Model-Integration with SpecIF

2019-11-20

Oskar von Dungern, Dr.-Ing., adesso AG

Topics Today

1. Basic Assumptions
2. What is SpecIF
3. Model Integration
4. SpecIF Goals and Benefits
5. Resources and Examples
6. Status

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
- There is an interest
  - to search, navigate and audit partial results in a *common context*
  - to exchange model information between organizations and tools

→ That‘s where SpecIF kicks in.

What is SpecIF ?

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

SpecIF
Open SE Models

- 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“)

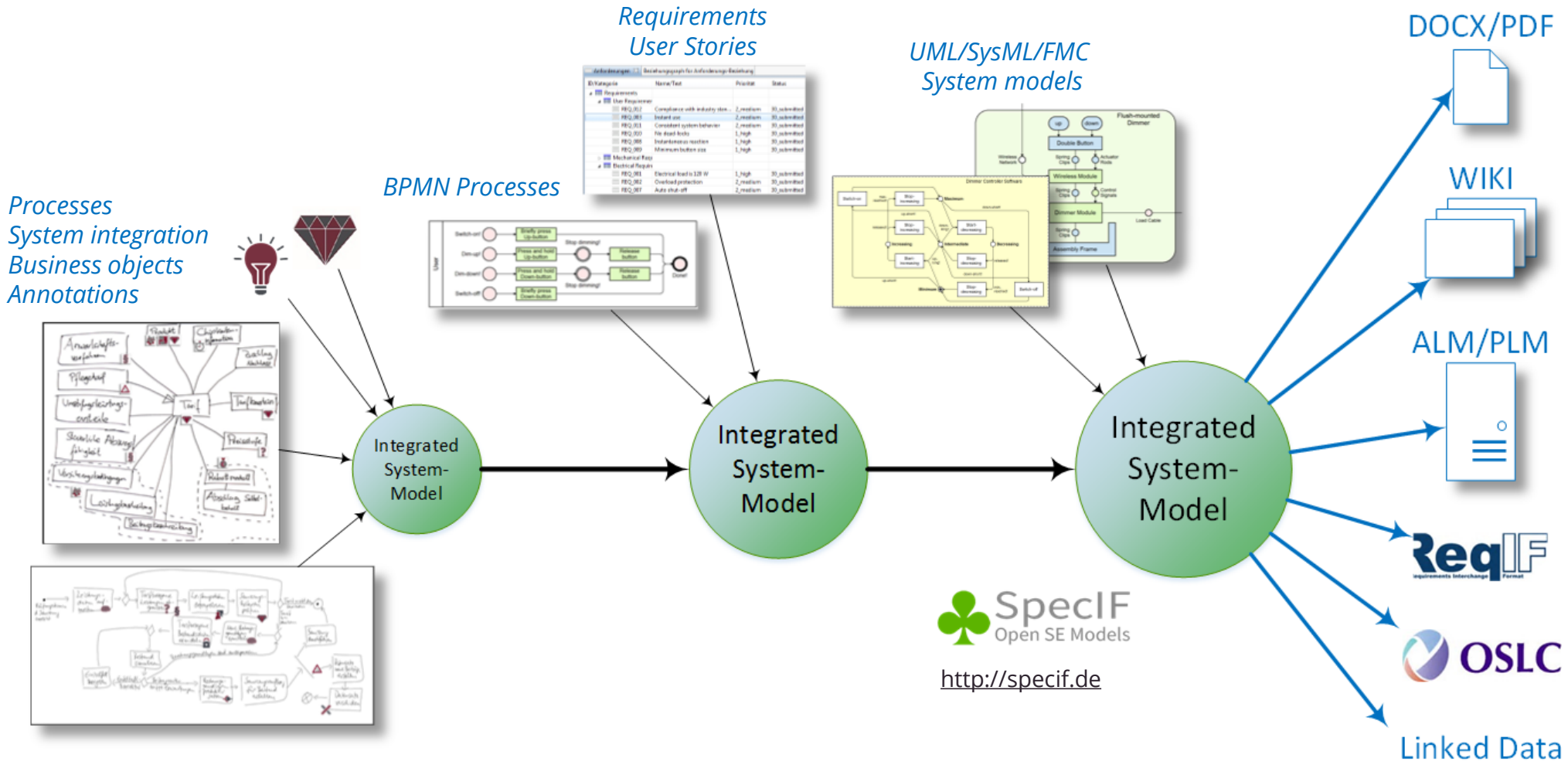
   „A Component *satisfies* a Requirement“
   „An Event *triggers* an Activity“

→ 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.

Add partial models step-by-step …



Five Principles of Model Integration

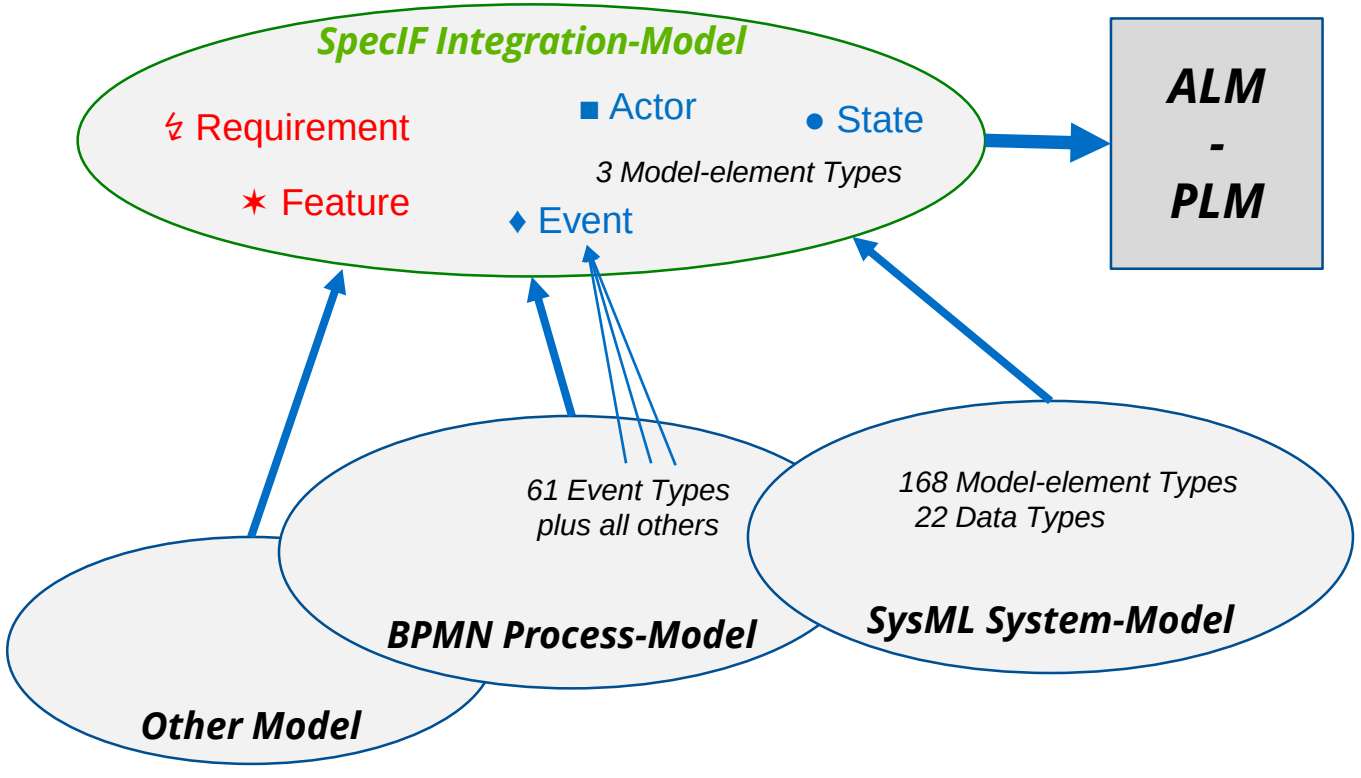
1. Separate View and Model
2. Abstract Model Element Types
3. Use a Vocabulary
4. Share Model Elements between Views
5. Interrelate Model Elements to build a semantic net

