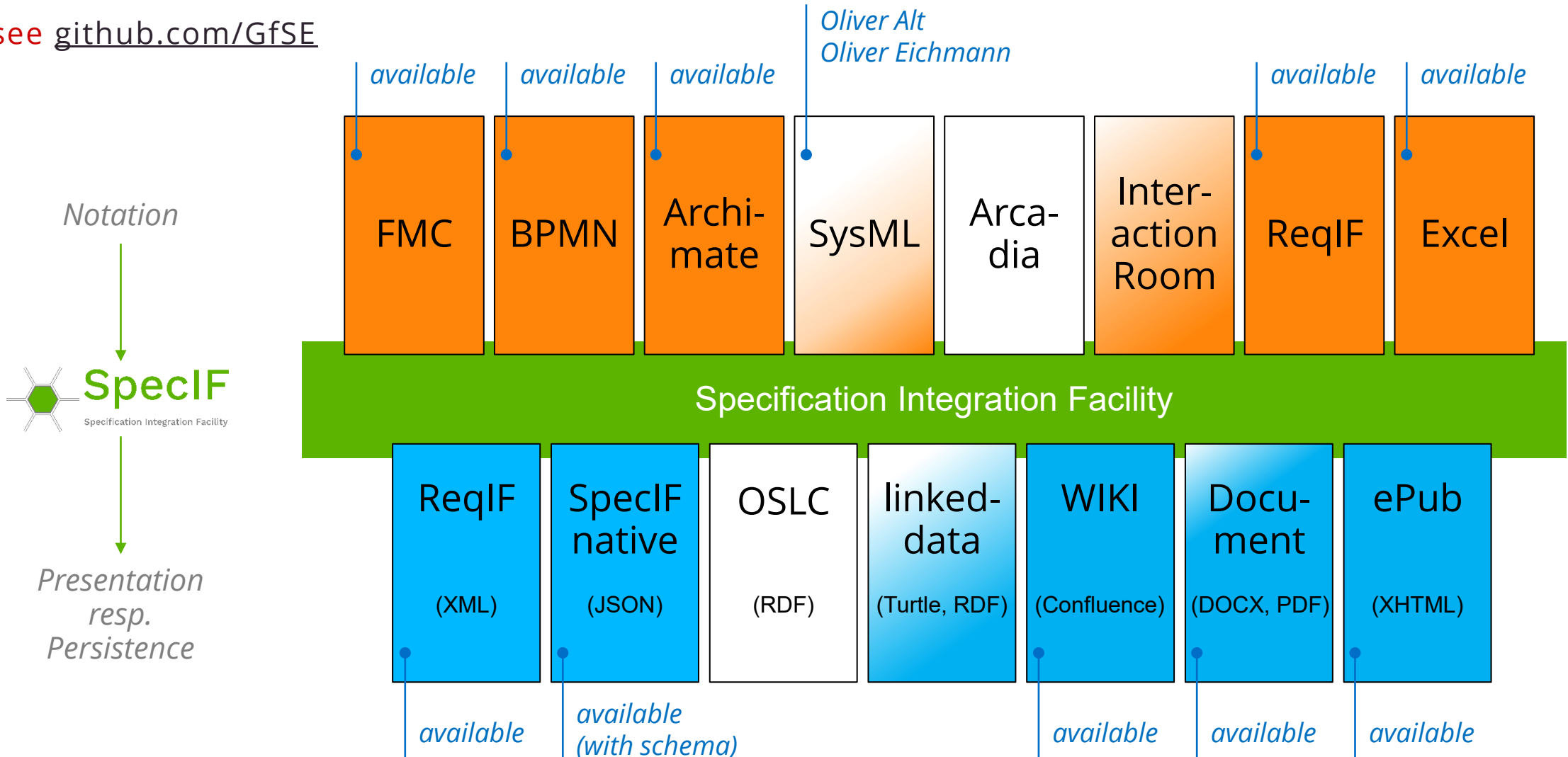


What is different ?

