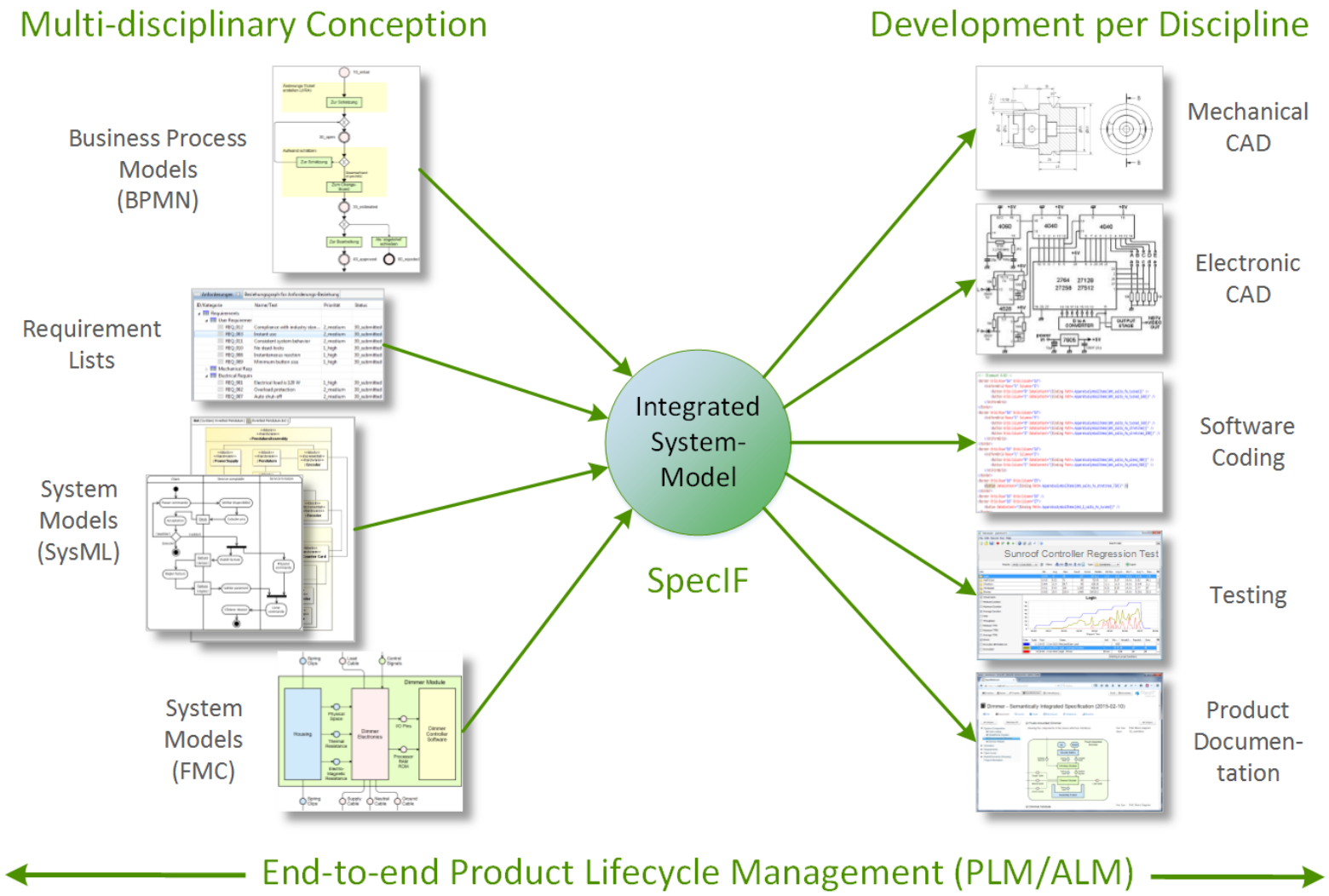
# How to Automatically Unterstand and Integrate System-models … and how SpecIF can help.

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

# Topics Today

1. The idea behind model integration … and SpecIF
   - Purpose
   - Use-Cases
   - Principles and Approach
2. An example
3. Positioning ReqIF, OSLC and SpecIF
4. Status

# Purpose: SpecIF integrates information from different sources



# Diverse systems name information elements differently

## *Requirement*

| | |
|---|---|
| *Object Short Text:* | Crisp Naming |
| *Object Text:* | Longer description as formatted text, sometimes with a diagram. |
| *Object Priority:* | 1_high |
| *Last modified on:* | 2017-09-12 |

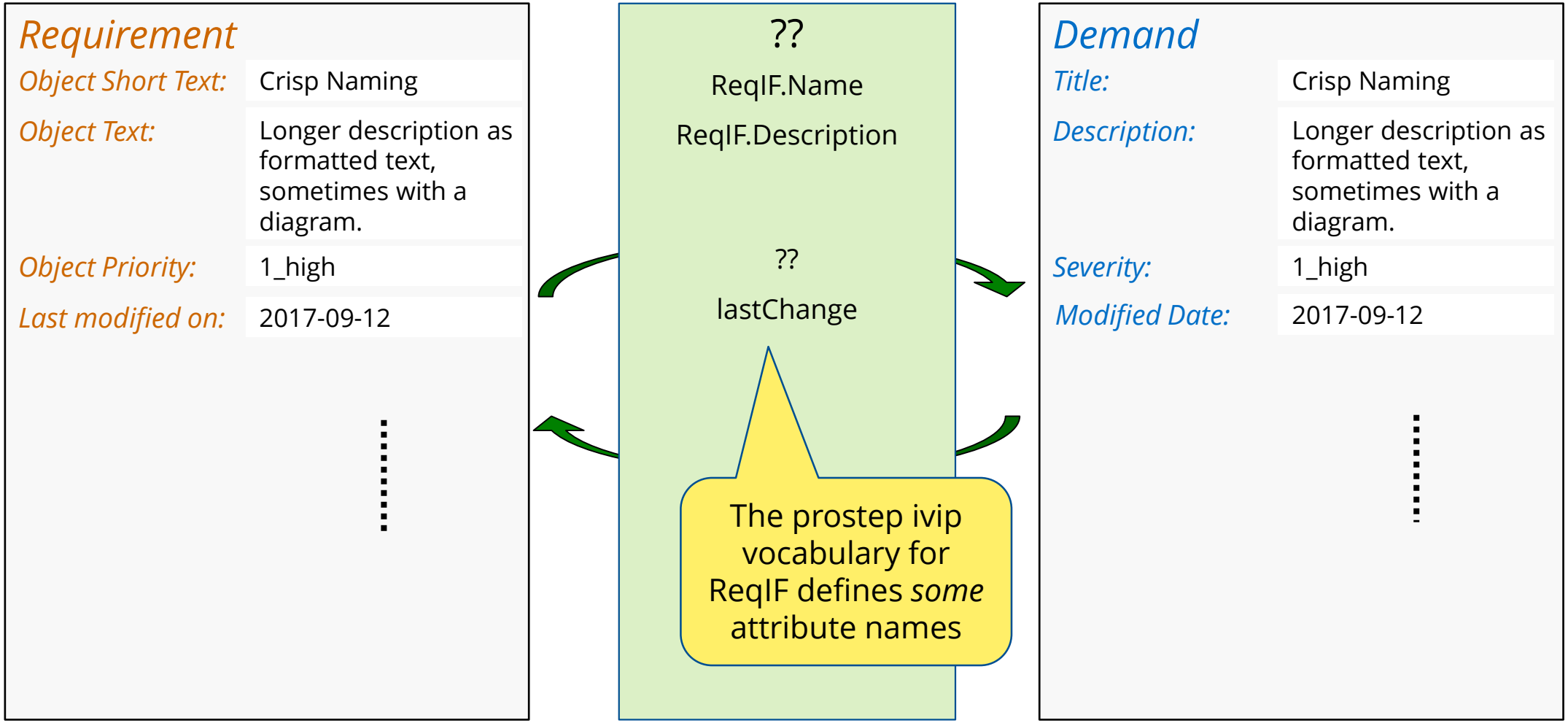
⋮

## *Demand*

| | |
|---|---|
| *Title:* | Crisp Naming |
| *Description:* | Longer description as formatted text, sometimes with a diagram. |
| *Severity:* | 1_high |
| *Modified Date:* | 2017-09-12 |