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:FitCriteria

Property Values

- SpecIF:priorityHigh

Use a Vocabulary

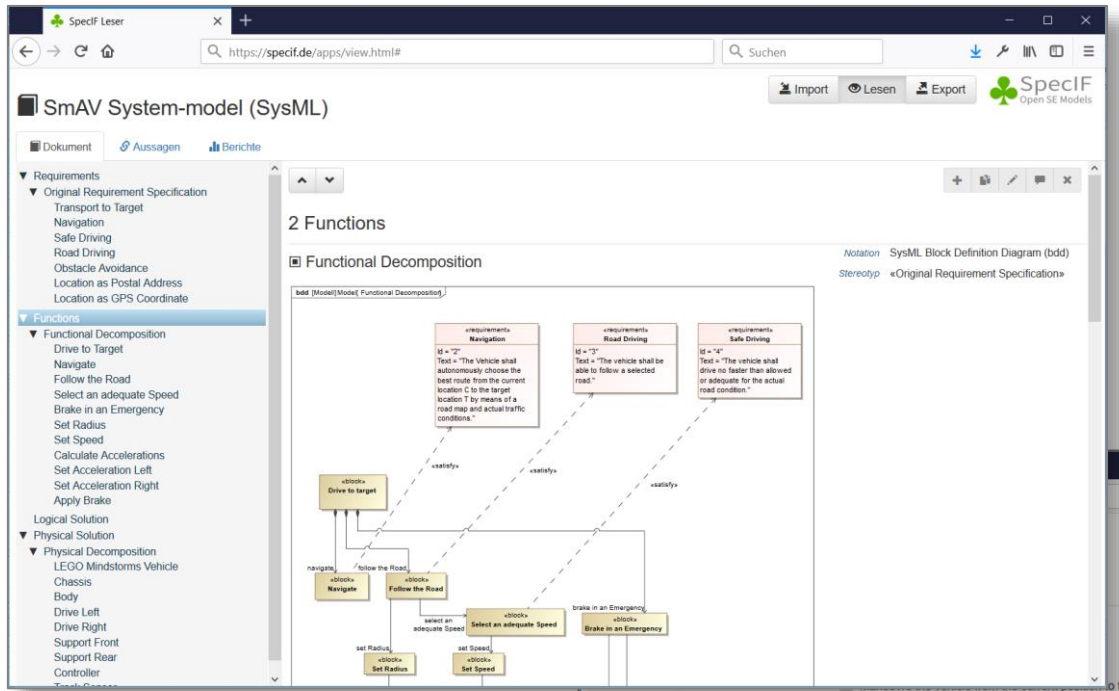
Example: Requirements in a spread-sheet

	B	C	D	E	F	G	H	I
1	**dcterms:title**	**dcterms:description**	**SpecIF:Status**	**SpecIF: Priority**	**Requestor**	**SpecIF:FitCriteria**	**IREB:refinedBy**	**oslc_rm:satisfiedBy**
2	Benefits for the author	- Volle Prozesskontrolle und Nachvollziehbarkeit: kein Datenaustausch, kein e-mail - Kein Informationsverlust durch unterschiedliche Datenmodelle verschiedener Tools - Daten werden nicht verteilt, „disclosure“ wird erschwert - Standardisierung durch HIS Austauschprozess und ReqIF Datenformat - Einfachere Konsolidierung von Lieferanten-Feedback	30_submitted	1_high	oskar.dungern@adesso.de			"Acceleration Command"
3	Benefits for the reviewer	- Kein Datenaustausch: kein Aufwand und keine Probleme - Kosten: Keine Lizenzen und keine Schulung für ein Anforderungs-Management-System, um an einer Lastenheft-Abstimmung teilnehmen zu können - Nachvollziehbarkeit durch zentrale Datenhaltung; kein e-mail	30_submitted	1_high	oskar.dungern@adesso.de		"Response time is < 0.5 sec"	"Acceleration Command"
4	Dialog layout and flow are process-o	The layout of the dialogs as well as the flow of dialogs follow the principal use cases. They are well organized and easy to understand.	60_completed	1_high	oskar.dungern@adesso.de	70% of new users (knowing the nature of his task) are able to use the application without studying a manual or having a training.	nonsense	
5								
6								

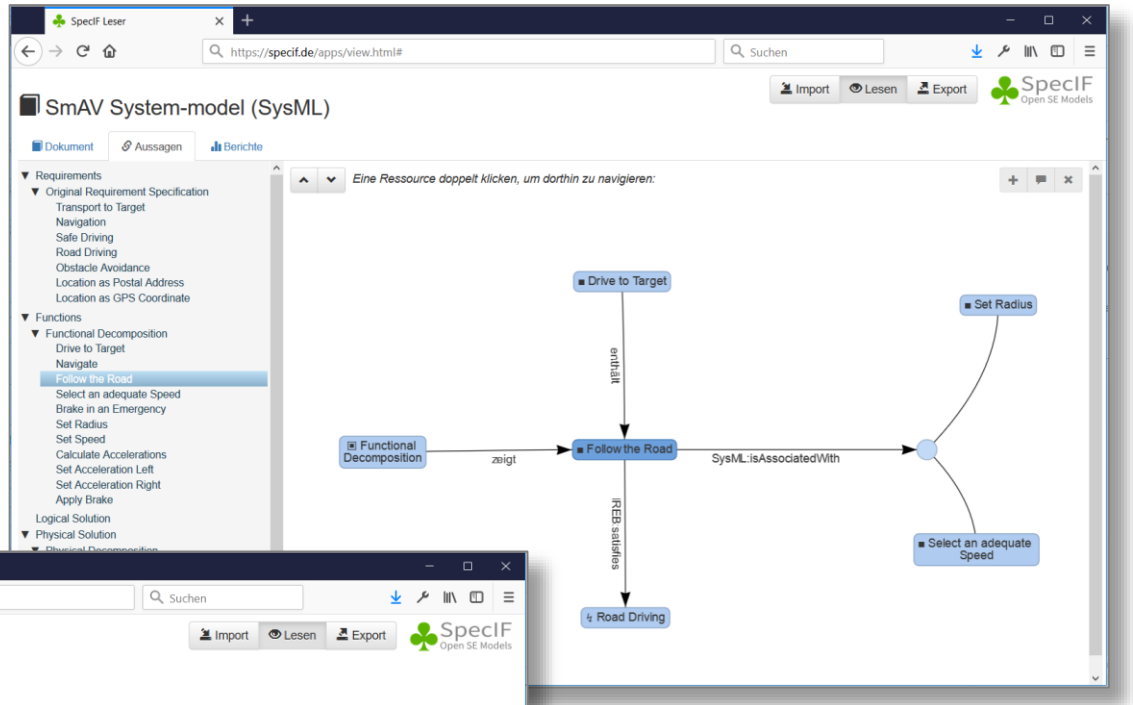
Simple Model-Integration „Adopt“

- 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

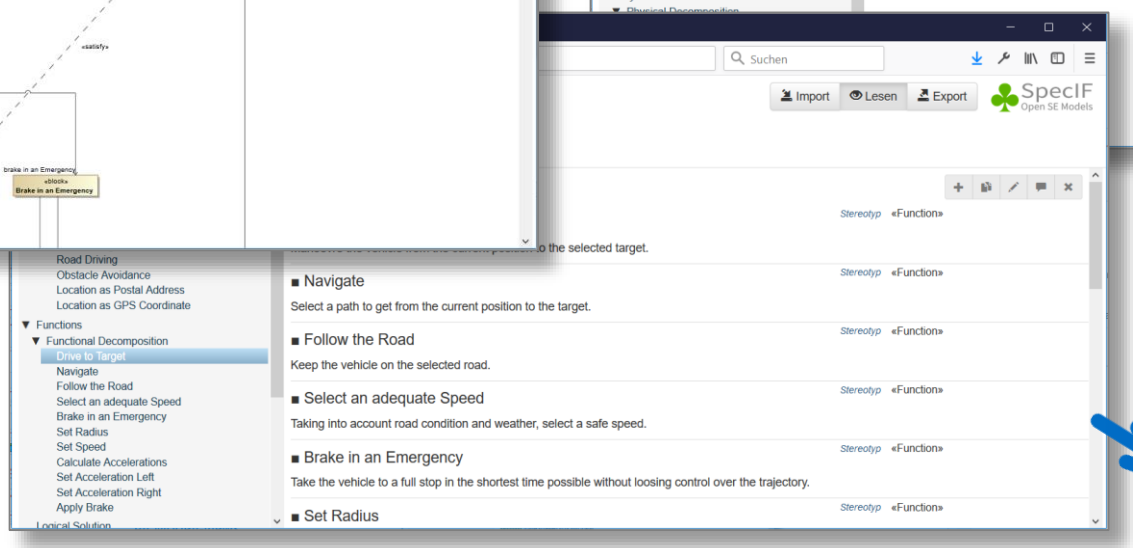
Example: Search, navigate and audit in a common context



