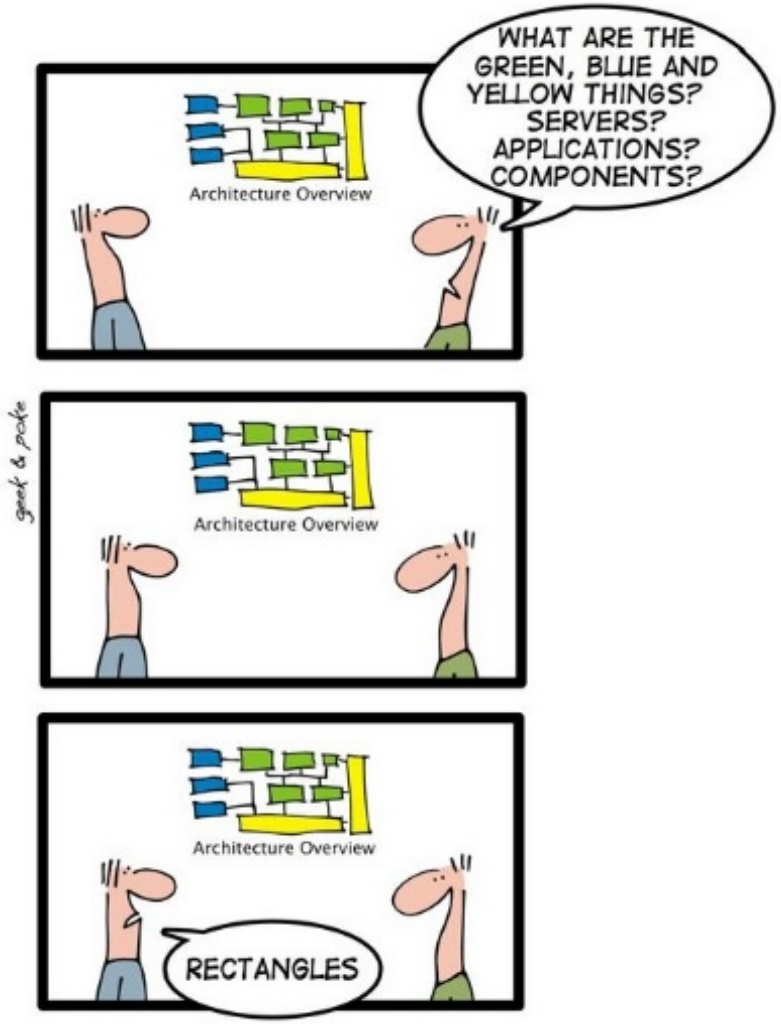
Ontology-based Metamodeling

Knut Hinkelmann

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Making the Knowledge in Models explicit



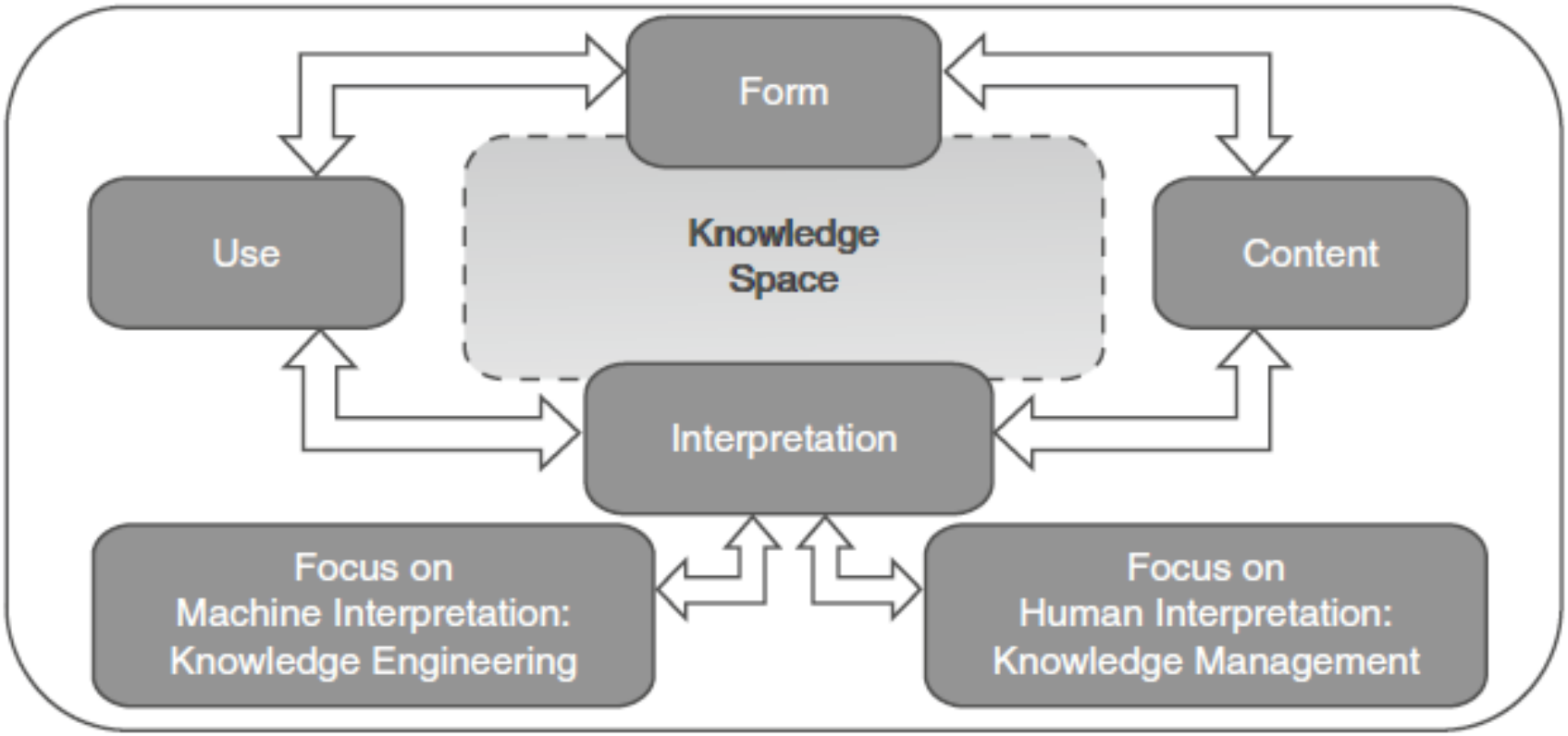
- Humans «know» the meaning of the modeling objects.
  - Example: Process Modeling

«Cook pasta» is a task about preparing food

- The objective is to represent the knowledge so that it can be interpreted by a system for decision making and problem solving

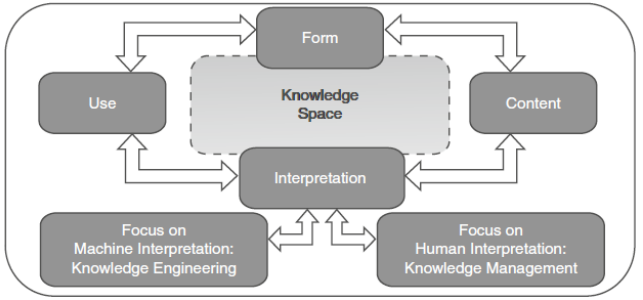
Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Dimensions of a Knoweldge Space



Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Dimensions of the Knowledge Space



Form: modeling language

Content: model information, represented in the description of elements

Use:

- process optimization requires knowledge about time and costs
- selection of a cloud service require knowledge about data and functionality