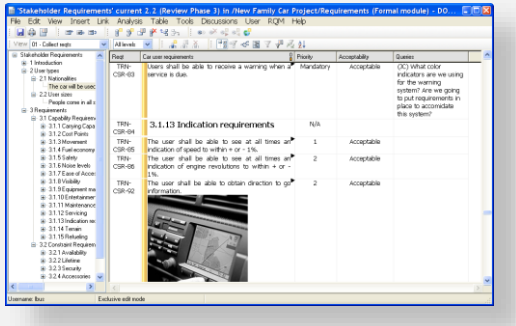
⋮

# A vocabulary or ontology assigns a meaning to information elements and allows information exchange without negotiation



# Use-Case: Exchange requirement specs between OEM and Supplier

## OEM

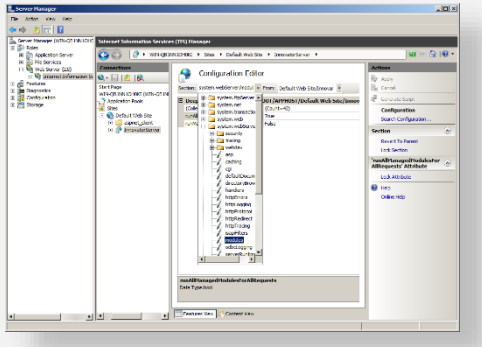


- Requirement-Management

ReqIF

ReqIF

## Supplier



- PLM/ALM

Standard ReqIF with
- prostep ivip vocabulary

# Use-Case: Requirement Agreement using the HIS-Process

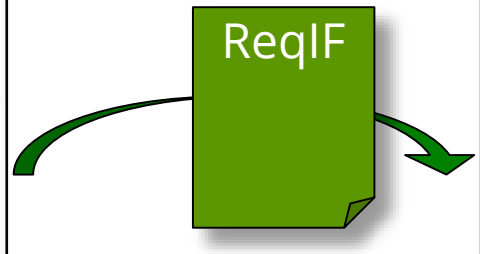
| *Requirement* | |
|---|---|
| *ReqIF.Name:* | Crisp Naming |
| *ReqIF.Description:* | Longer description as formatted text, sometimes with a diagram. |
| *Priority:* | 1_high |
| ⋮ | |
| *Supplier-Status:* | |
| *Supplier-Comment:* | |
| *OEM-Status:* | to-evaluate |
| *OEM-Comment:* | |

*An attribute pair for each of the suppliers*

*An attribute pair for the OEM*

# Use-Case: Requirement Agreement using the HIS-Process

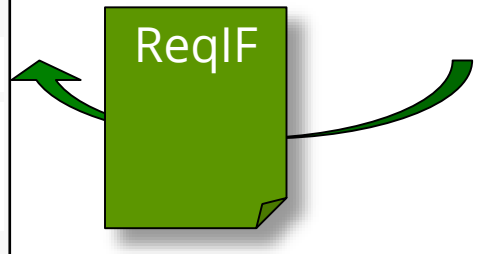
| *Requirement* | |
|---|---|
| *ReqIF.Name:* | Crisp Naming |
| *ReqIF.Description:* | Longer description as formatted text, sometimes with a diagram. |
| *Priority:* | 1_high |
| *Supplier-Status:* | |
| *Supplier-Comment:* | |
| *OEM-Status:* | to-evaluate |
| *OEM-Comment:* | |




| *Requirement* | |
|---|---|
| *ReqIF.Name:* | Crisp Naming |
| *ReqIF.Description:* | Longer description as formatted text, sometimes with a diagram. |
| *Priority:* | 1_high |
| *Supplier-Status:* | partly-agreed |
| *Supplier-Comment:* | Comments or questions of the supplier |
| *OEM-Status:* | to-evaluate |
| *OEM-Comment:* | |

# Use-Case: Requirement Agreement using the HIS-Process

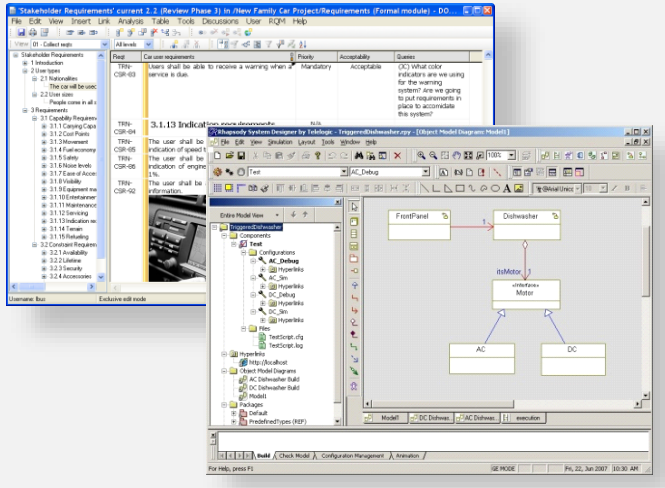
| *Requirement* | |
|---|---|
| *ReqIF.Name:* | Crisp Naming |
| *ReqIF.Description:* | Longer description as formatted text, sometimes with a diagram. |
| *Priority:* | 1_high |
| *Supplier-Status:* | partly-agreed |
| *Supplier-Comment:* | Comments or questions of the supplier |
| *OEM-Status:* | not-accepted |
| *OEM-Comment:* | Reactions and Answers of the OEM |




| *Requirement* | |
|---|---|
| *ReqIF.Name:* | Crisp Naming |
| *ReqIF.Description:* | Longer description as formatted text, sometimes with a diagram. |
| *Priority:* | 1_high |
| *Supplier-Status:* | partly-agreed |
| *Supplier-Comment:* | Comments or questions of the supplier |
| *OEM-Status:* | to-evaluate |
| *OEM-Comment:* | |