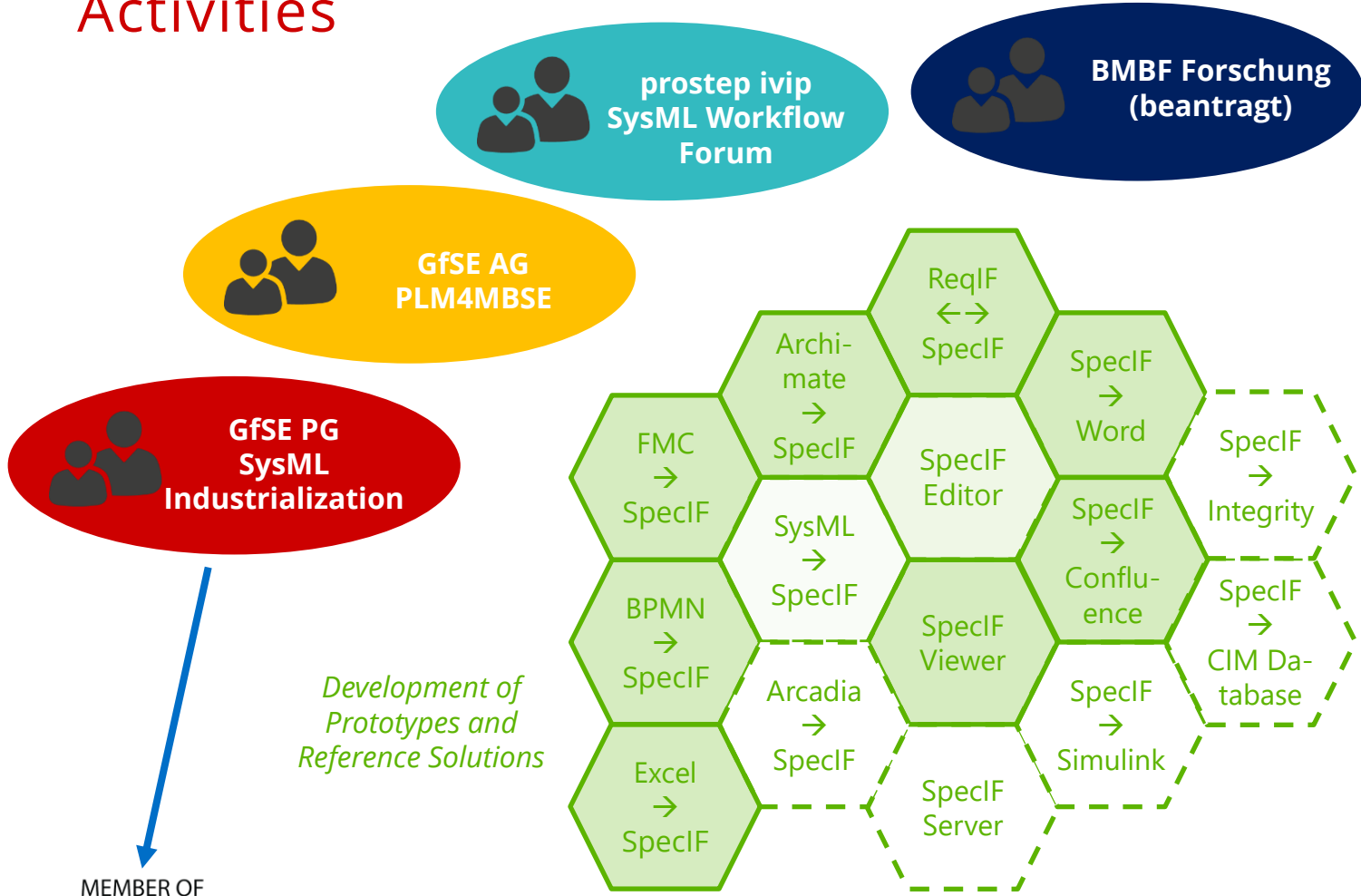
1. SpecIF builds on known notations and technical formats; doesn't replace any.
2. Conveys meaning through defined vocabulary and simple predicate logic.
3. SpecIF works because it uses *few* fundamental model-element types.
4. SpecIF is a semantic net („graph-data“) with typed nodes and edges.
5. Graph data is highly scalable – searching is ultra-fast.
6. Dynamic data-model – strict meta-model with schema and constraint checker.
7. Users drive open-source development – don't expect product vendors to invest; time-to-production 7 years → 1 year.