All Modell-views



All Relations

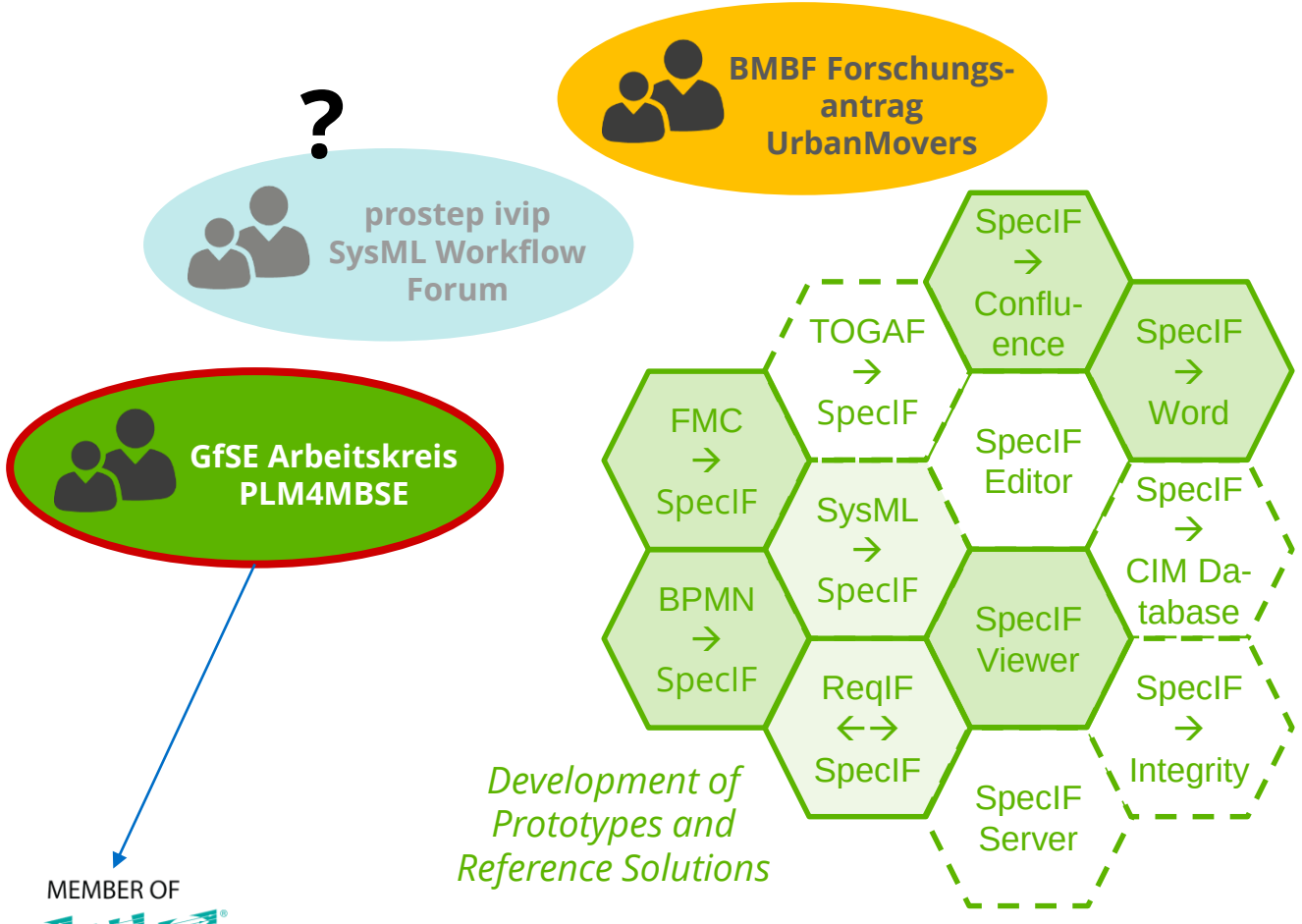


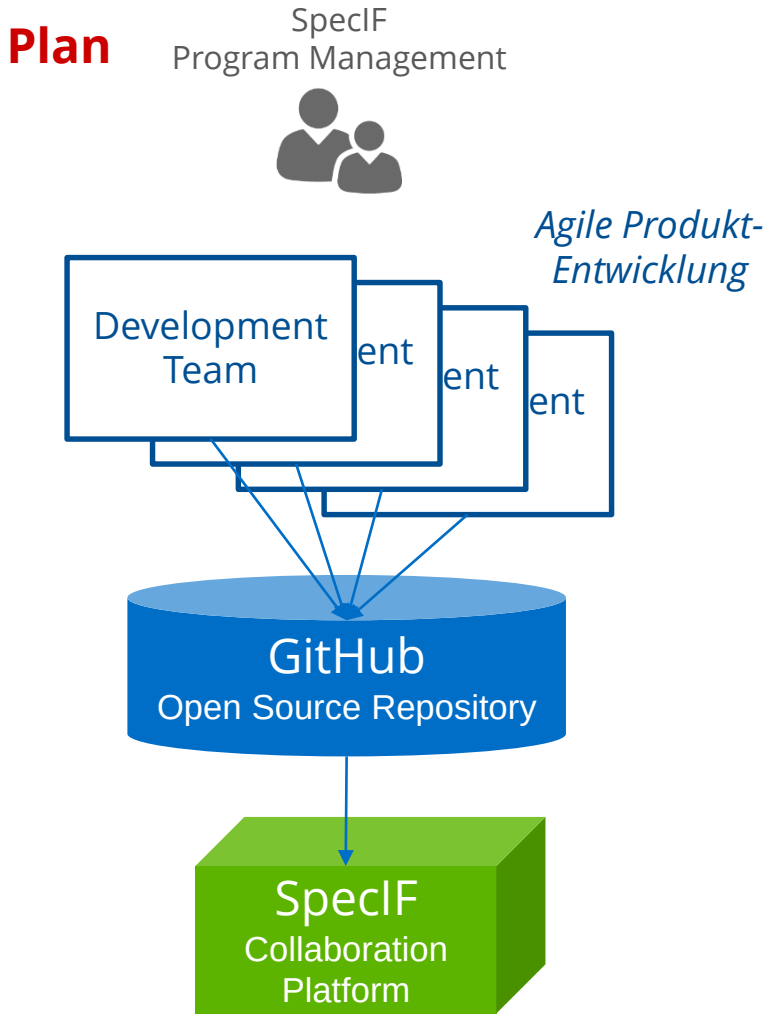
All Model-elements

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 ReqIF	
• Import XLSX	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)

Who's behind ?





MEMBER OF
OMG
OBJECT MANAGEMENT GROUP®
WE SET THE STANDARD®
www.omg.org

Who's behind ?

SCHAEFFLER
DAIMLER
VPE

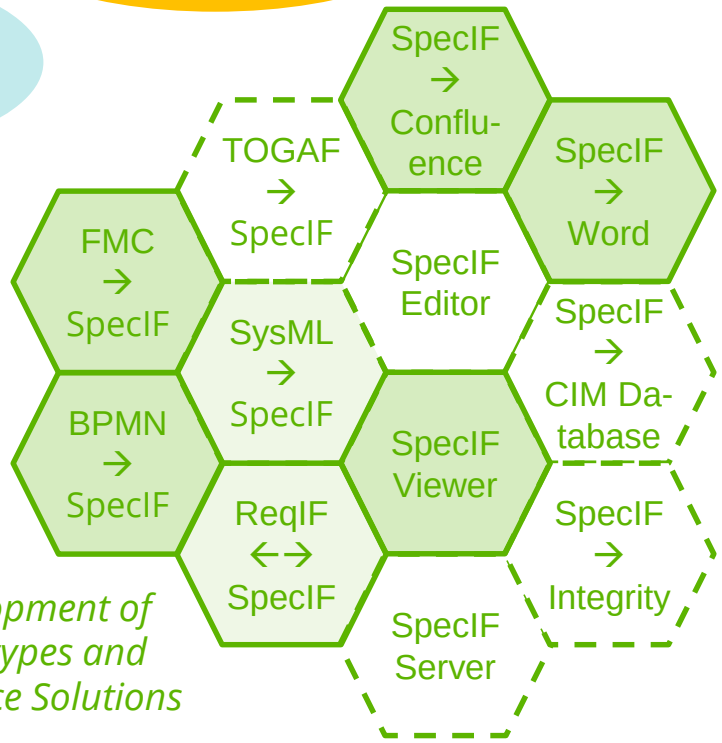
BMBF Forschungs-antrag UrbanMovers

prostep ivip SysML Workflow Forum

ZF

GfSE Arbeitskreis PLM4MBSE

MEMBER OF
OMG
OBJECT MANAGEMENT GROUP®
WE SET THE STANDARD®
www.omg.org



Development of Prototypes and Reference Solutions

adesso business. people. technology.