# Use-Case: Exchange model-based specs between OEM and Supplier

## OEM

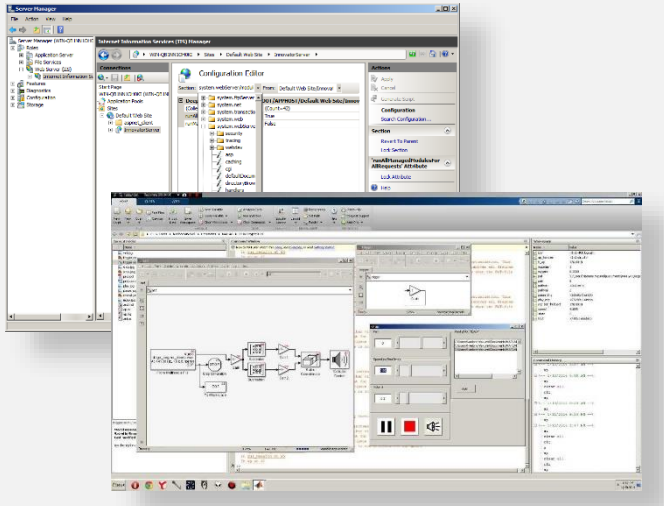


- Requirement-Management
- System-Modelling

ReqIF

ReqIF

## Supplier

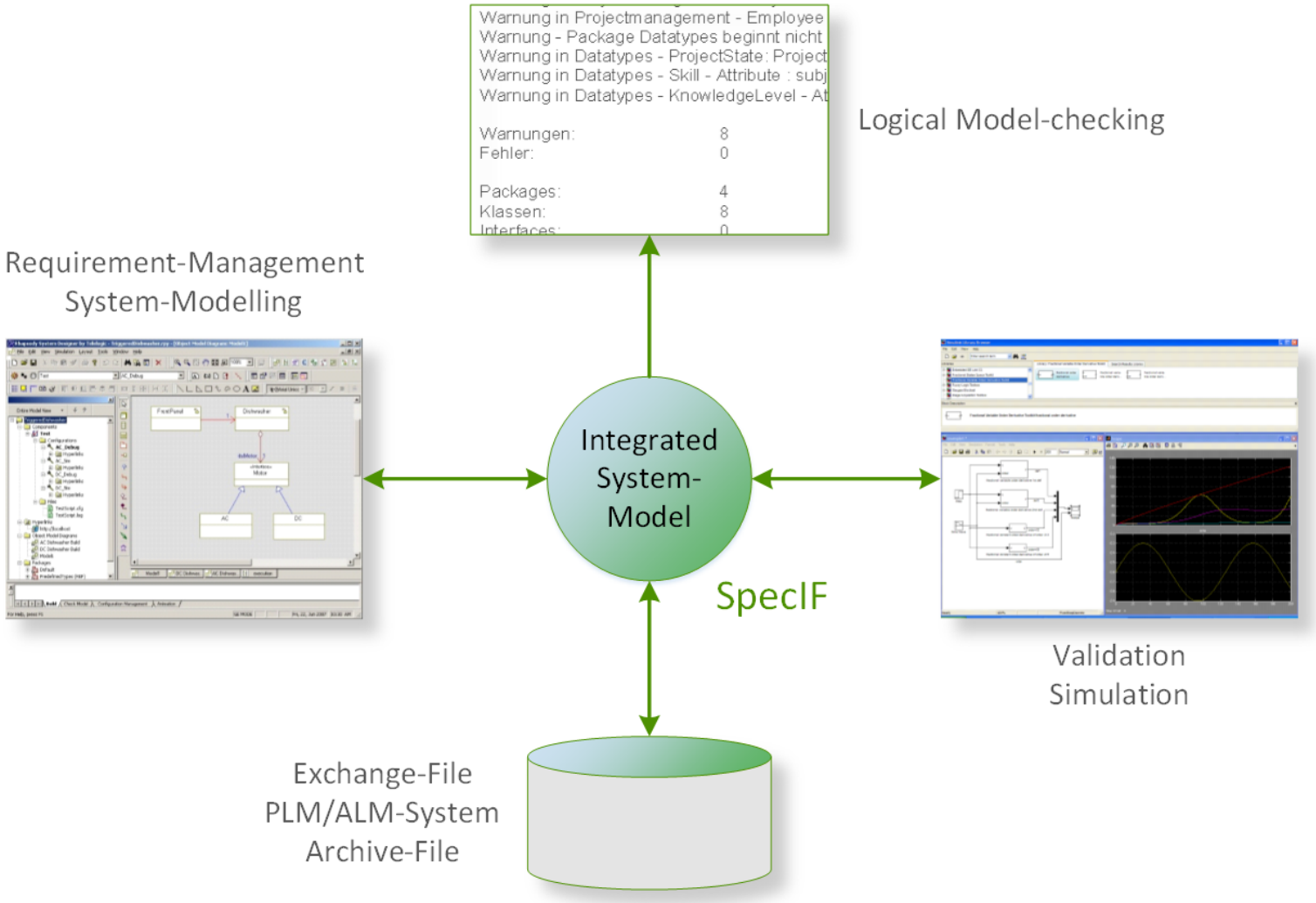


- PLM/ALM
- Simulation

Standard ReqIF with
- SpecIF Vocabulary
- SpecIF Assertions

# 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



# What is SpecIF ?

- „**Spec**ification **I**ntegration **F**acility“

- SpecIF *adds* conventions to express meaning
  to 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“)