Status ...

see github.com/GfSE

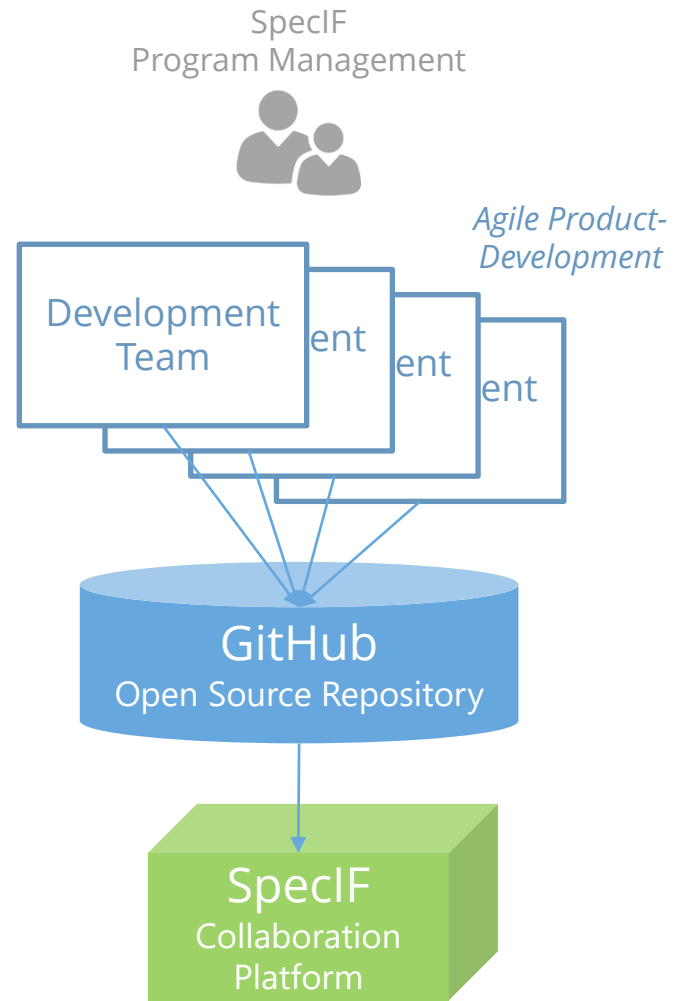


Activities



Development of Prototypes and Reference Solutions

Plan



Resources

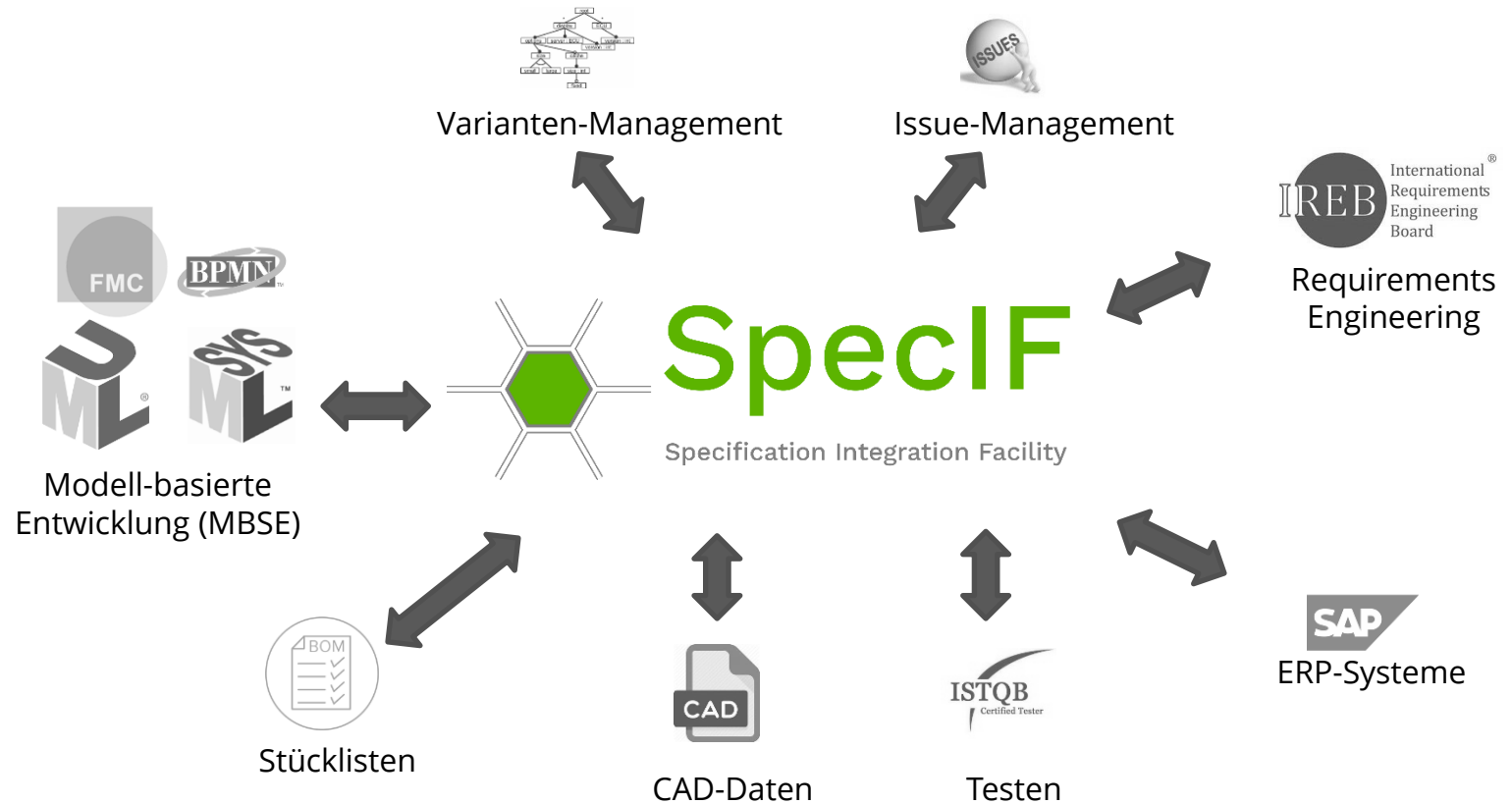
- [SpecIF Home](#)
- [SpecIF Schema and Tools on GitHub \(Open Source, Apache License\)](#)
→ Use the GitHub Ticket System for questions, proposals and requests
- Hosted SpecIF [Schema](#) and [Consistency-check](#) (CORS-enabled, watch for [new releases](#))
- SpecIF [Viewer](#) und [Editor](#)
- Examples:
 - [Telephone Connection Request](#) (Notation BPMN+Archimate+XSLX)
 - [System Engineering Collaboration](#) (Notation Archimate)
 - [Dimmer](#) (Notation FMC)
 - [Small Autonomous Vehicle](#) (Notation: SysML)
- SpecIF [Vocabulary](#)