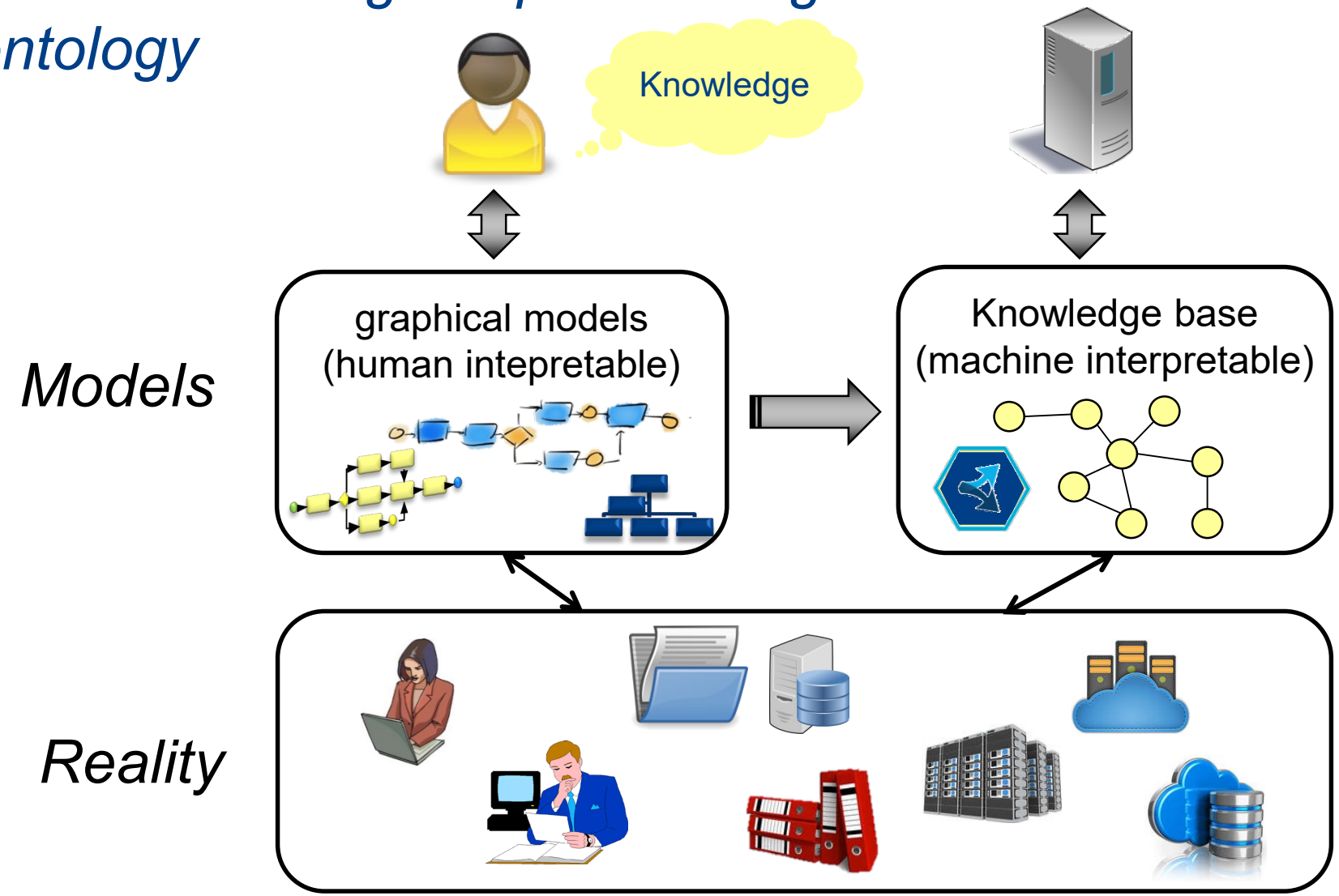
- ***Form***: Syntax and semantic of **modeling language**.
- ***Content***: **Domain** in which knowledge engineering is applied.
- ***Use***: Stakeholders and their concerns determine the relevant subset of the knowledge and reasoning
- ***Interpretation***: Graphical models typically are cognitively more adequate for human interpretation and ontologies can be interpreted by machines.

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Semantic Lifting

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

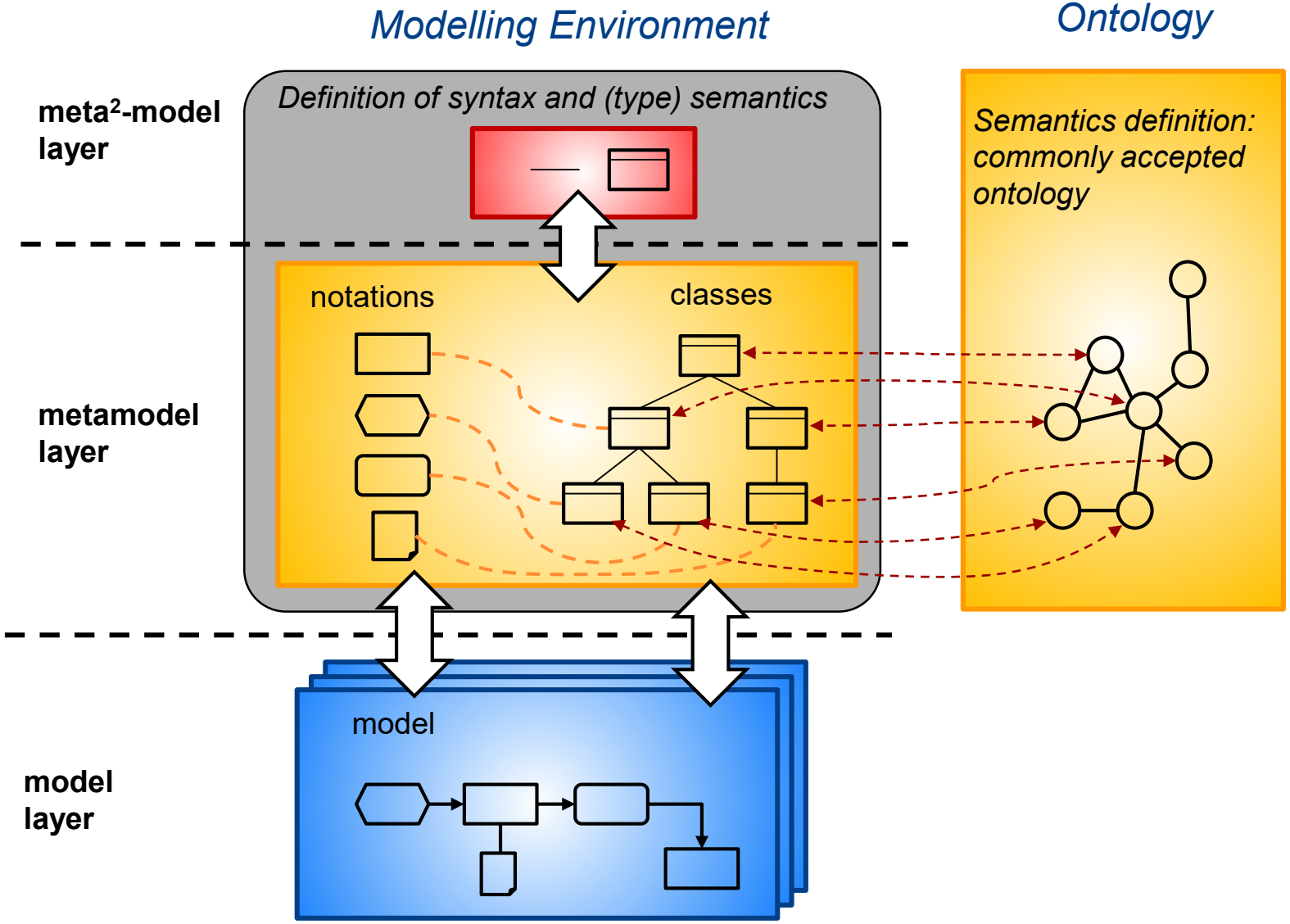
Semantic Lifting: Map knowledge of models into an ontology



Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

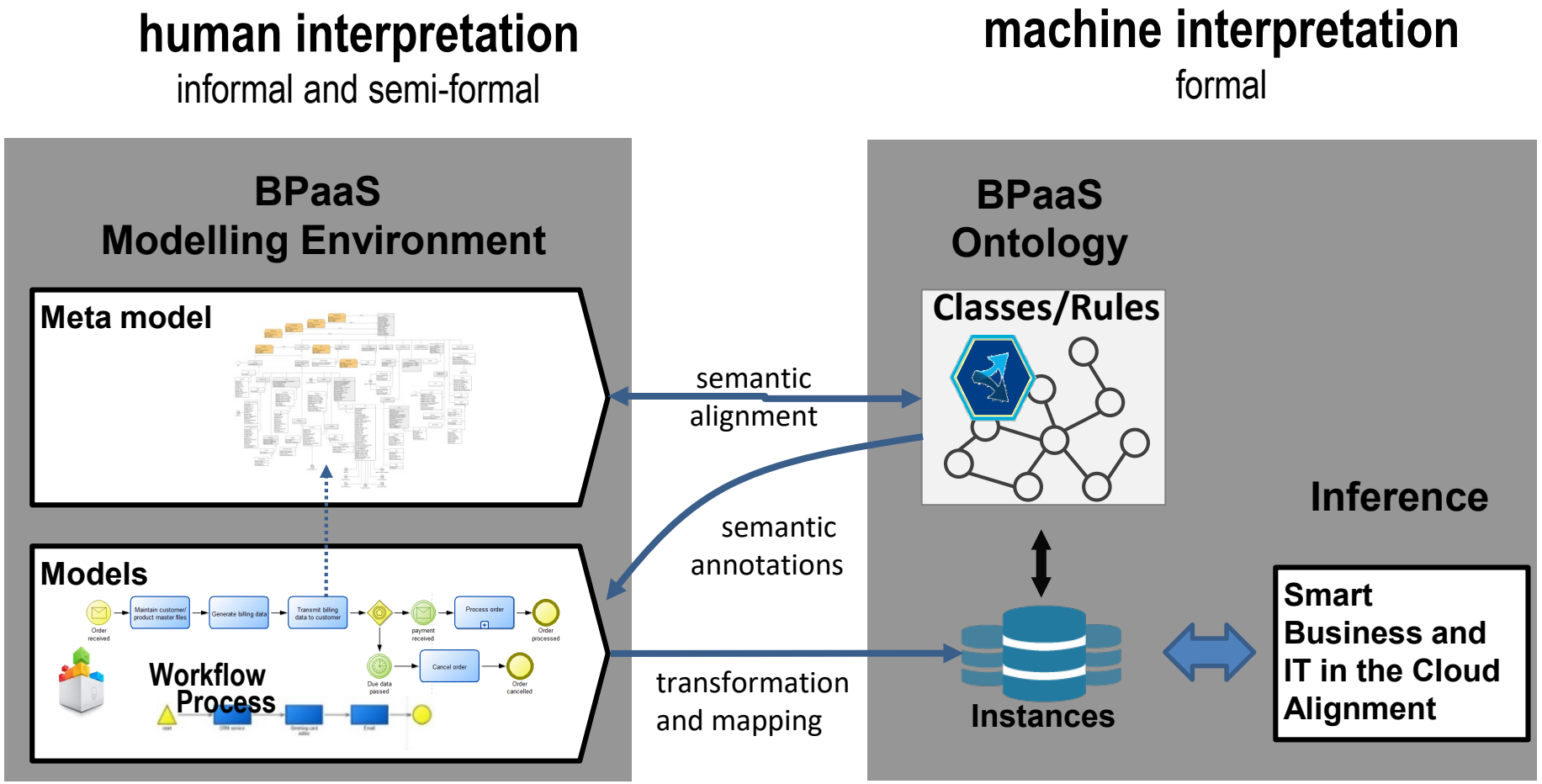
Semantic Lifting

- Map knowledge of models into an ontology
  - Semantics: Classes of the metamodel are aligned with classes in the ontology
  - Interpretation: For each element in a model an instance of the ontology is created
  - Content: Model elements are annotated with domain knowledge from ontology
  - Inference of the ontology can be applied to the knowledge base



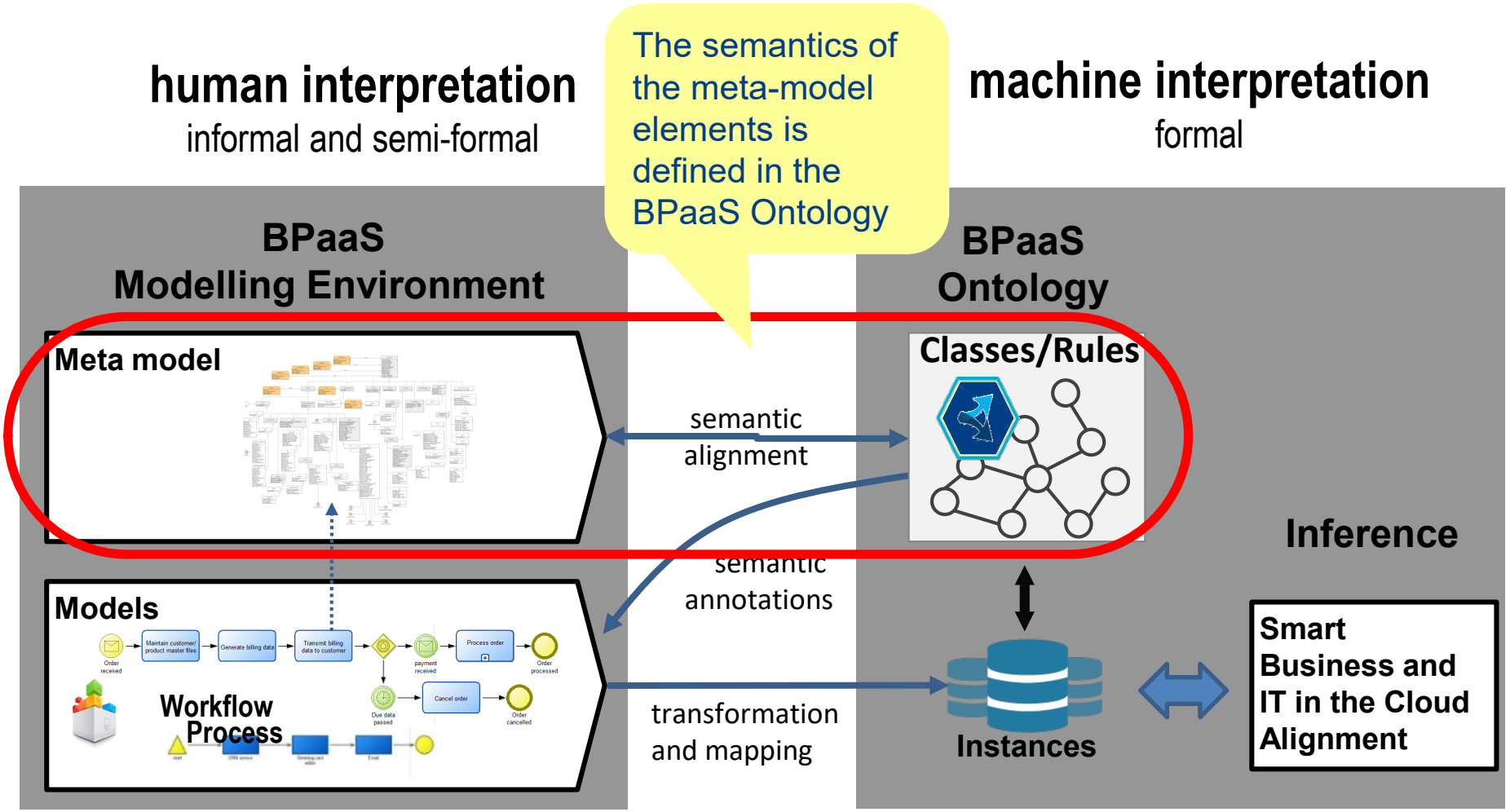
Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Example: Business Process as a Service