     „An Actor *satisfies* a Requirement“
     „An Event *triggers* an Actor“

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

# SpecIF carries both the Visible and the Meaning

*Medium/Format*
*(The Presentation)*

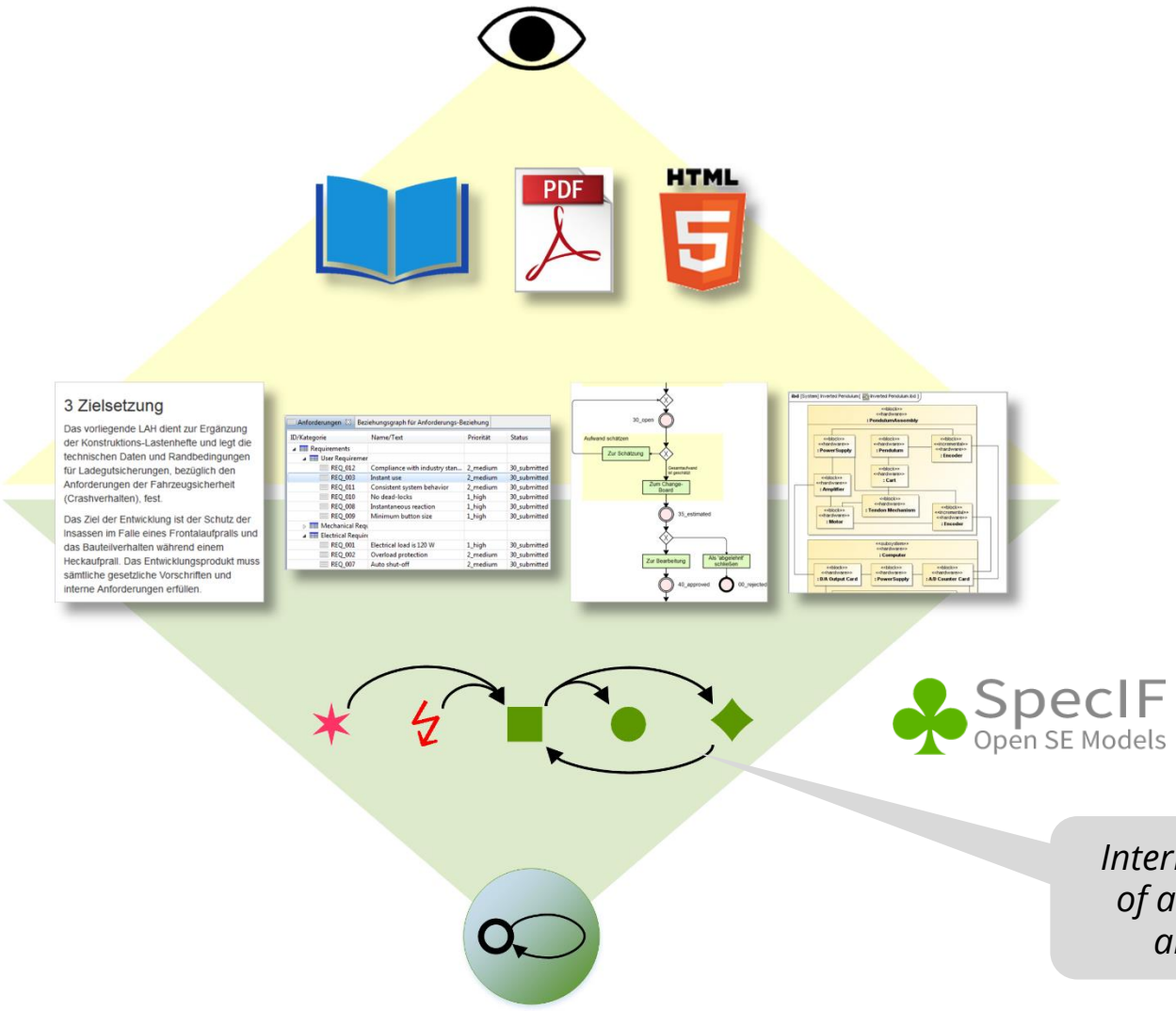
*original, loss-less*

*Notation*
*(The Visible)*

*abstracted*

*Integrated Model*
*(The Meaning)*

*Objects and Relations*
*(The Persistence)*




*Interrelates all elements of all model diagrams and all notations*

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

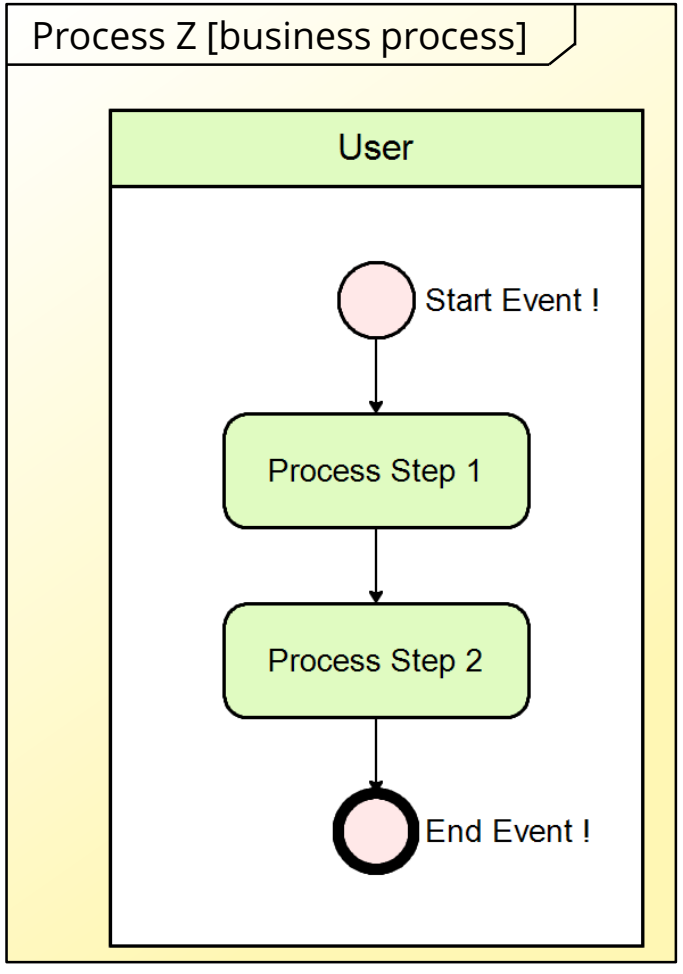
| | Model-element Type | ▣ Model-diagram Type (Notation) | | | | |
|---|---|---|---|---|---|---|
| | | BPMN Business-process | State-machine | System-composition | Organi-sation Chart | UML Classes |
| ■ | **Actor** | X | (X) | X | X | |
| ● | **State** | (X) | X | X | | X |
| ♦ | **Event** | X | X | | | |

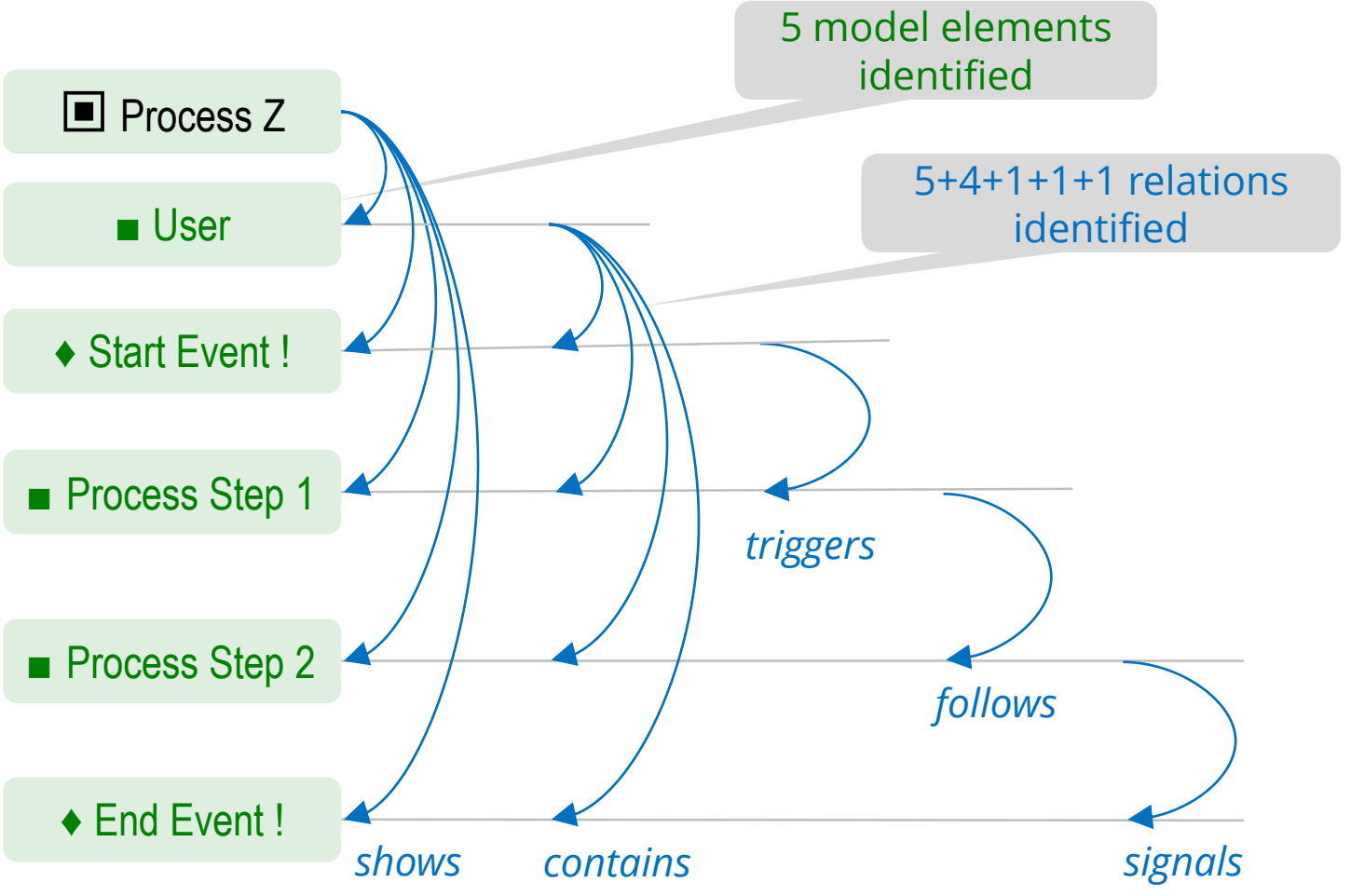
* Prof. Dr. Siegfried Wendt, Founding Director of the Hasso-Plattner-Institute, Potsdam: Fundamental Modelling Concepts