Literature

- [1] Wendt, S.: Ein grundlegender Begriffsrahmen für das Wissensmanagement im Software-Engineering. In Proceedings „Knowtech“ Dresden 2001.
- [2] Knöpfel, A.; Gröne, B.; Tabeling, P.: Fundamental Modelling Concepts – Effective Communication of IT Systems. ISBN-13: 978-0-470-02710-3. John Wiley&Sons, Chichester, 2005.
- [3] Kaufmann, U., Pfenning, M.: 10 Theses about MBSE and PLM.
- [4] Object Management Group: Systems Modeling Language (OMG SysML™), Version 1.3, June 2012.
- [5] Object Management Group: Requirements Interchange Format (ReqIF).
- [6] Open Services for Lifecycle Collaboration (OSLC).
- [7] Specification Integration Facility (SpecIF).
- [8] Dungern, O.v.: Semantic Model-Integration for System Specification – Meaningful, Consistent and Viable, 7.Grazer Symposium Virtuelles Fahrzeug, Graz, Mai 2014.
- [9] Dungern, O.v.: Integration von Systemmodellen mit fünf fundamentalen Elementtypen. TdSE Tag des Systems Engineering der GfSE, Ulm, November 2015.
- [10] Dungern, O.v.: Von Anforderungslisten zu vernetzten Produktmodellen – am Beispiel der Gebäudeautomation. REConf, Unterschleißheim, März 2016.
- [11] Dungern, O.v.: Semantic Model Integration for System Specification. TdSE Tag des Systems Engineering der GfSE, Herzogenaurach, October 2016.
- [12] Uphoff, F.: Konzept und prototypische Implementierung der Modellintegration der Interaction-Room-Methode in die Specification Integration Facility, Kamp-Lintfort, März 2017.
- [13] Mochine, P.; Sünnetcioglu, A.; Dungern, O.v.; Stark, R.: SysML-Modelle maschinell verstehen und verknüpfen. TdSE Tag des Systems Engineering der GfSE, Paderborn, Oktober 2017.
- [14] Alt, O.: SpecIF - Die kommende vielschichtige Datenquelle für Spezifikationsdaten. Fachgruppentreffen GI-RE, Nürnberg, November 2018.
- [15] Dungern, O.v.: Model-Integration with SpecIF. ProSTEP ivip e.V. SysML-Workflow-Forum November 2019.

Interessant ?

od@enso-managers.de
oliver.alt@mdd4all.de



Details for further discussion

What is SpecIF ?

- „**Spec**ification **I**ntegration **F**acility“, a GfSE initiative



- SpecIF adds **conventions to convey meaning** using known technical formats such as ReqIF or OSLC.

1. Vocabulary for Objects,
Relations and Attributes

„Requirement“, „Actor“, „State“, „Event“ ...
„satisfies“, „reads“, „contains“, „triggers“, ...

2. Logic Assertions
(„First-order predicate logic“)

„A Component *satisfies* a Requirement“
„An Event *triggers* an Activity“

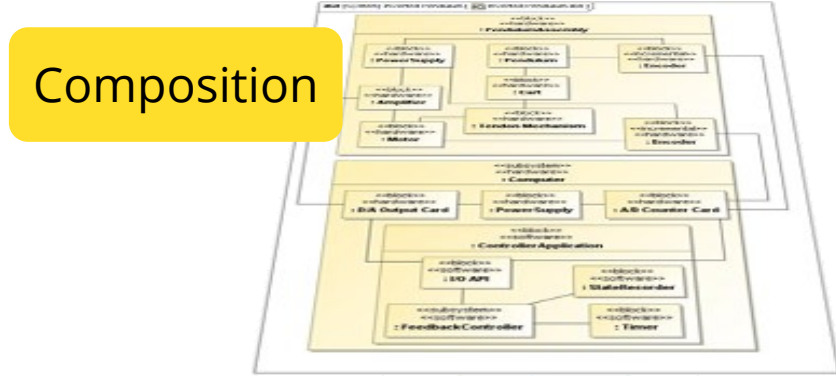
→ SpecIF carries both the „Visible“ and the „Meaning“