From: CoudSocket Project

Example: Business Process as a Service

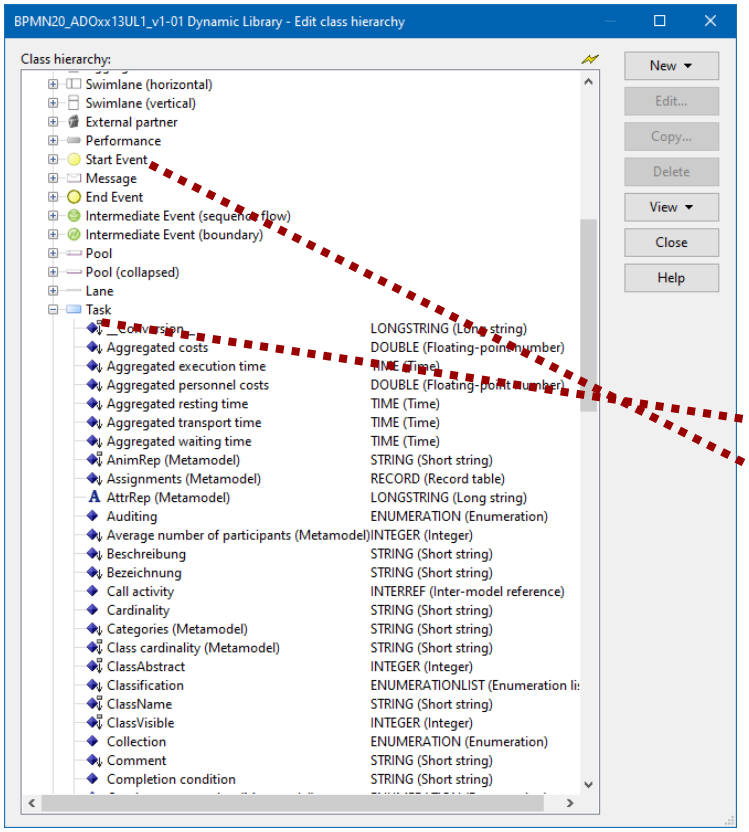


From: CoudSocket Project

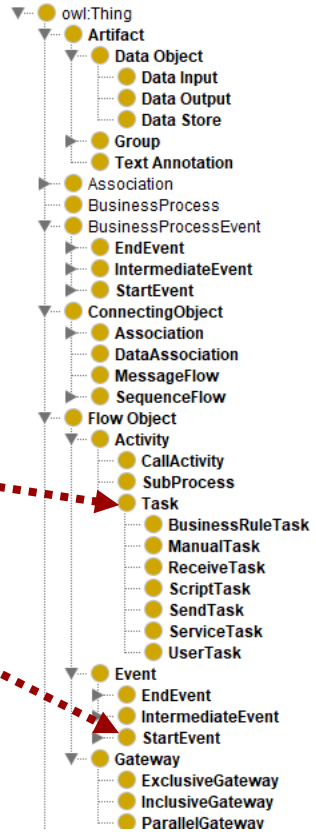
Semantic Alignment

The ontology contains classes for all modelling elements

BPMN Modelling Language in ADOxx



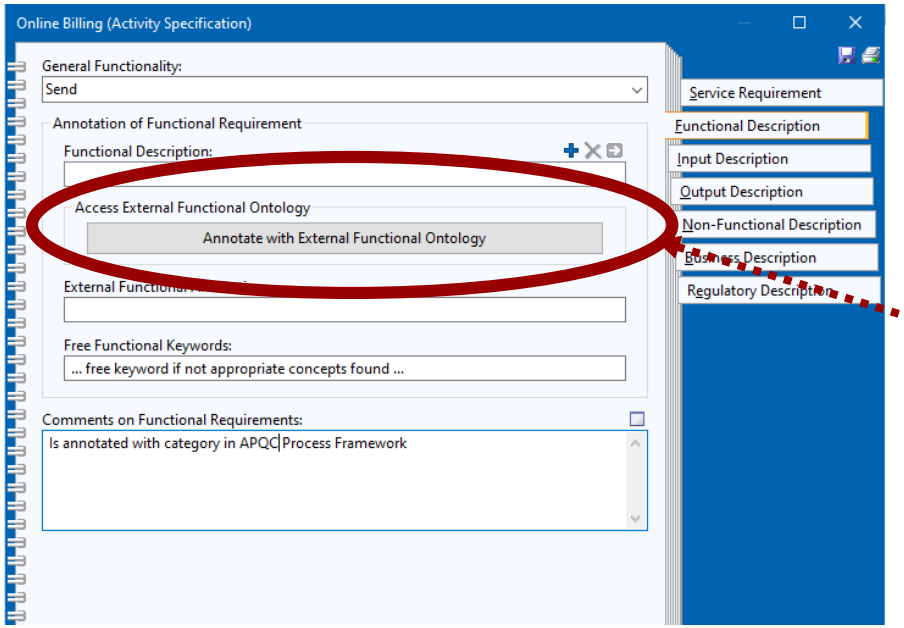
BPMN Ontology



Semantic Annotations

Annotate modeling elements with classes from the domain ontology

Example: Functionality of a Service

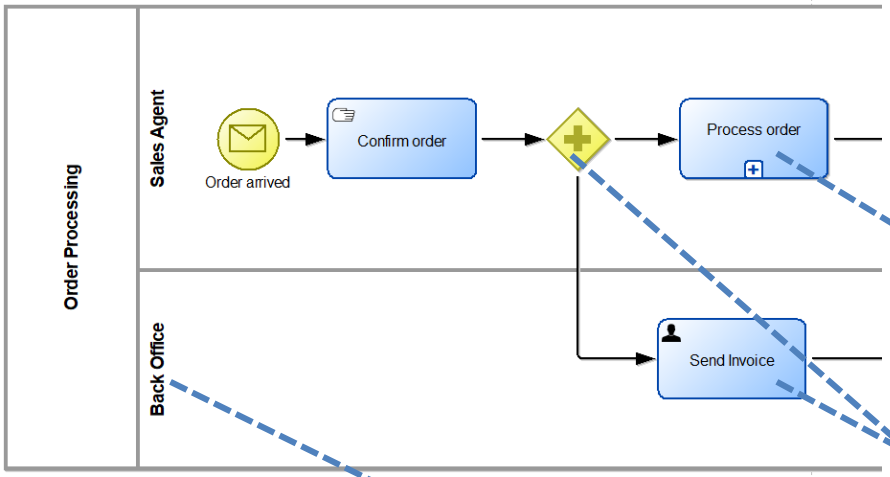


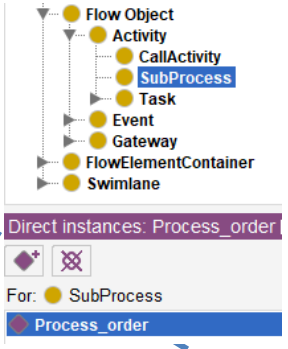
Ontology for APQC
Process Classification Framework

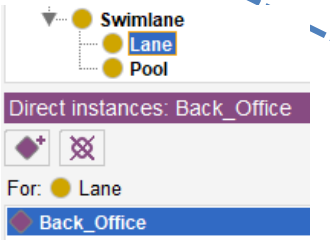
- owl:Thing
  - American Productivity and Quality Center
    - Acquire, Construct, and Manage Assets
    - Deliver Physical Products
    - Deliver Services
    - Develop and Manage Business Capabilities
    - Develop and Manage Human Capital
    - Develop and Manage Products and Services
    - Develop Vision and Strategy
    - Manage Customer Service
    - Manage Enterprise Risk, Compliance, Remediation, and Resiliency
    - Manage External Relationships
    - Manage Financial Resources
      - Manage fixed-asset project accounting
      - Manage internal controls
      - Manage international funds/consolidation
      - Manage taxes
      - Manage treasury operations
      - Perform general accounting and reporting
      - Perform global trade services
      - Perform planning and management accounting
      - Perform revenue accounting
        - Invoice customers
          - Generate customer billing data
          - Maintain customer/product master files
          - Post receivable entries
          - Resolve customer billing inquiries
          - Transmit billing data to customers
        - Manage and process adjustments/deductions
        - Manage and process collections
        - Process accounts receivable (AR)
        - Process customer credit
      - Process accounts payable and expense reimbursements
      - Process payroll
    - Manage Information Technology (IT)
    - Market and Sell Products and Services

Transformation and Mapping

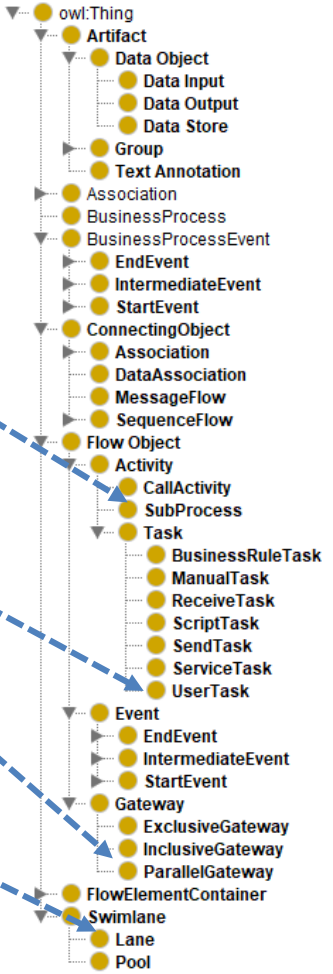
The model elements are exported as instances ontology classes