# 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 |

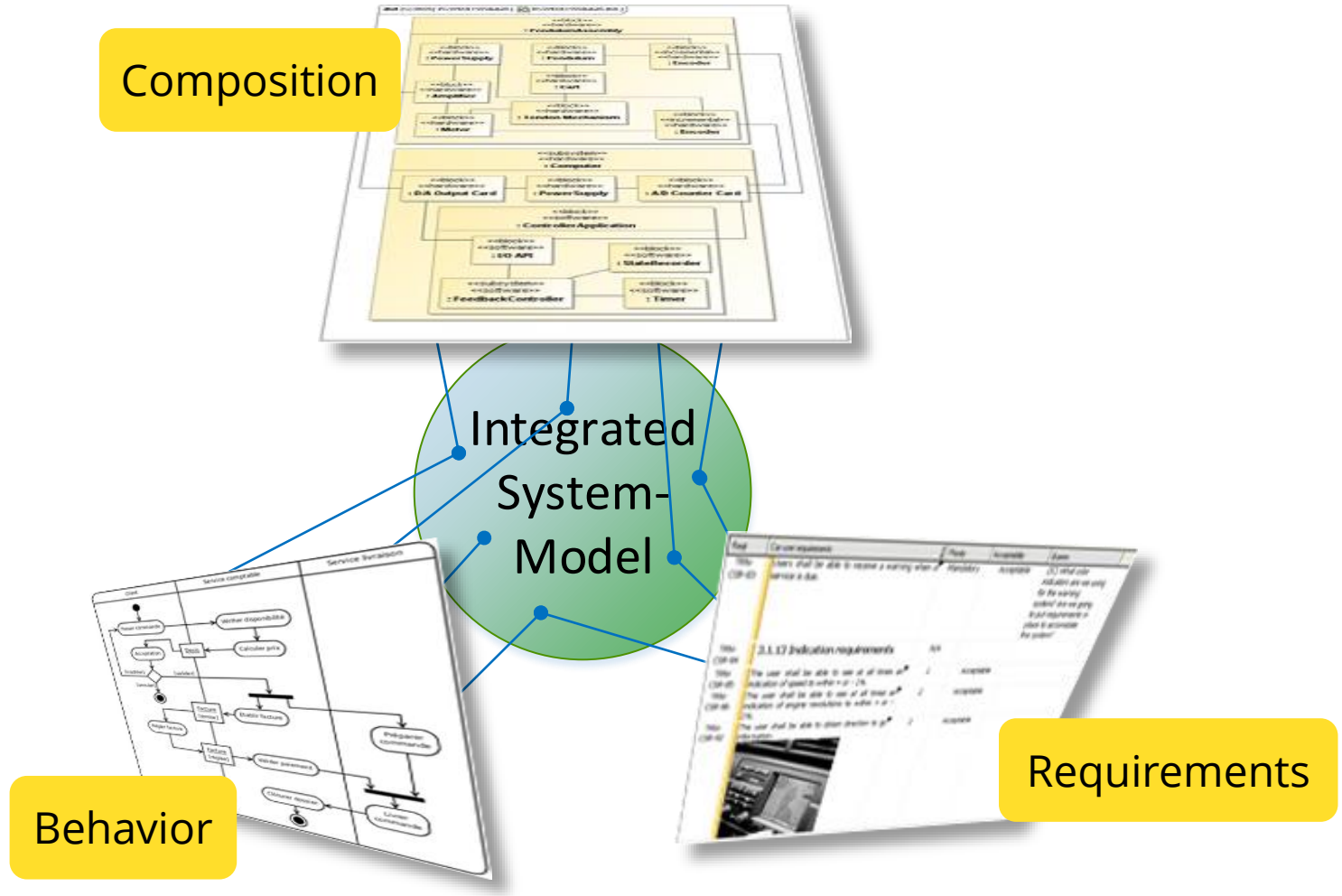
# Taking a model diagram, automatically identify model elements and their relations to build a semantic net



# 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



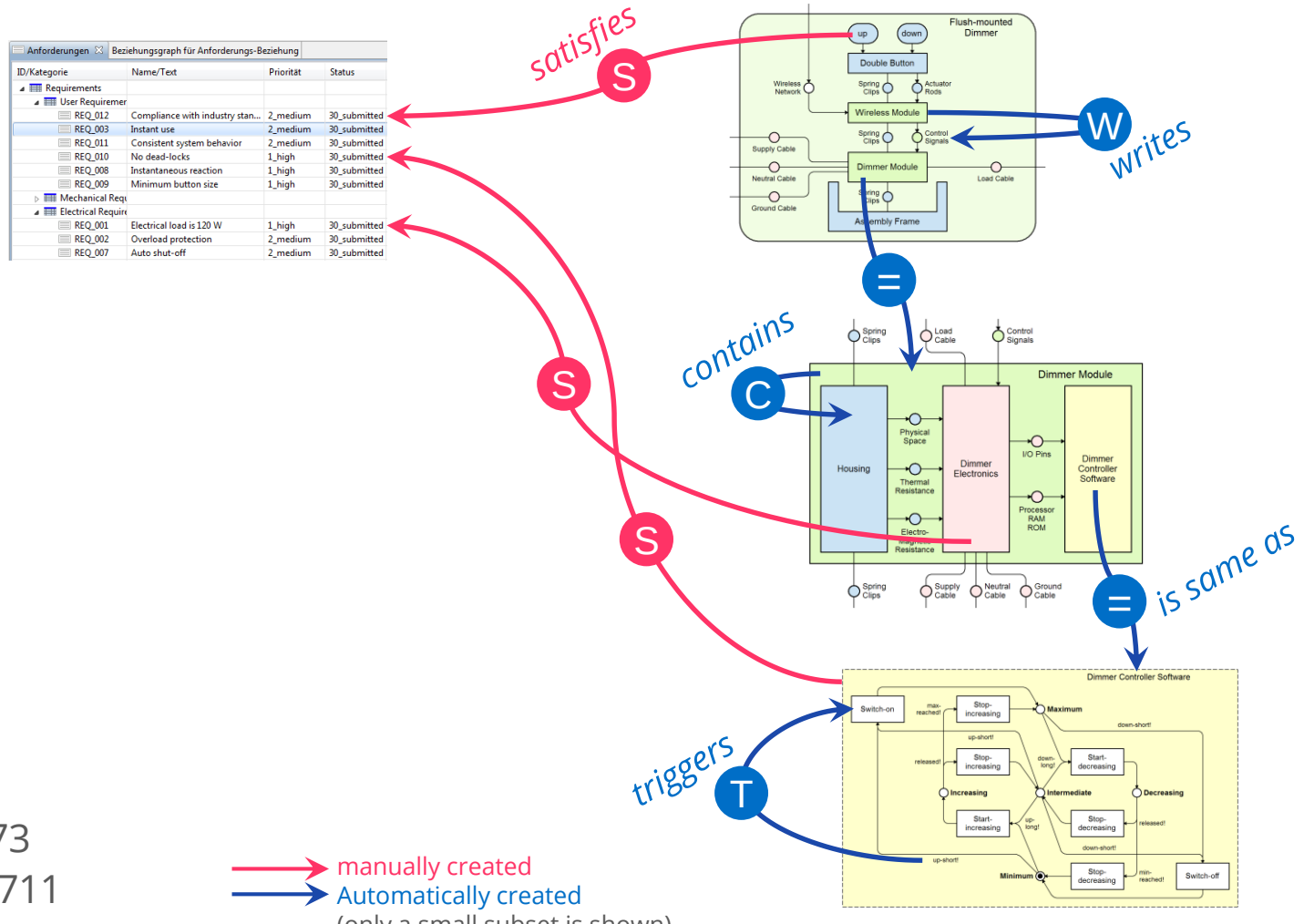
# Summary: 4 steps to create an Integrated System Model