Purpose: Model Integration

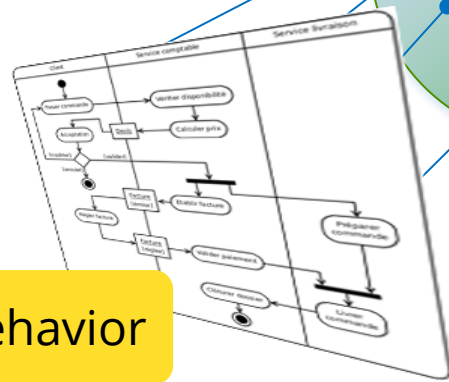
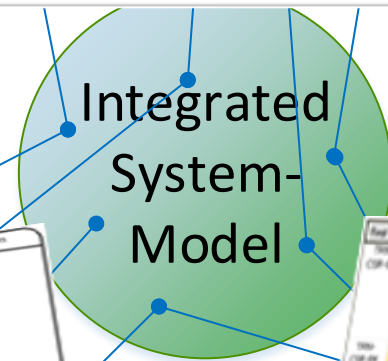
- Search, navigate and audit partial engineering results in a common context
- Interrelate information elements of disparate sources to create a common view
- Find inconsistencies and gaps between different views
- Use-Cases:
 - Localize requirements on a BoM: Which components are affected when changing a requirement (or vice versa)?
 - When a use-case mentions a data-object, which system components and interfaces are involved?
 - Interrelate system structures with process models: Which activities are affected when modifying a system component (or vice versa)?
 - Associate a FMI simulation-routine with a SysML system component (block)
 - Collect and compare information about an element from different sources.

Consolidate model elements from different diagrams

- Key to success is the abstraction using 5 fundamental model element types
- Impossible with 162 model element types in SysML and almost as many in BPMN
- Even within SysML the current tools fail to properly consolidate model elements from different model views



Composition



Behavior

Req	Use case requirements	Priority	Acceptance	Class
CP-03	The user shall be able to receive a warning when service is due.	High	Accept	(3) what color indication as warning for the warning when the user gets a requirement in plan to activate the order
CP-04	3.1.17 Indication requirements	High	Accept	
CP-05	The user shall be able to use at all times of activation of speed to within 1 or 2h.	2	Accept	
CP-06	The user shall be able to use at all times of activation of engine resolution to within 1 or 2h.	2	Accept	
CP-07	The user shall be able to obtain direction to go.	2	Accept	

Requirements

What's Needed to Integrate System-Models with Requirements







- Import FMC (ARCWAY Cockpit) available
- Import SysML → O. Alt (EA), O. Eichmann (Cameo)
- Import BPMN-XML available
- Import Archimate/Open-Exchange model-elements and relations: yes, diagrams manually
- Import ReqIF
- Import Arcadia (Capella)
- Import XLSX, XLS, CSV available
- Model-Integration per „Adopt“ available
- SpecIF Server → O. Alt (.Net Core), O. v. Dungern (Node.js)
- Export ReqIF available
- Export ePub available
- Export OOXML (MS Word) available (early version)

All model diagram types (notations) consist of three fundamental model-element types*

Model-element Type	Diagram Type (Notation)				
	BPMN Business-process	State-machine	System-composition	Organisation Chart	UML Classes
■ Actor	X	(X)	X	X	
● State	X	X	X		X
◆ Event	X	X			

* Prof. Dr. Siegfried Wendt, Founding Director of the Hasso-Plattner-Institute, Potsdam: Fundamental Modelling Concepts