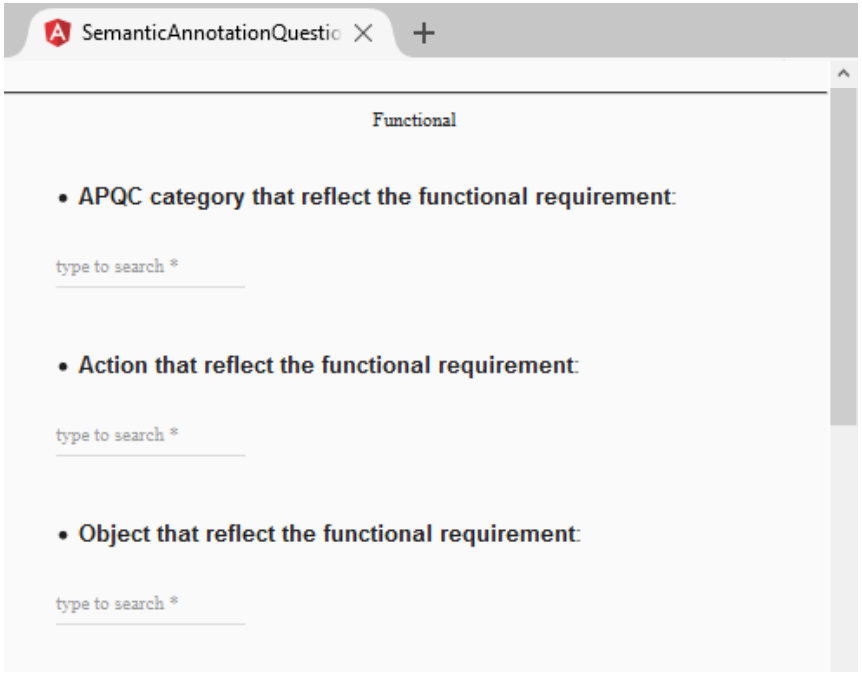
BPMN Ontology



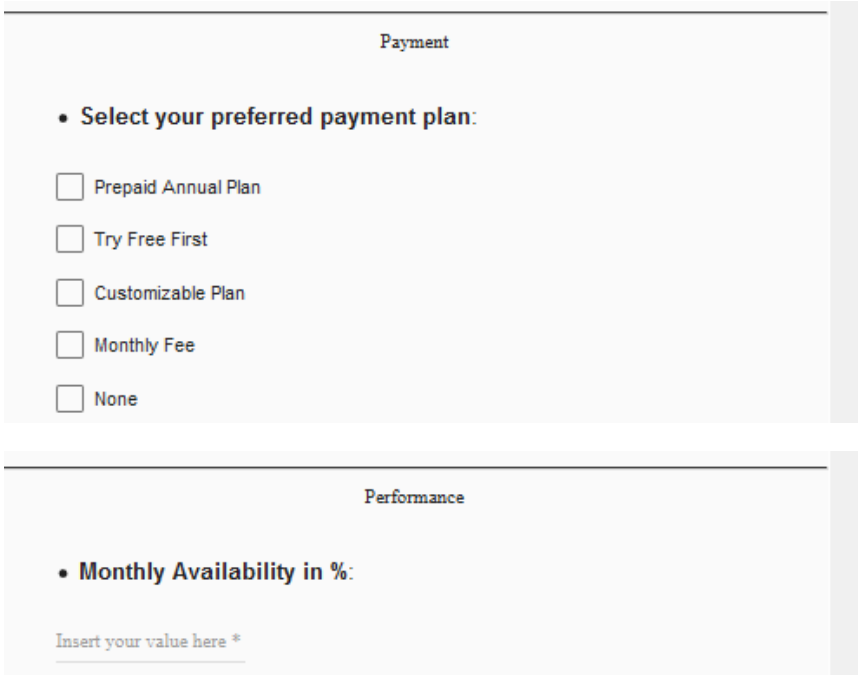
Inferencing: Cloud Service Selection

Cloud Service Selection

Functionality



Non-functional requirements



Thanks to Emanuele Laurenzi

Problems with Semantic Lifting

- Separate Environments for
  - Modelling
  - Knowledge Base (Inferencing)
- Inconsistency: Both metamodel and ontology must be aligned but are maintained independently:
  - Metamodel and ontology must represent the same semantics
  - Each change in metamodel must be reproduced in the ontology and vice versa
- Effort: After each change the model must be translated again into the ontology instances