1. Identify
   Every element on a diagram
   Every line in a spread-sheet

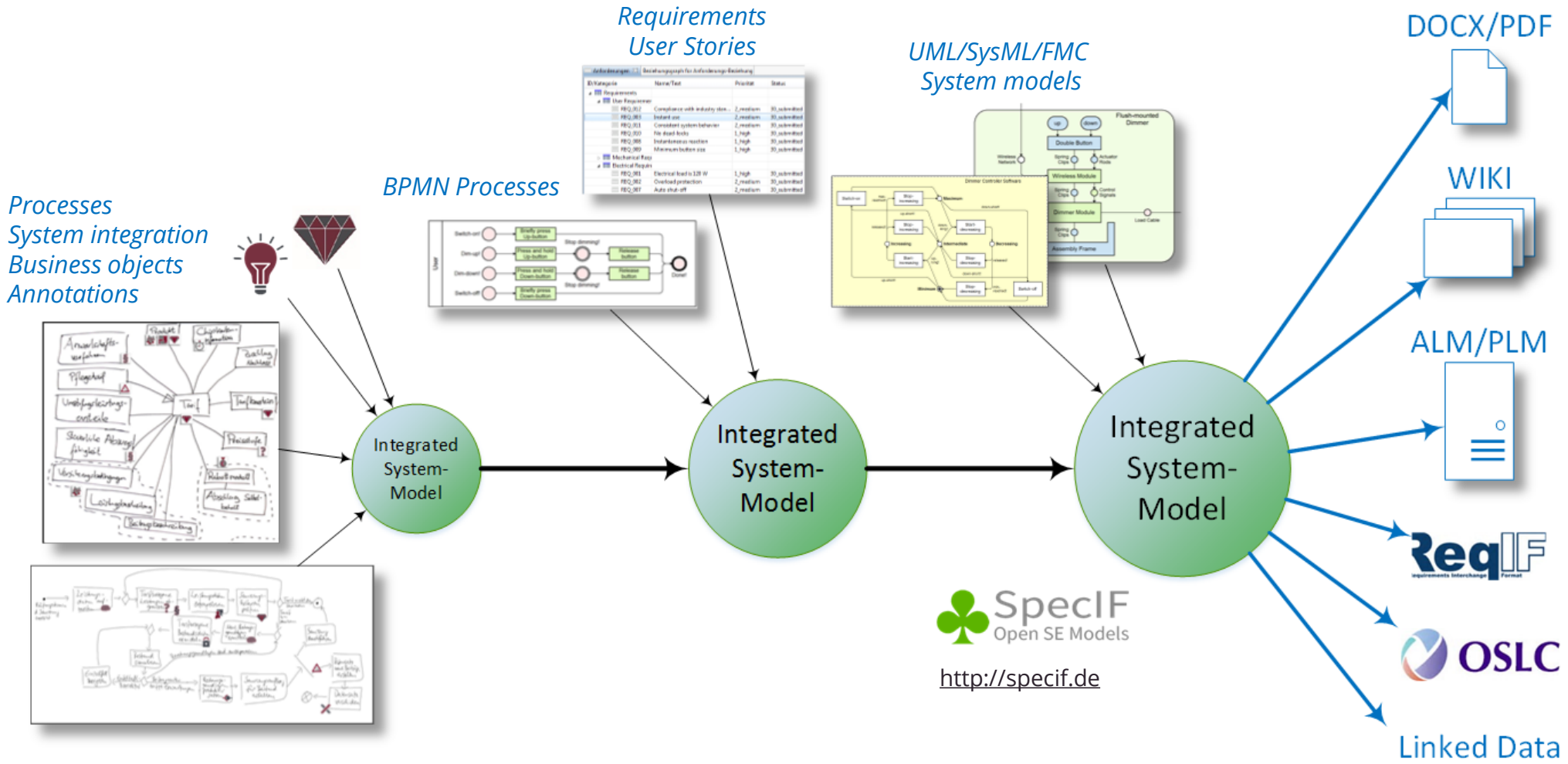
2. Assign fundamental element type
   - ■ Actor
   - ● State
   - ♦ Event
   - ✶ Feature
   - ϟ Requirement

3. Consolidate

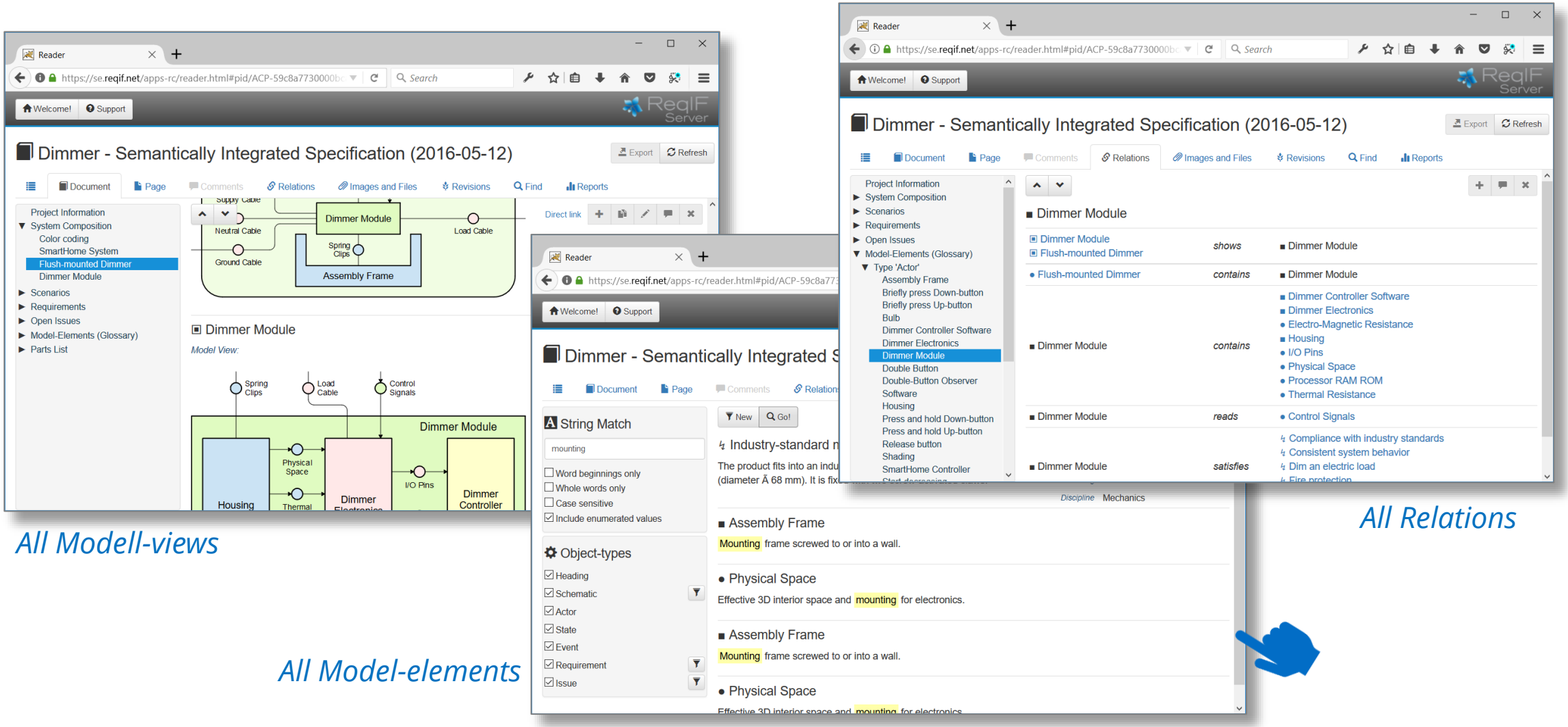
4. Semantically Interrelate
   Component-070 *contains* Component-073
   Component-070 *satisfies* Requirement-4711