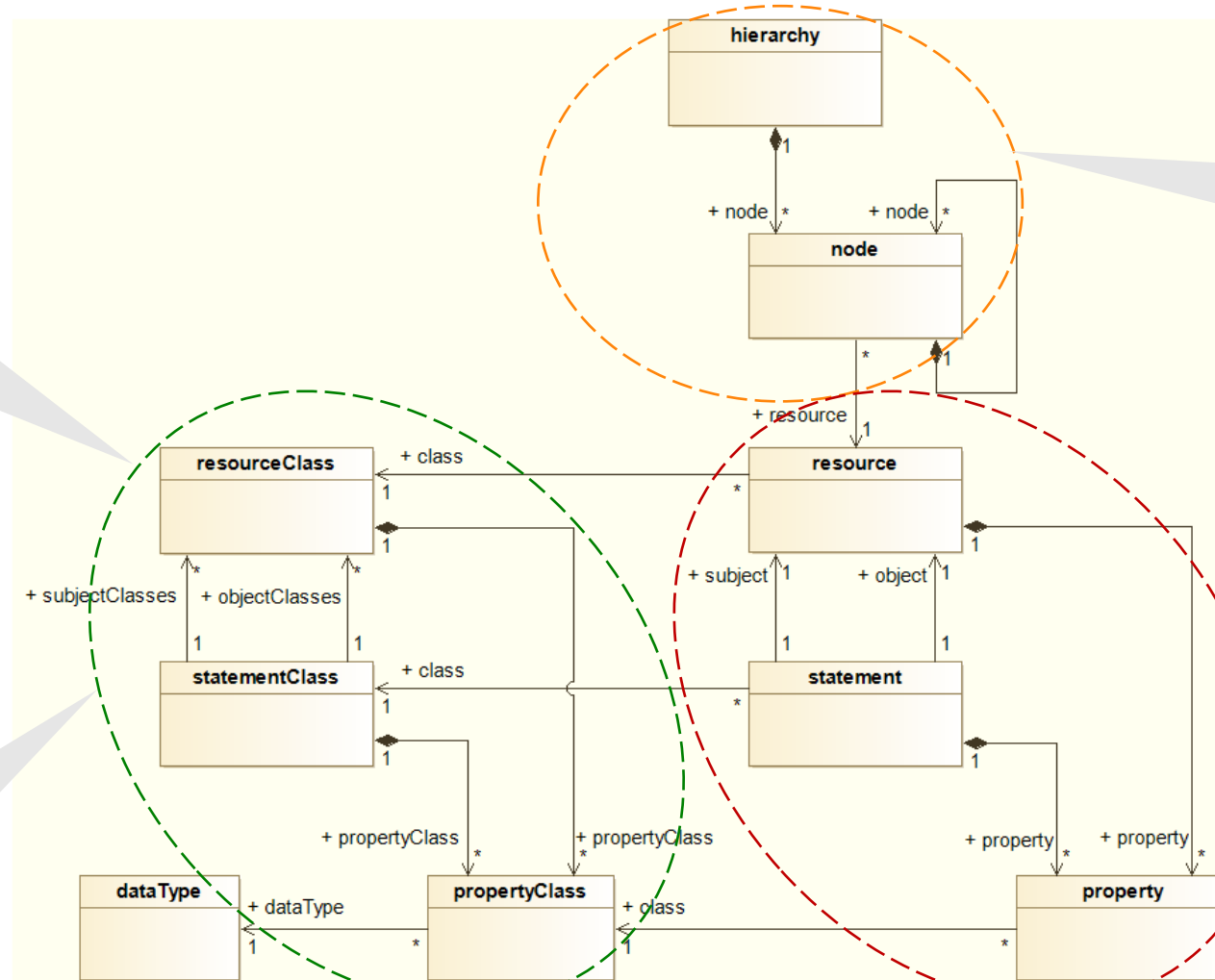
A complete specification needs also ,feature' and ,requirement'

Model-element Type	 Diagram Type (Notation)					
	BPMN Business- process	State- machine	System- composition	Organi- sation Chart	UML Classes	Document Outline
 Actor	X	(X)	X	X		X
 State	X	X	X		X	X
 Event	X	X				X
 Feature						X
 Requirement						X

A SpecIF data set contains both the Set types („model“) and the instances („data“ = „payload“)