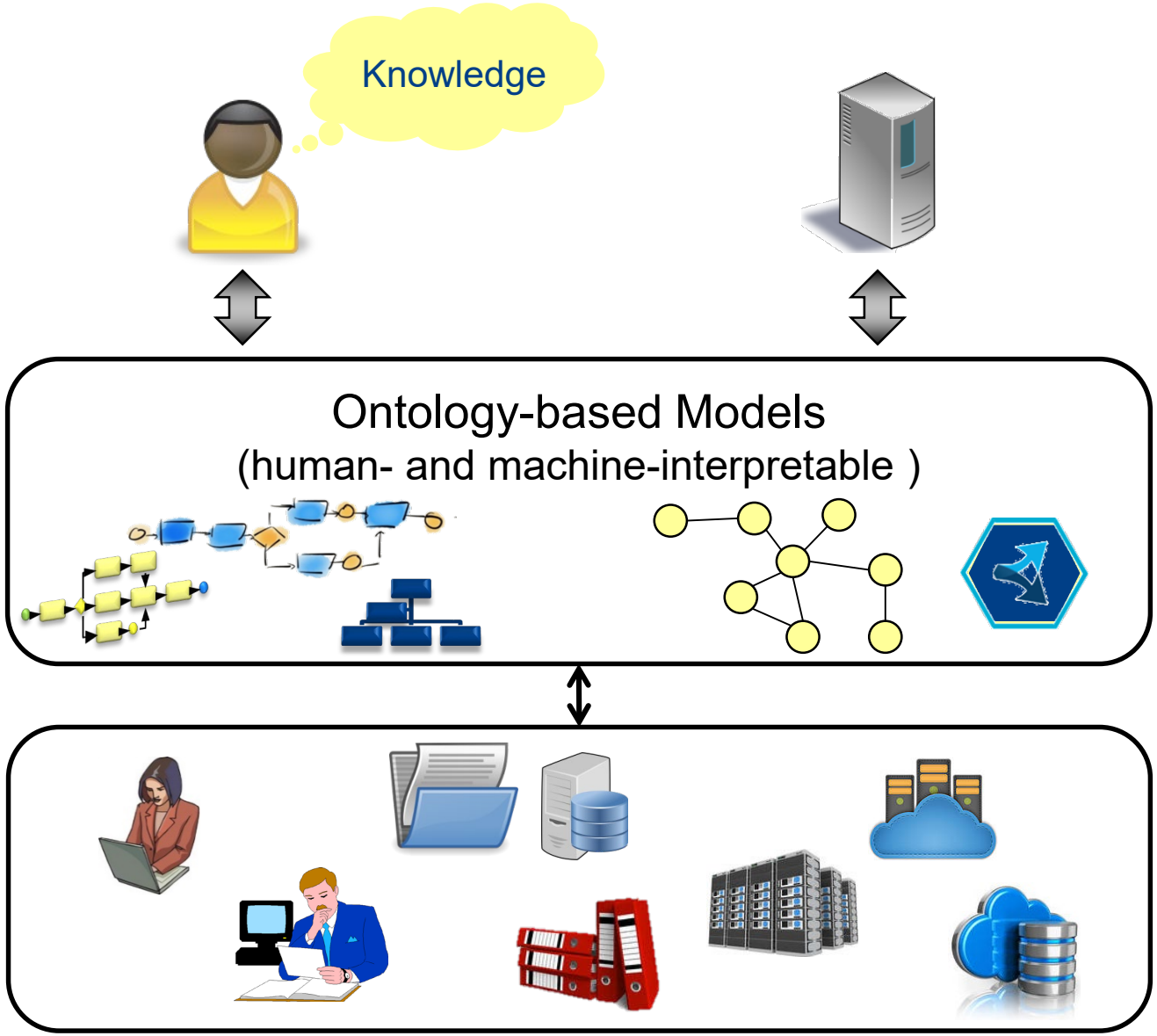
Ontology-based Metamodelling

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Objective

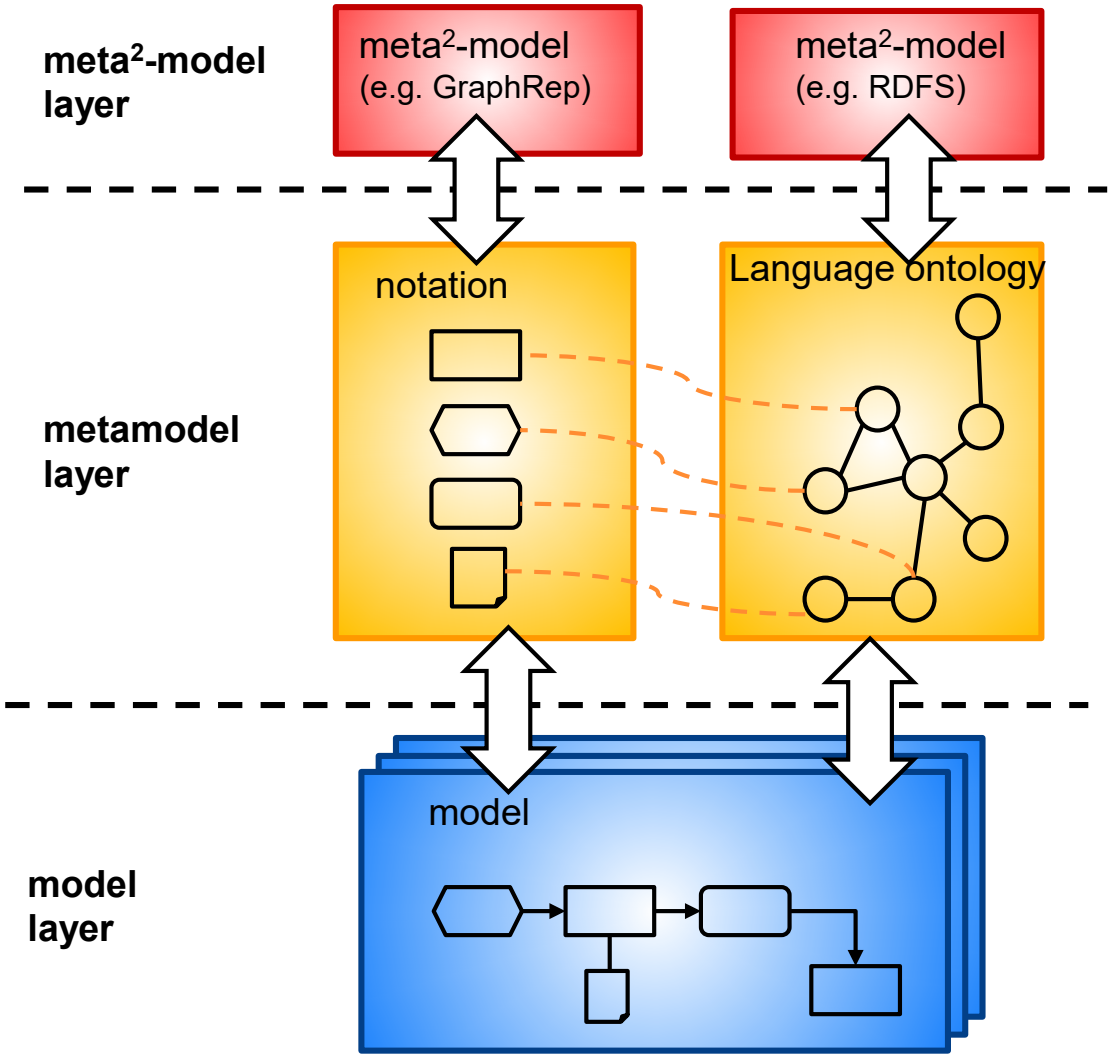
Models + Knowledge

Reality



Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

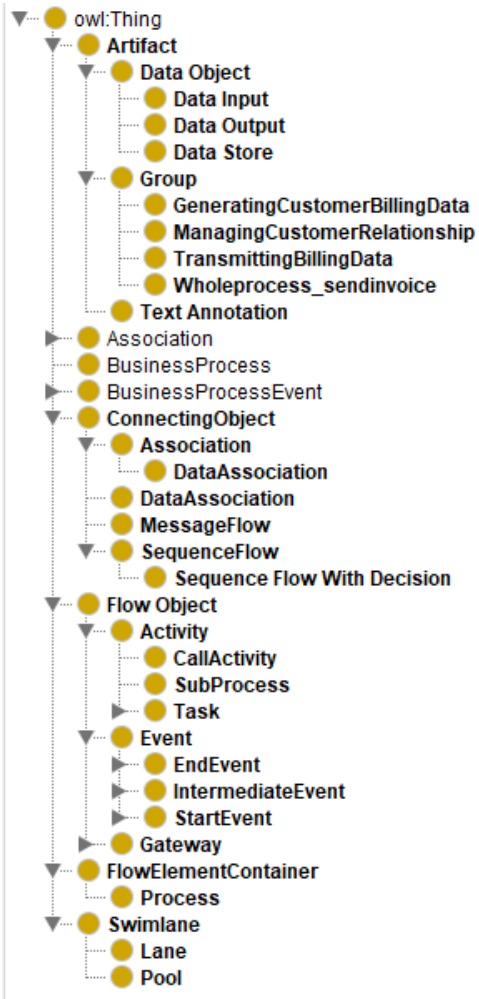
Ontology-based Metamodeling (1): Metamodel is represented as an Ontology



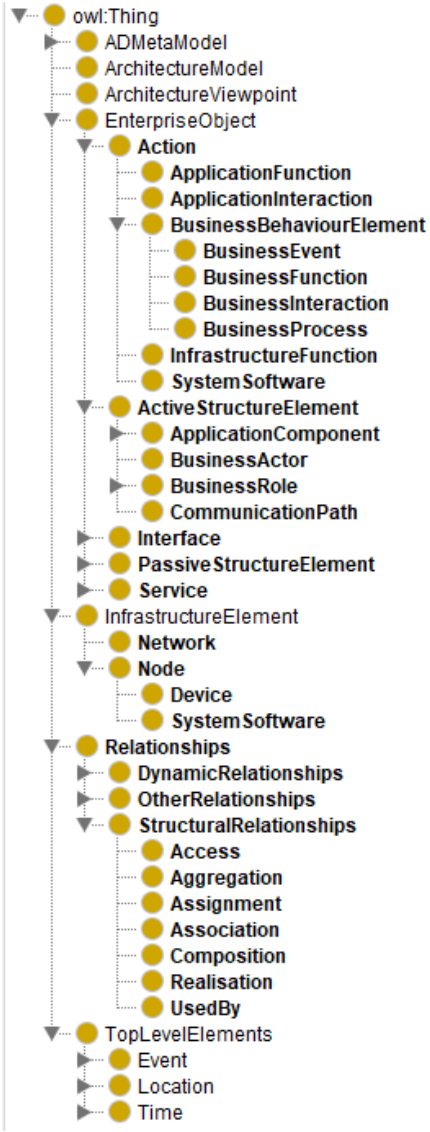
Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