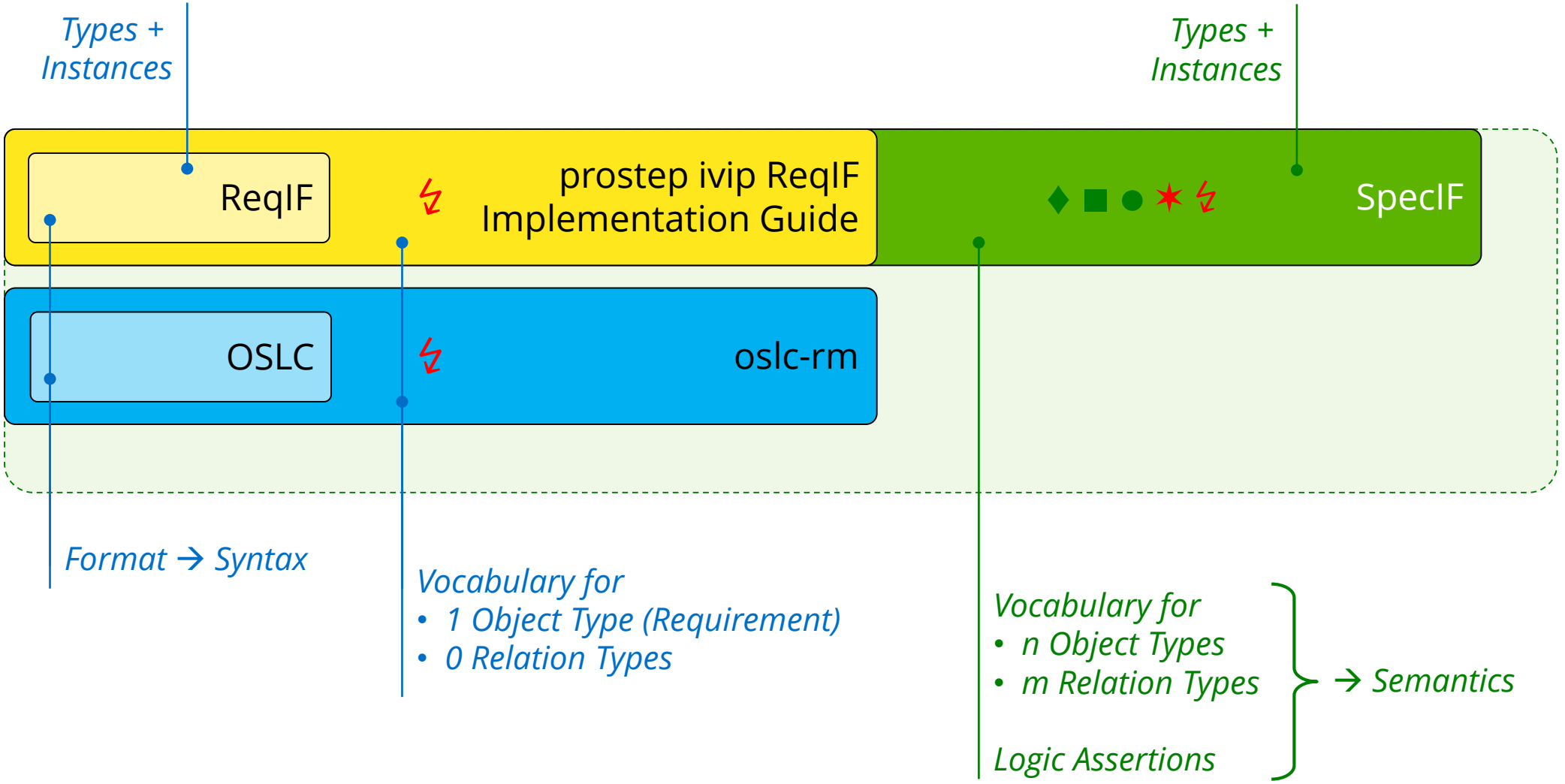
# Add partial models step-by-step … and use/deliver in known technical formats



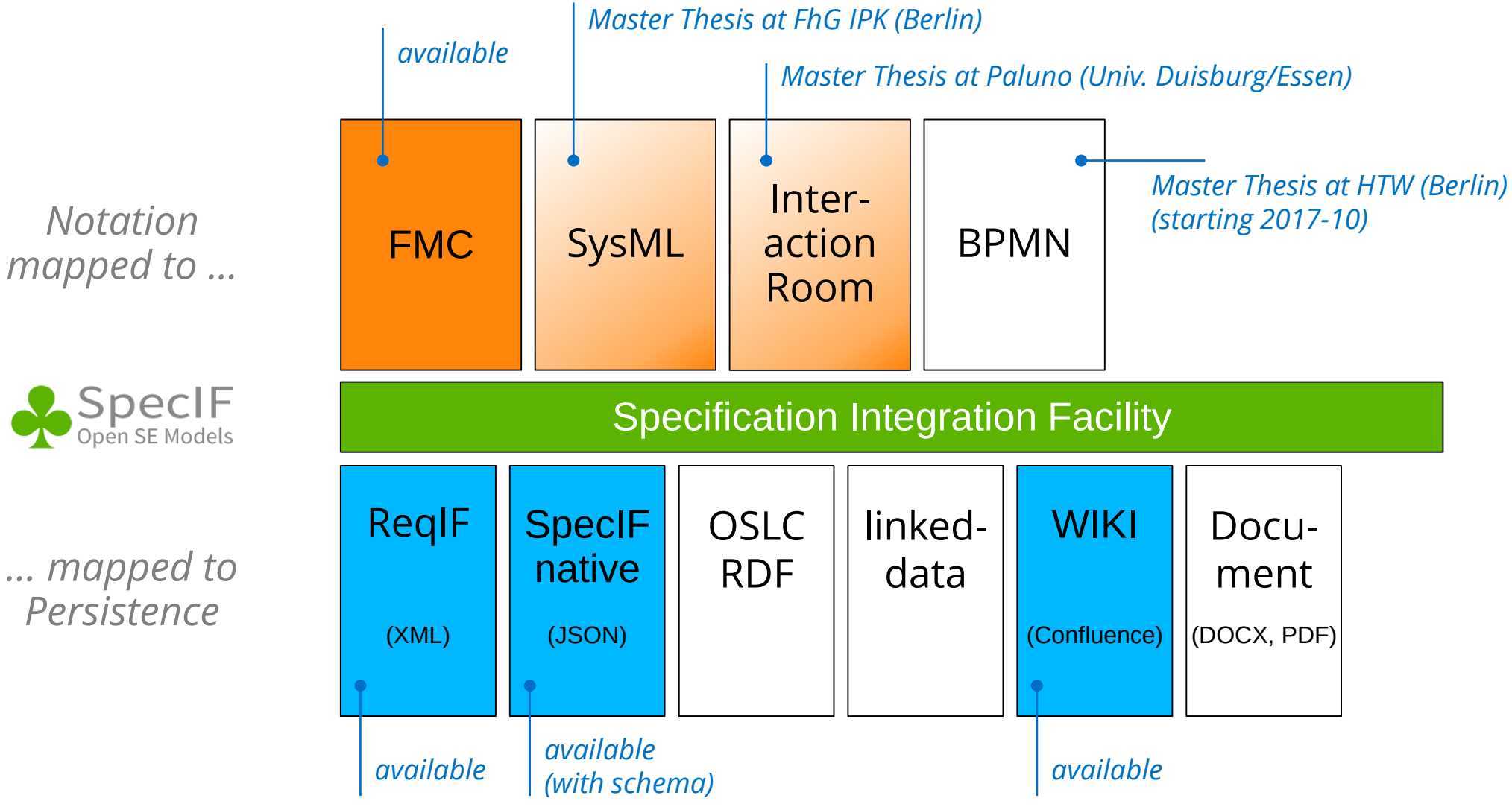
# Example: Search, navigate and audit in a common context



# Positioning ReqIF, OSLC and SpecIF



# Status


Notation mapped to ...
available
Master Thesis at FhG IPK (Berlin)
Master Thesis at Paluno (Univ. Duisburg/Essen)
Master Thesis at HTW (Berlin) (starting 2017-10)
FMC
SysML
Inter-action Room
BPMN
SpecIF
Open SE Models
Specification Integration Facility
... mapped to Persistence
ReqIF
(XML)
SpecIF native
(JSON)
OSLC RDF
linked-data
WIKI
(Confluence)
Docu-ment
(DOCX, PDF)
available
available (with schema)
available

# SpecIF Goals and Benefits

- Exchange model-based spezifications between tools and organizations.
- Combine (formatted) texts and models from different tools.
- Search, navigate, and check in a common context.
- Manage the product lifecycle from the beginning („end-to-end“):
  - Overarching engineering-disciplines
  - Combining methods
  - Technology-neutral
  - Vendor-neutral
  - Schema-conforming
  - Standard-conforming
  - Open and cooperative

# 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, March 2017.

[13] Mochine, P.; Sünnetcioglu, A.; Dungern, O.v.; Stark, R.: SysML-Modelle maschinell verstehen und verknüpfen. To appear: TdSE Tag des Systems Engineering der GfSE, Paderborn, October 2017.