MDD4All.de

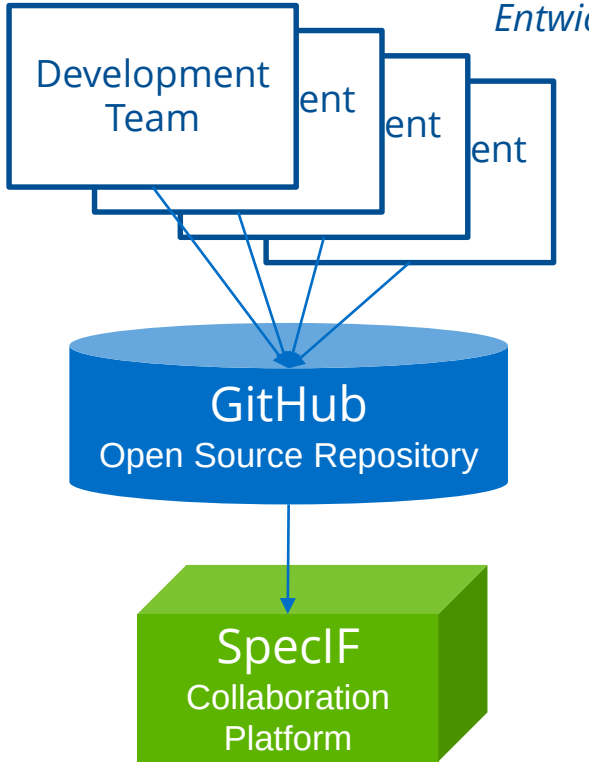
Fraunhofer IPK

FKS Institut für Flugzeug-Kabinensysteme

Plan

SpecIF Program Management

Agile Produkt-Entwicklung



SpecIF Goals and Benefits

- Exchange model-based specifications between organizations and tools.
- Combine texts and models from different tools.
- Search, navigate, 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

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
- Examples:
  - Fahrtbeginn (BPMN)
  - Dimmer (FMC)
  - SmAV (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.

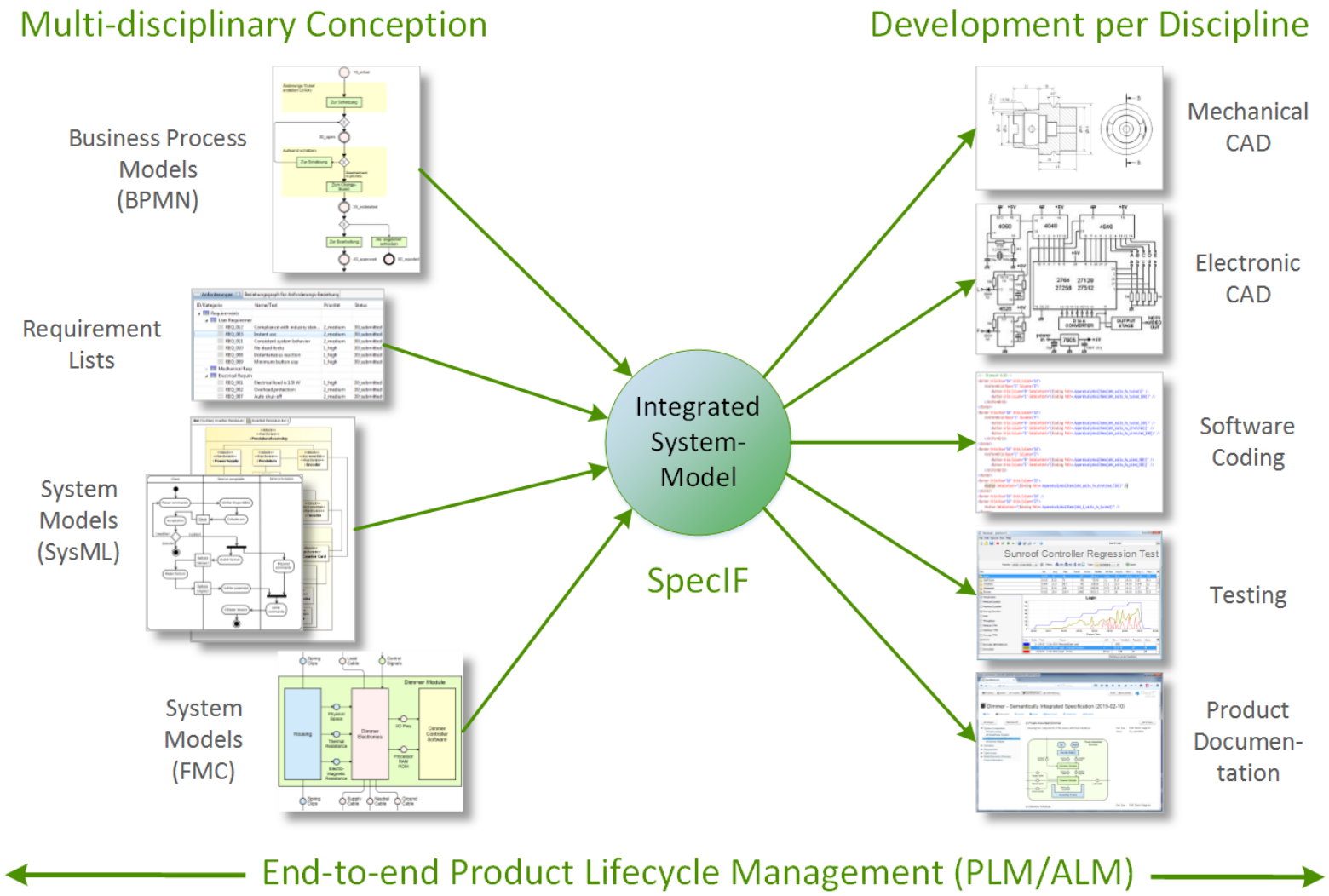
Interesting ?

oskar.dungern@adesso.de

adesso AG
Adessoplatz 1
44269 Dortmund
T +49 231 7000-7000
F +49 231 7000-1000
www.adesso.de