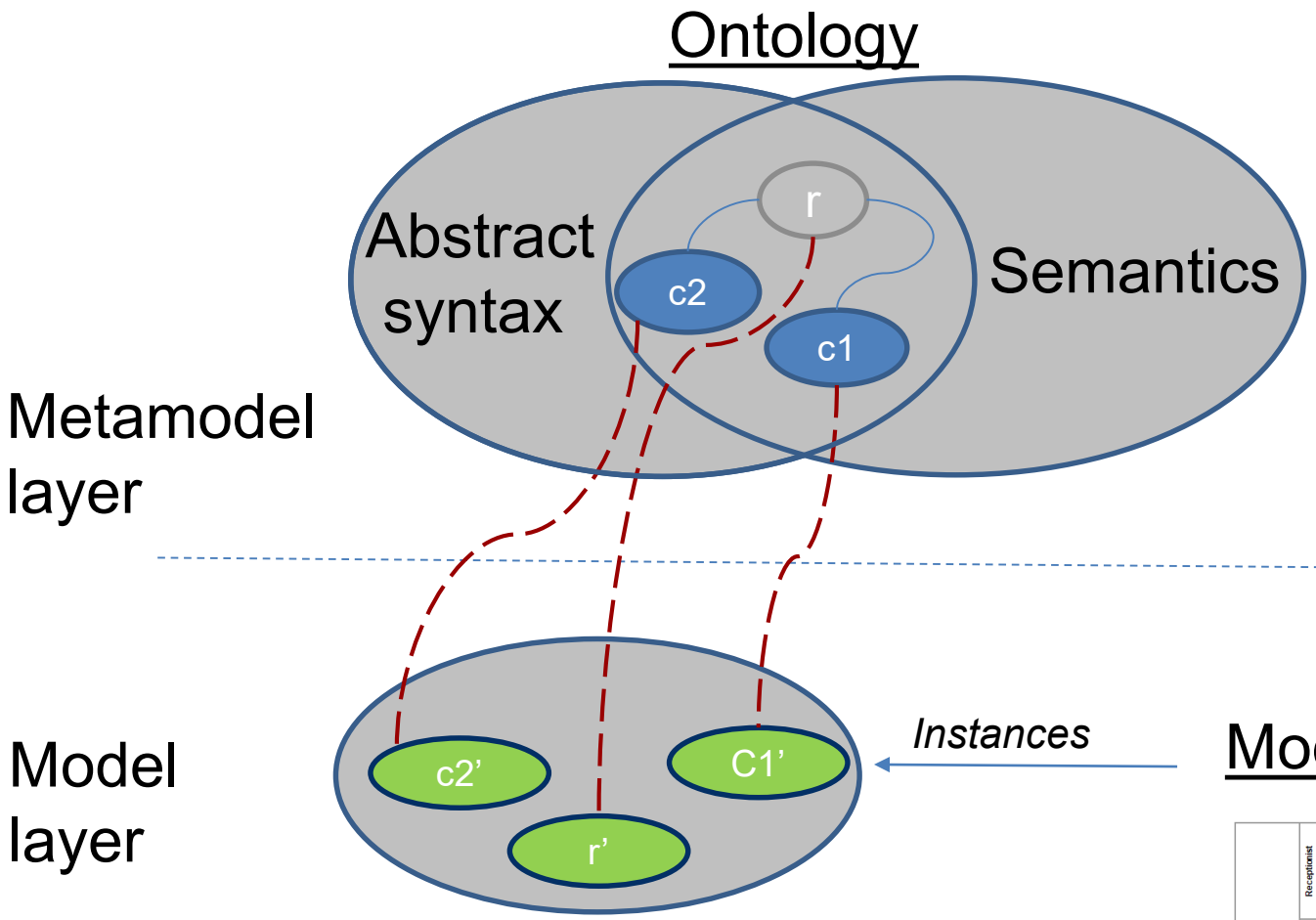
Modelling Language Ontologies

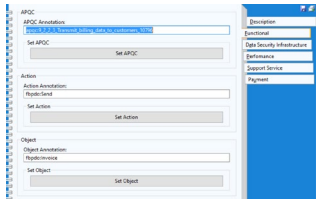
BPMN

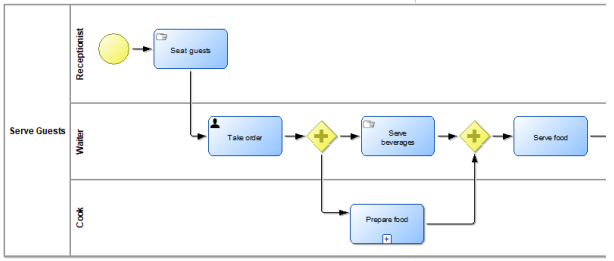


Archimate



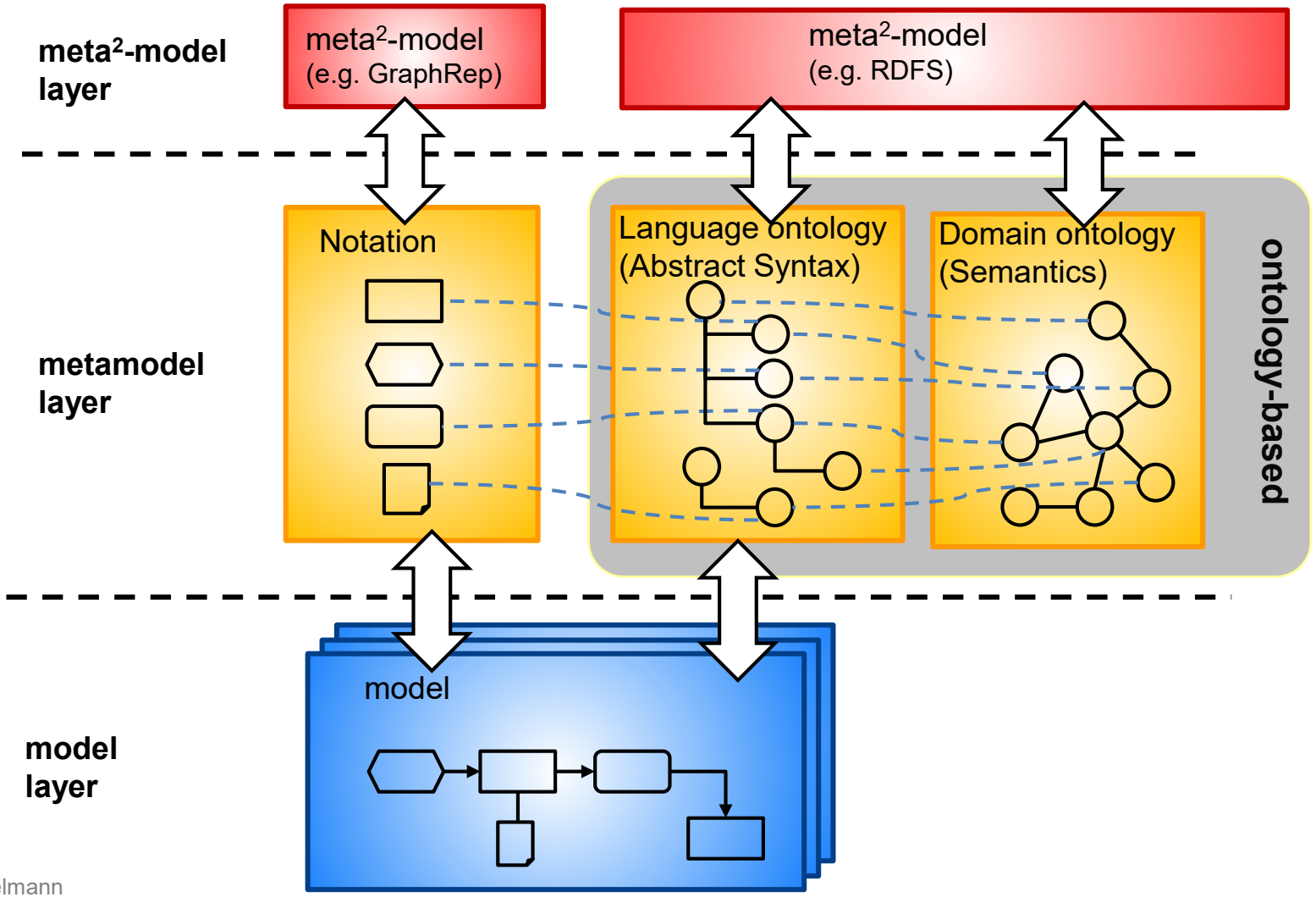






Thanks to Emanuele Laurenzi

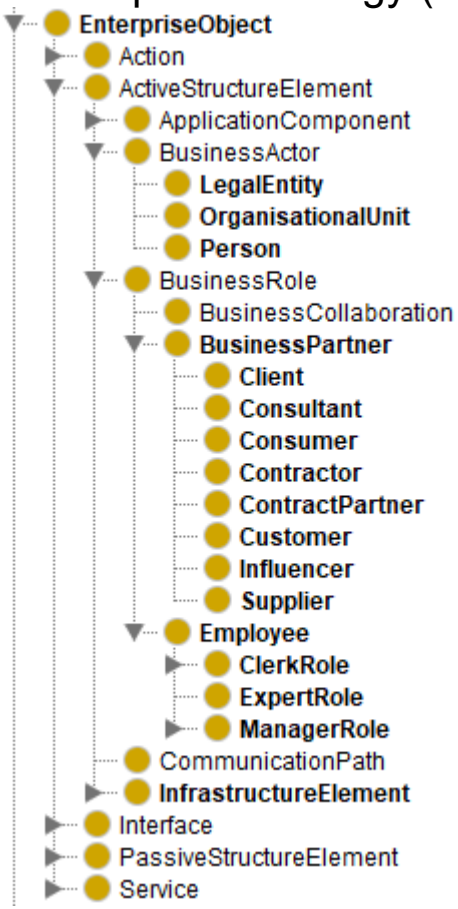
Ontology-based Metamodeling (2): Ontologies for Metamodel and Content