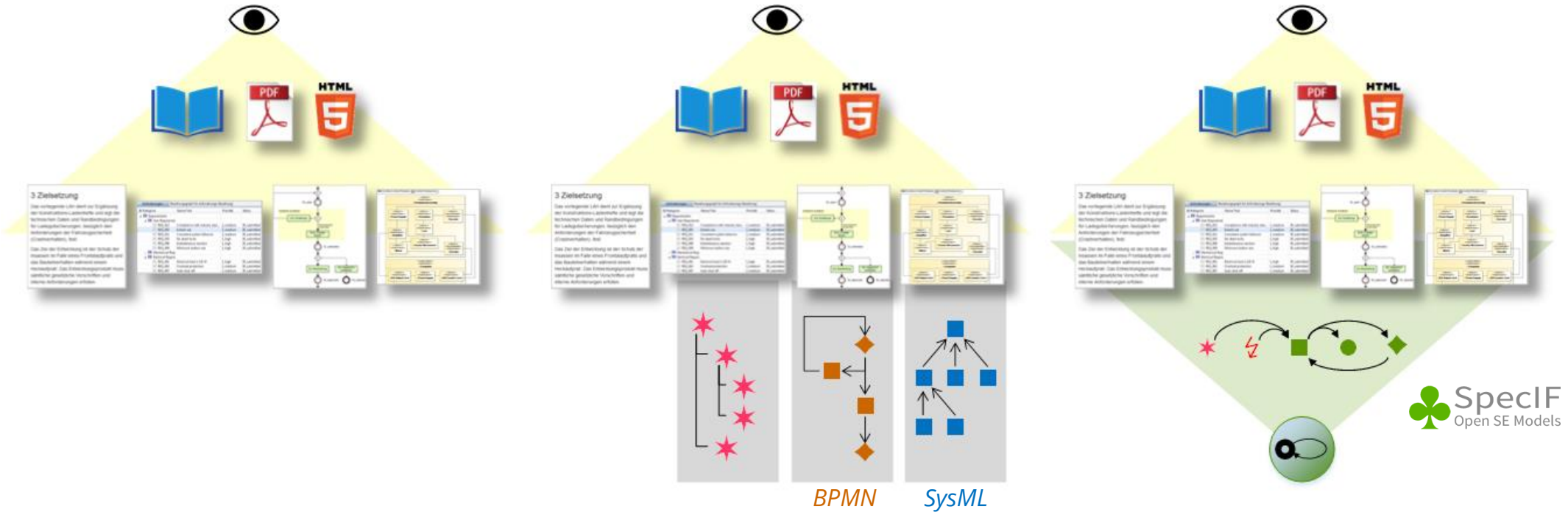
# Interesting ?

oskar.dungern@adesso.de


**adesso AG**
Stockholmer Allee 20
44269 Dortmund
T +49 231 7000-7000
F +49 231 7000-1000
www.adesso.de

# Details for further discussion

# The eye sees the same – SpecIF adds interrelated model data




## Creating the „Visible“ → Partial Modelling → Model Integration

- Text editing and image „drawing“
- Brain and Discipline to build and keep it consistent

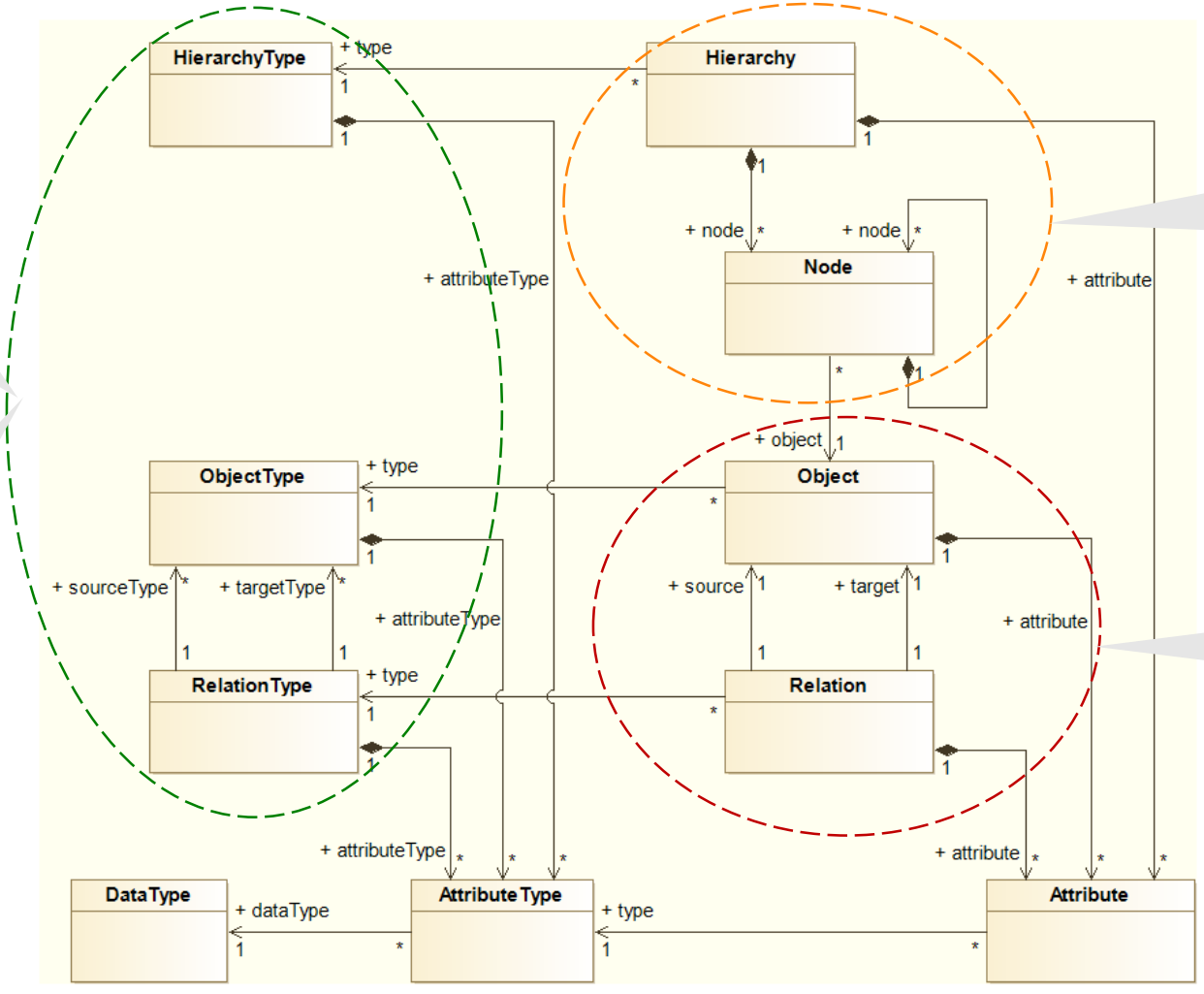
- Text editing and modelling per method
- Tool support within the methods

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

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

*SpecIF model with Hierarchy-, Object- and Relation-types*

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




*Hierarchical ordering of Objects (e.g. for convenient reading)*

*SpecIF data (payload) with Objects and Relations*

# The discussed SpecIF Model with 5 Fundamental Model-element Types is proven for Model Integration