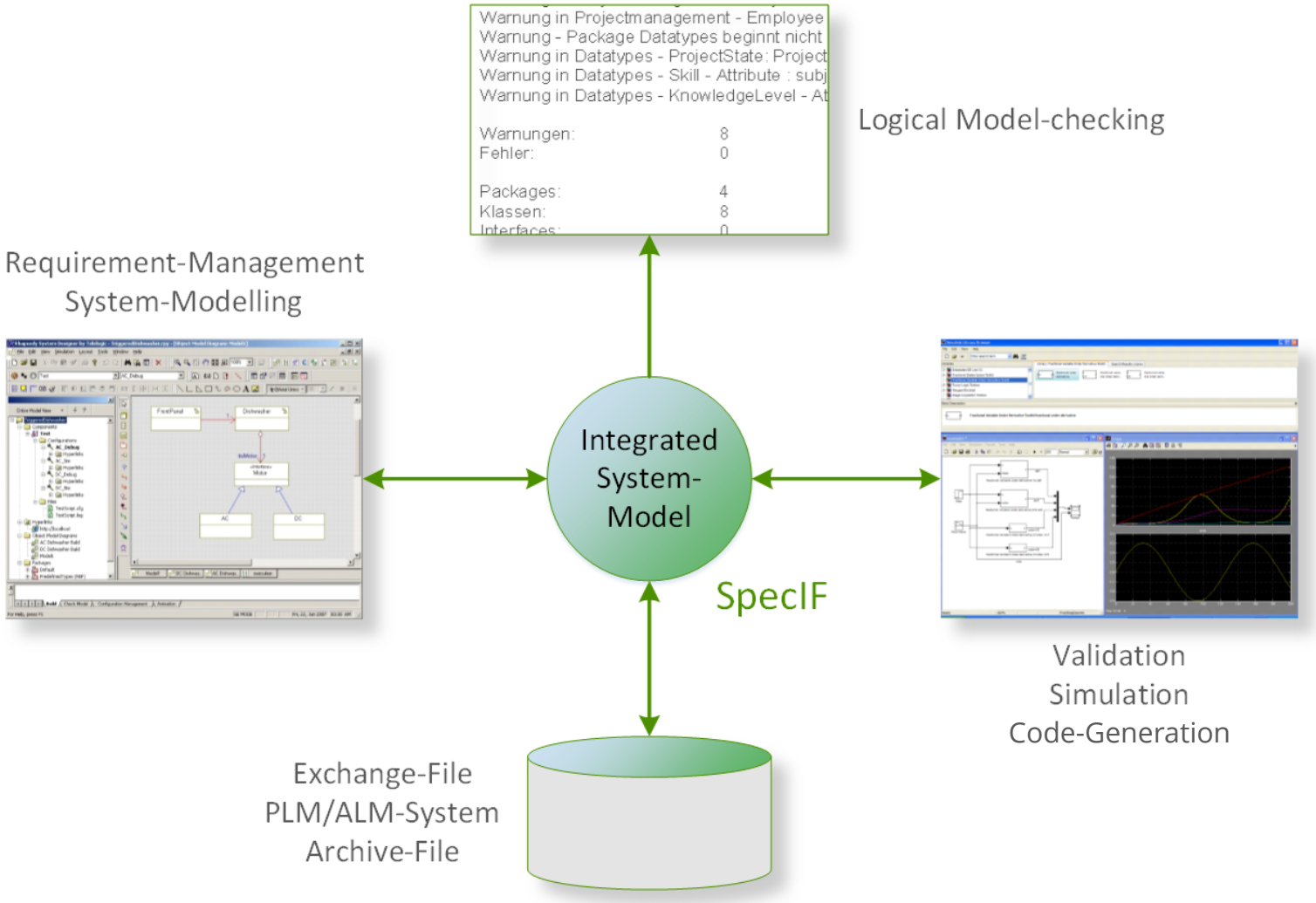
Details for further discussion

Purpose: SpecIF integrates information from different sources

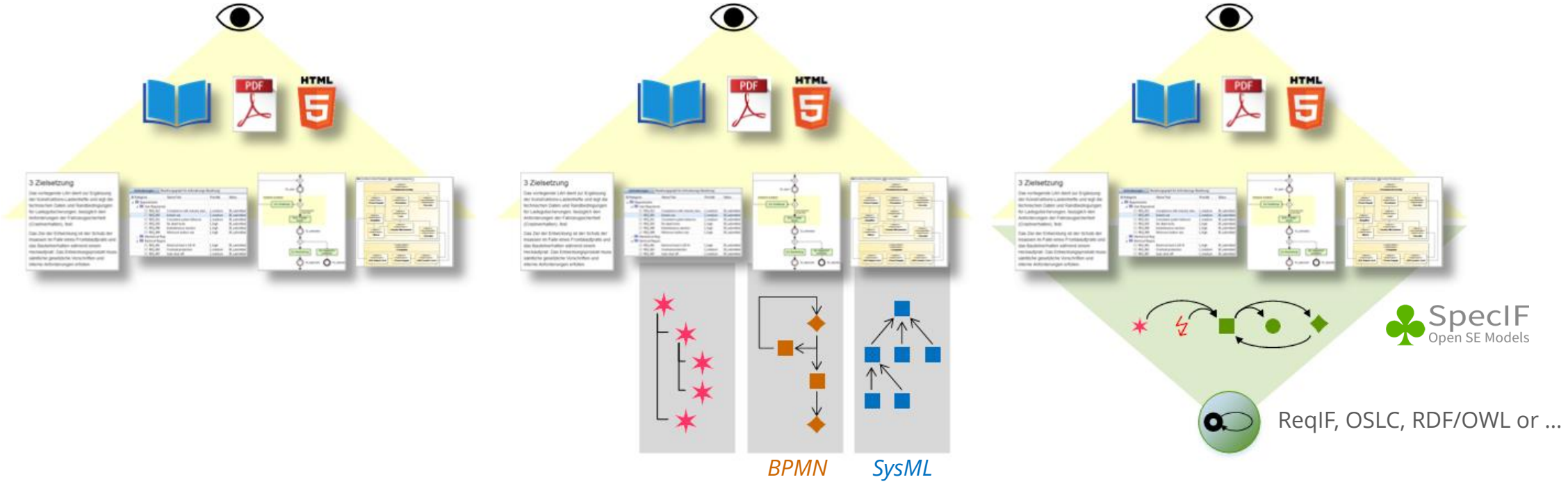


Use Case: Exchange model information between tools

1. Change a state diagram in SysML
2. Check consistency of the specification
3. Validate the behavior in a simulation tool
4. Generate code for an embedded controller
5. Store in a Product Lifecycle Management System



The eye sees the same – SpecIF adds interrelated model data



Creating the „Visible“ → Partial Modelling → Model Integration

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

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

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

SpecIF carries both the Visible and the Meaning

Medium/Format
(The Presentation)

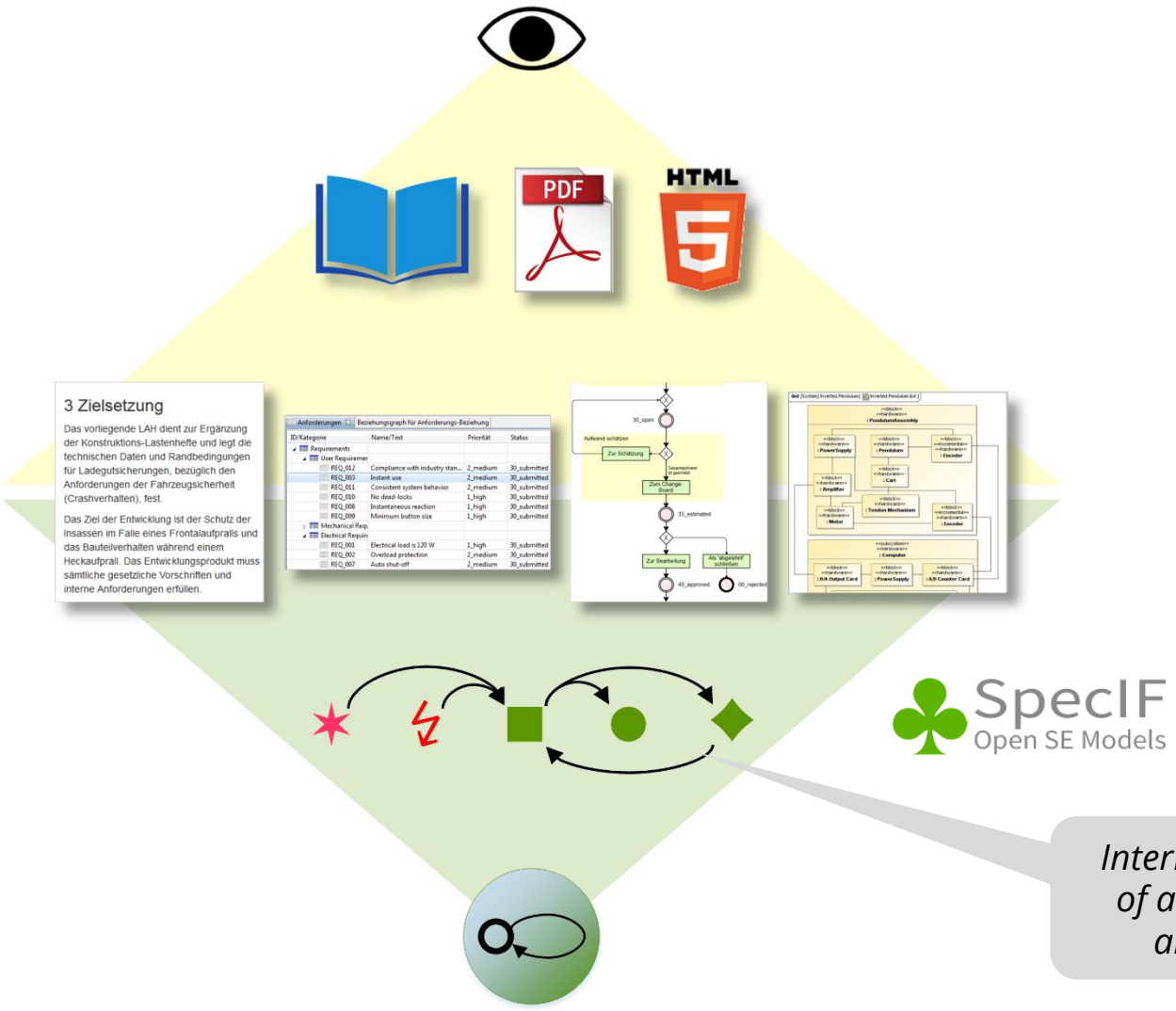
original, loss-less

Notation
(The Visible)

abstracted

Integrated Model
(The Meaning)