SpecIF model with Resource- and Statement-Classes

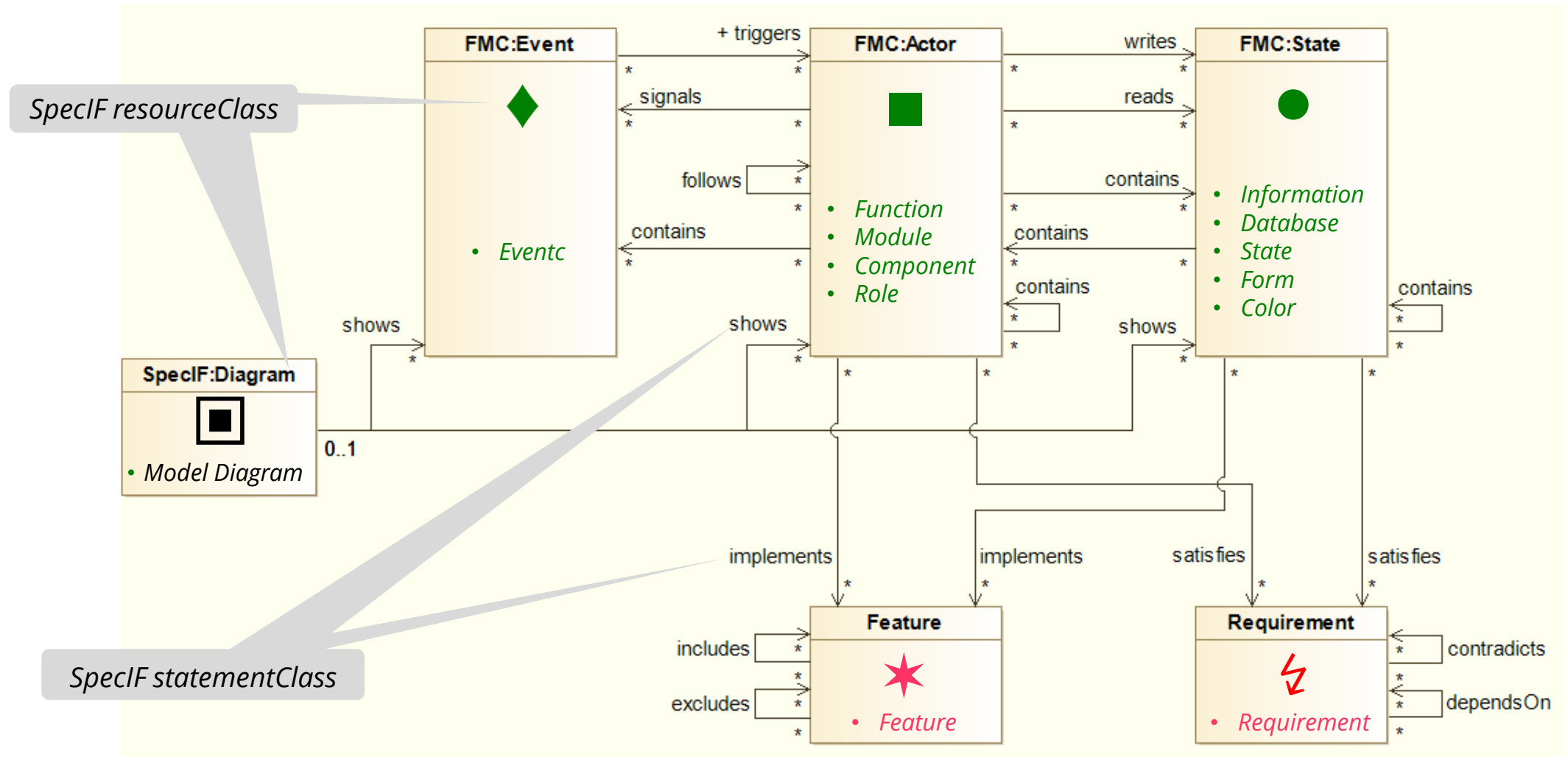
The classes can be defined at runtime („dynamic model“)



Hierarchical ordering of Resources (e.g. for convenient reading)

SpecIF data (payload) with Resources and Statements

The SpecIF Integration Model with 5 Fundamental Model-element Types



Simple Model-Integration „Adopt“

- Adopt existing model-element types, if they are equivalent
- Add new diagrams (views)
- Adopt existing model-elements having the same title
- Add new statements
- Build new glossary of model-elements, sorted by fundamental type