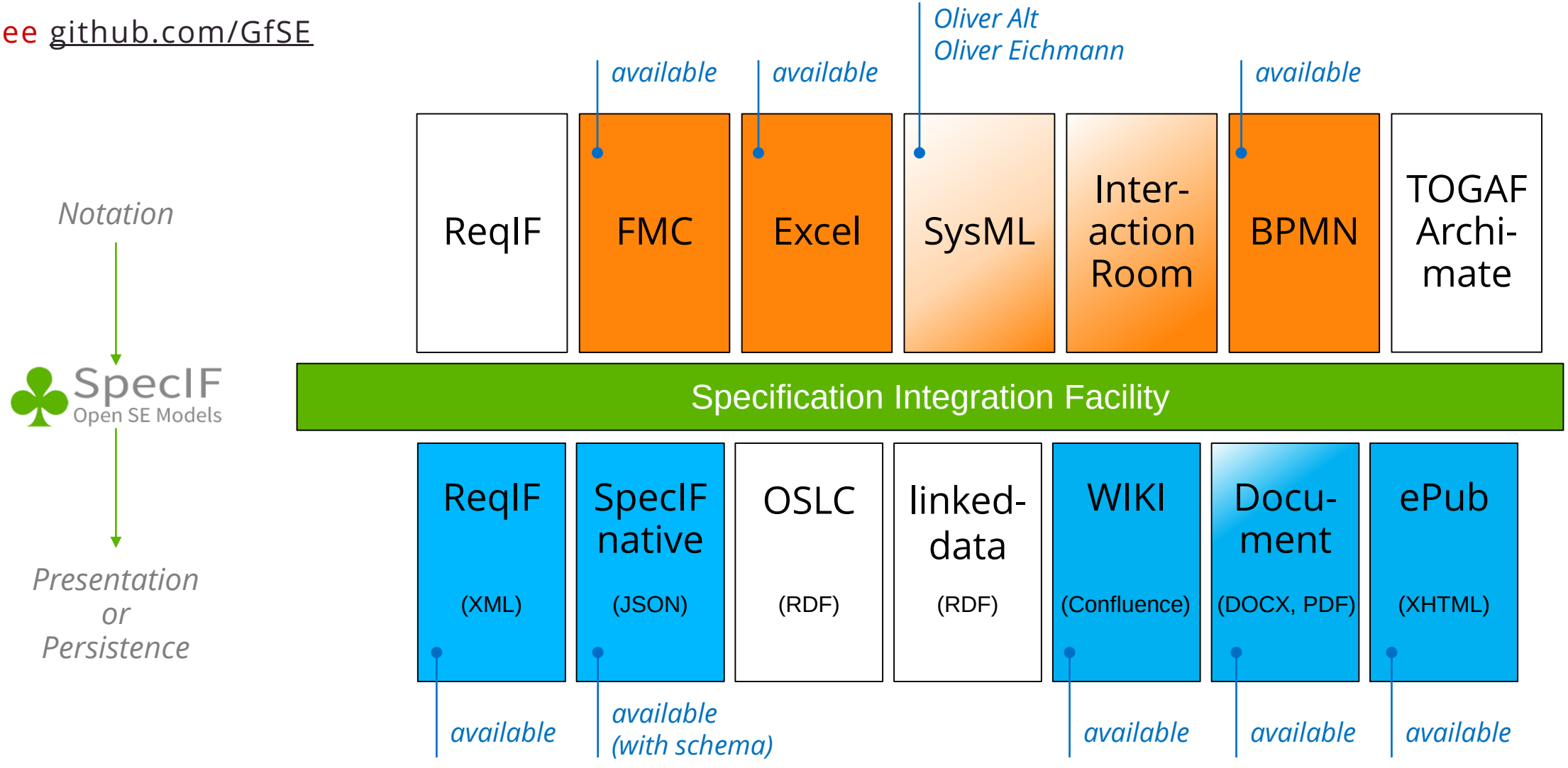
Objects and Relations
(The Persistence)



Interrelates all elements of all model diagrams and all notations

Status …

see github.com/GfSE



Syntax and Semantics

Semantics

(Meaning)

No vocabulary:

Import and export by the same vendor (Extessy → Atego)

2004

RIF

Syntax

(Format)

Syntax and Semantics

Semantics

(Meaning)

No vocabulary:

Import and export by the same vendor (Extessy → Atego)

ReqIF Implementation Guide, oslc_rm Vocabulary:
→ property names

Store/exchange Requirements vendor-independently

2004 → **2011**

Syntax

(Format)

RIF

ReqIF

oslc_rm

distributed

Syntax and Semantics

Semantics

(Meaning)

No vocabulary:

Import and export by the same vendor (Extessy → Atego)

ReqIF Implementation Guide, oslc_rm Vocabulary:
→ property names

Store/exchange Requirements vendor-independently

SpecIF Vocabulary:
→ resource names, statement names, property names property values

Store/exchange Requirements and Models vendor-independently

2004 → **2011** → **2016** →

Syntax

(Format)

RIF

ReqIF

oslc_rm

distributed

SpecIF

simplified polymorph

Syntax and Semantics

Semantics

(Meaning)

No vocabulary:

Import and export by the same vendor (Extessy → Atego)

ReqIF Implementation Guide, oslc_rm Vocabulary:
→ property names

Store/exchange Requirements vendor-independently

SpecIF Vocabulary:
→ resource names, statement names, property names property values

Store/exchange Requirements and Models vendor-independently

Applicable to all formats below

2004 → **2011** → **2016** →

Syntax

(Format)

RIF

ReqIF

oslc_rm

SpecIF

distributed

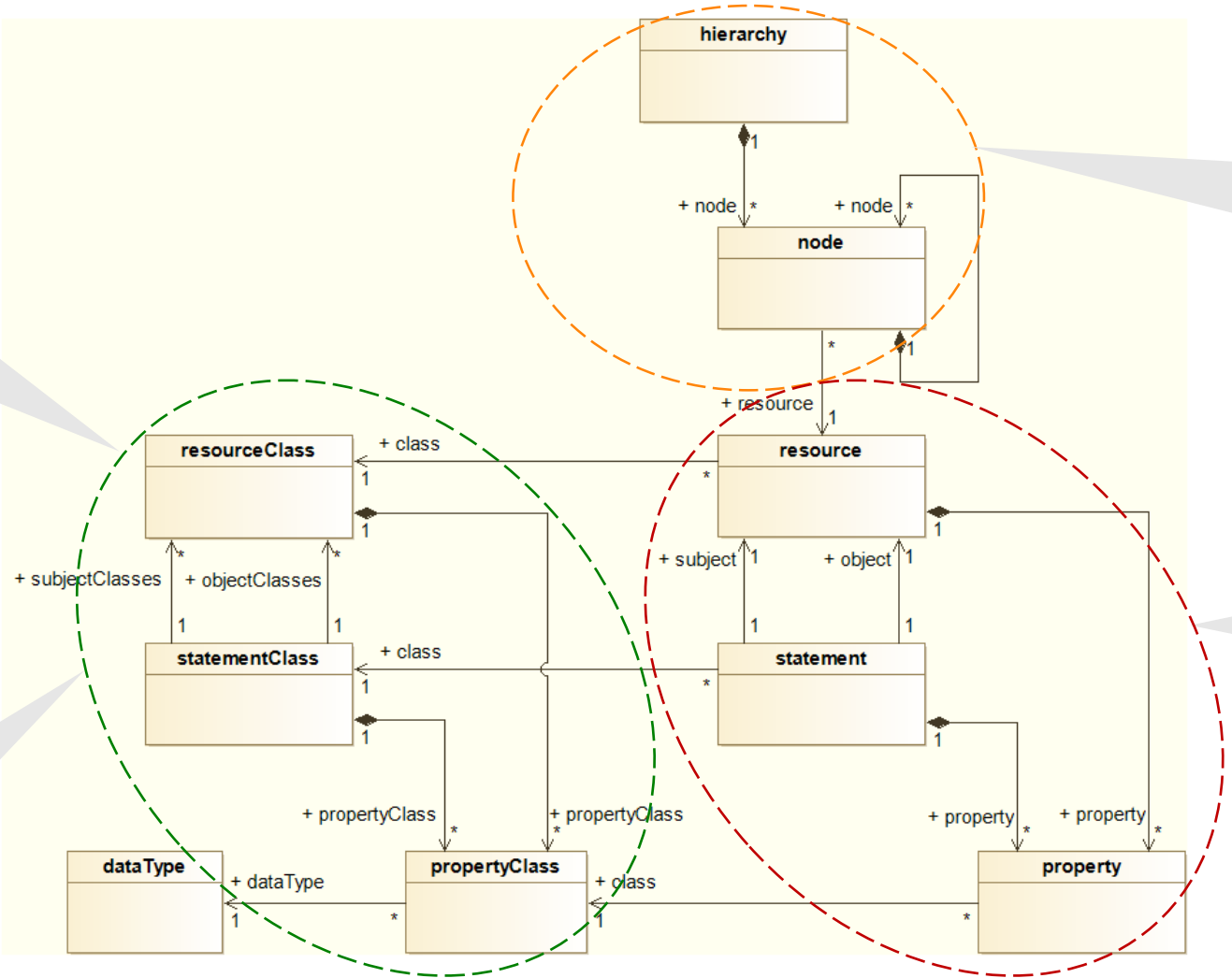
simplified polymorph

Lossless Transformation possible

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

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

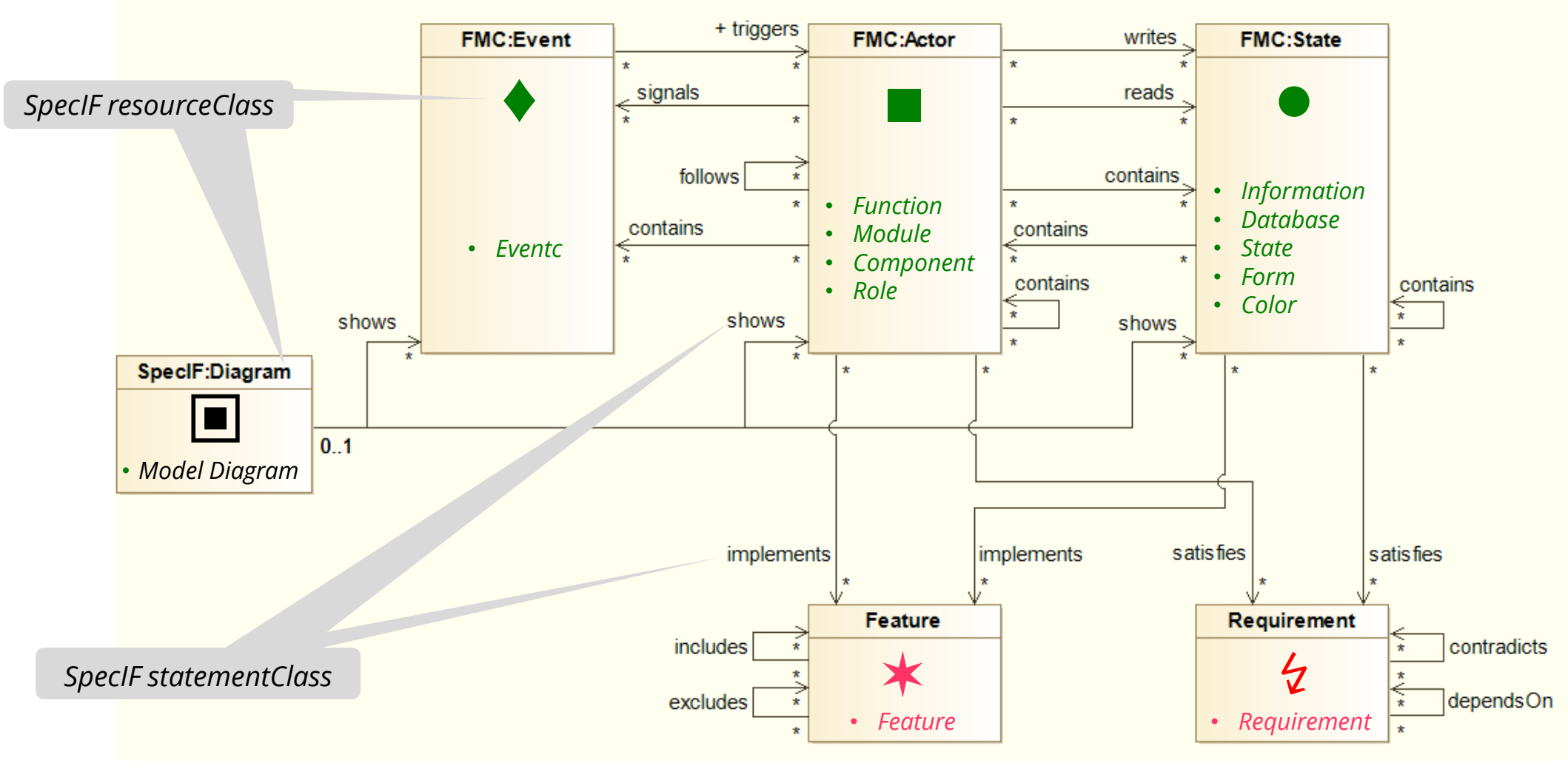
SpecIF model with Resource- and Statement-Classes



SpecIF data (payload) with Resources and Statements

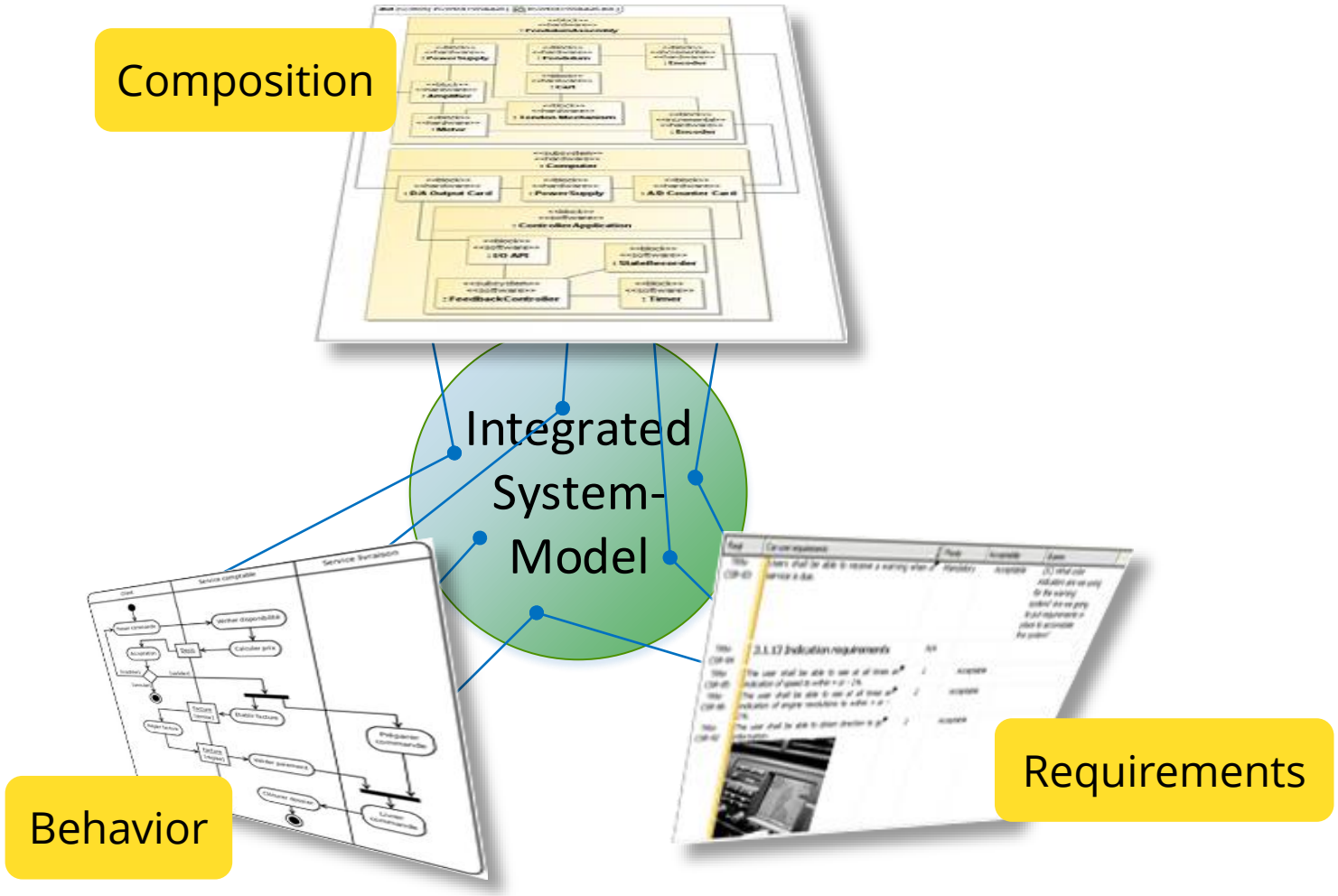
The classes can be defined at runtime („dynamic model“)

The SpecIF Integration Model with 5 Fundamental Model-element Types



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

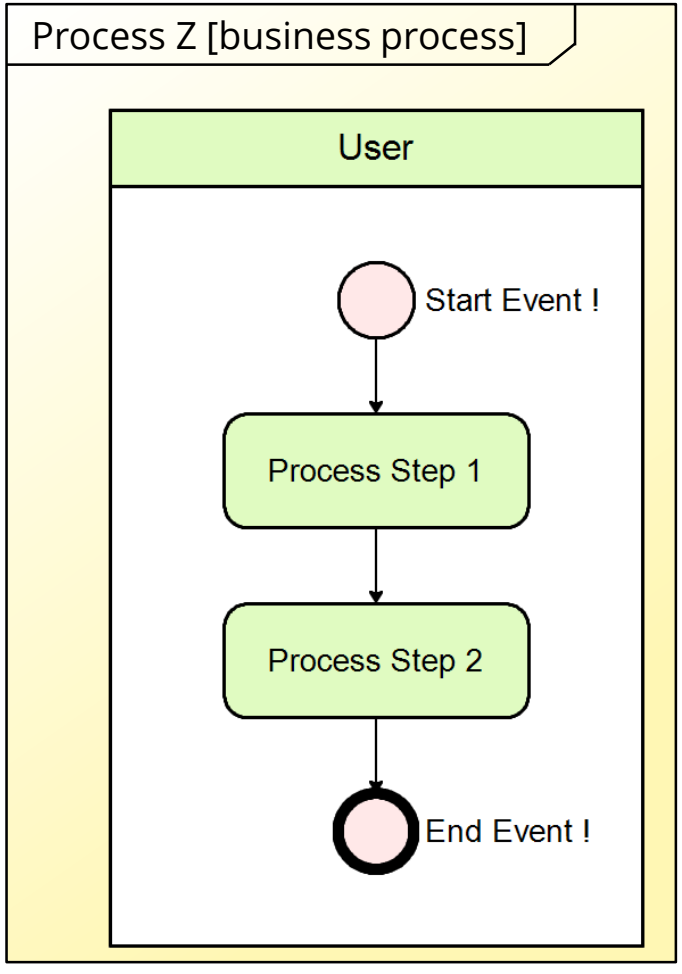
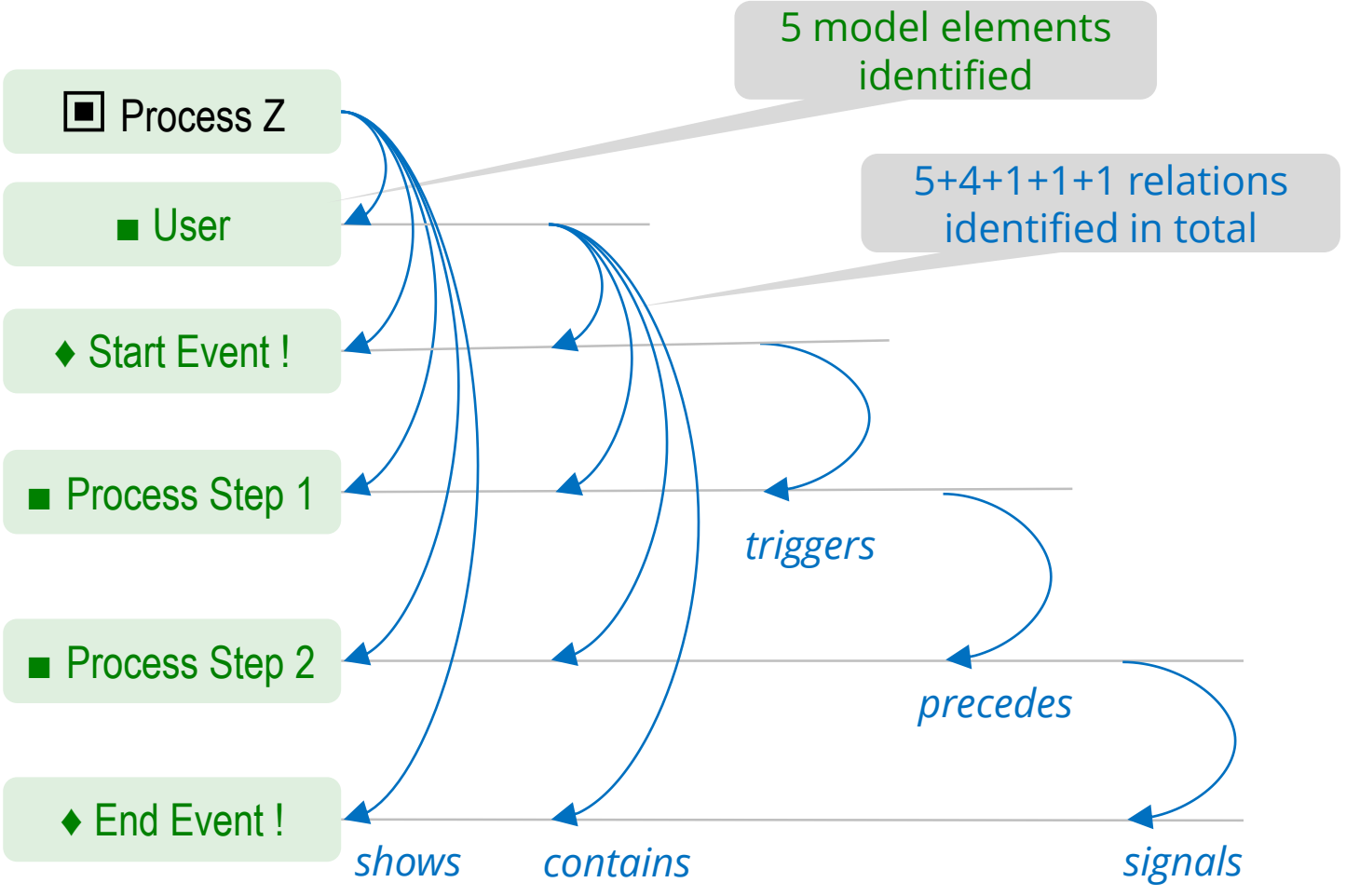


Abstraction: All model diagram types (notations) consist of three fundamental model-element types*

	Model-element Type	▣ Model-diagram Type (Notation)				
		BPMN Business-process	State-machine	System-composition	Organi-sation 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	View 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
ϟ	**Requirement**						X

Taking a model diagram, automatically identify model elements and their relations to build a semantic net