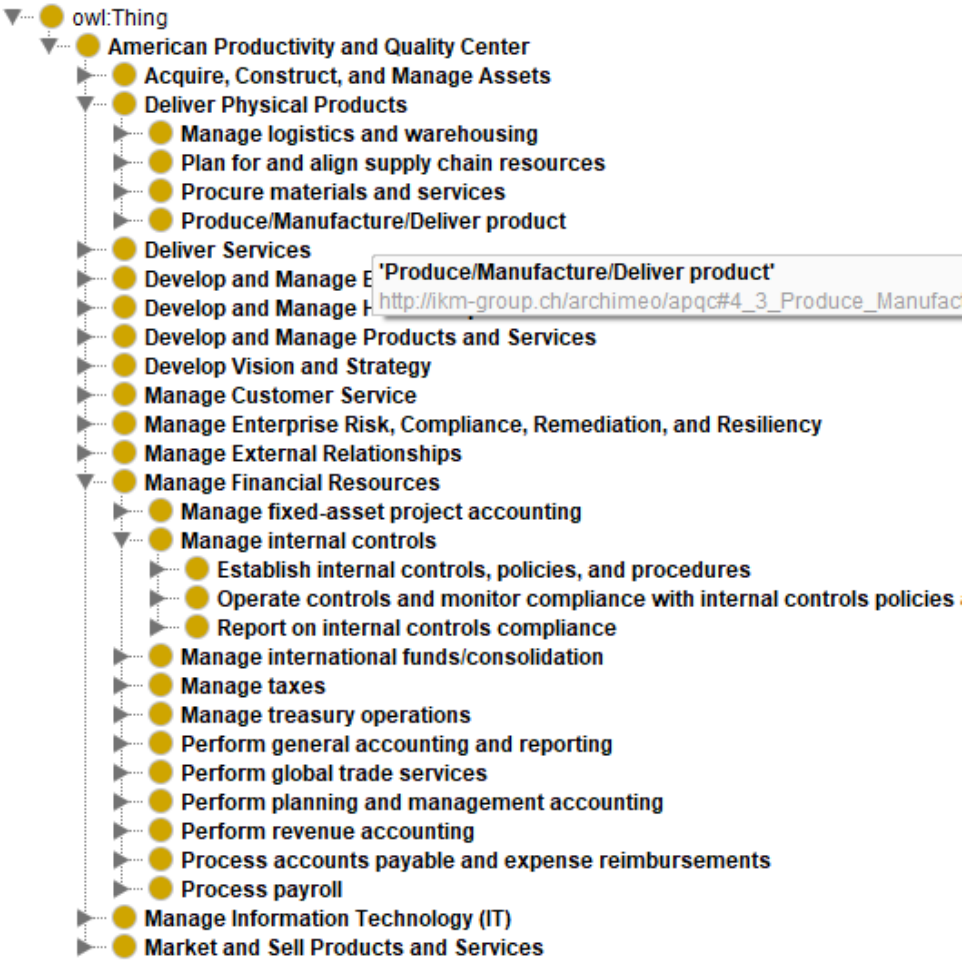
Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Domain Ontologies

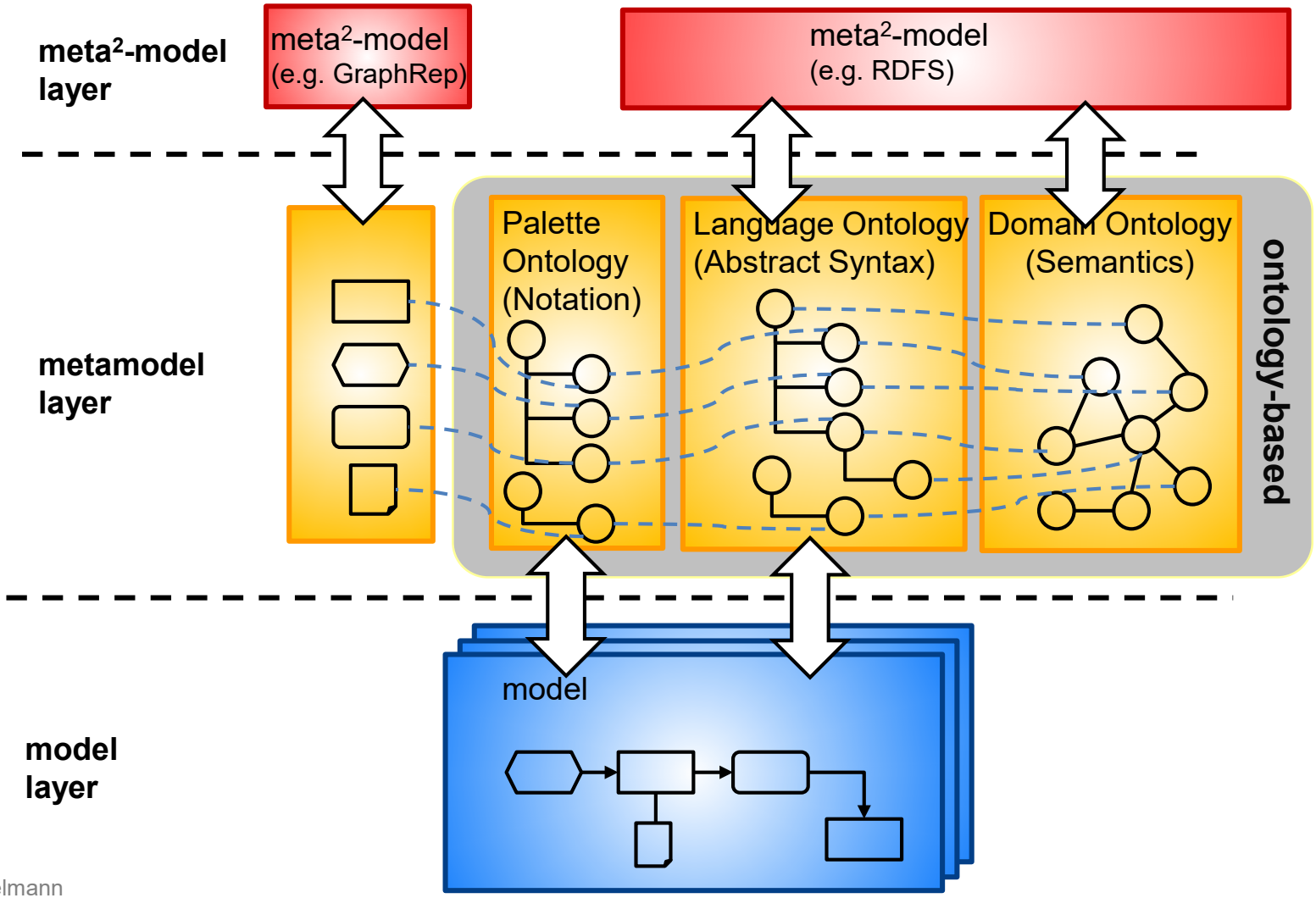
Enterprise Ontology (excerpt)



APQC Process Classification Framework

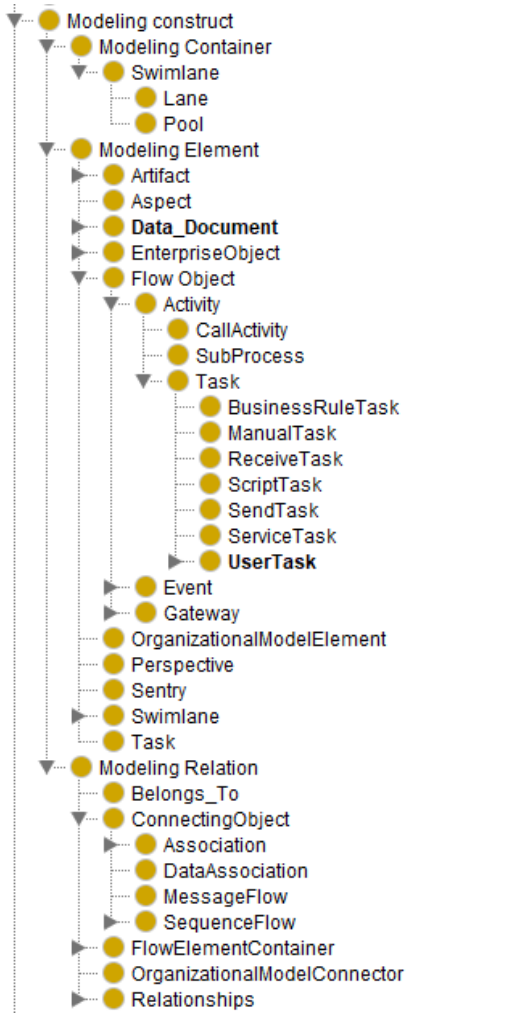


Ontology-based Metamodeling (3): Ontologies for Language, Metamodel and Content



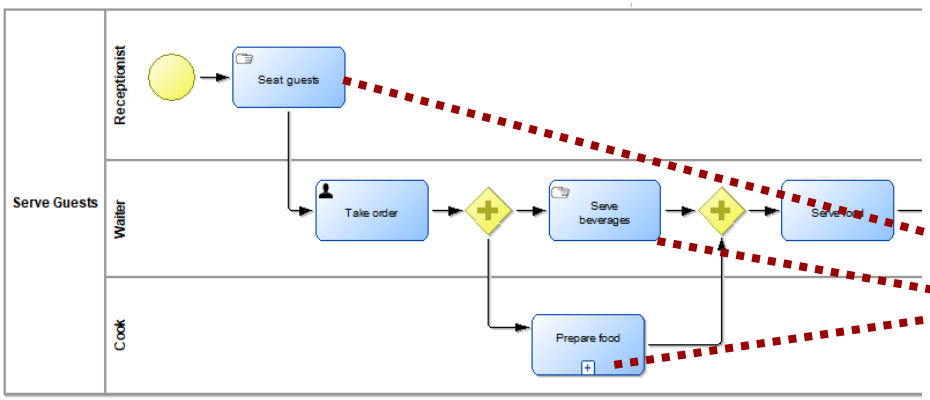
Palette Ontology

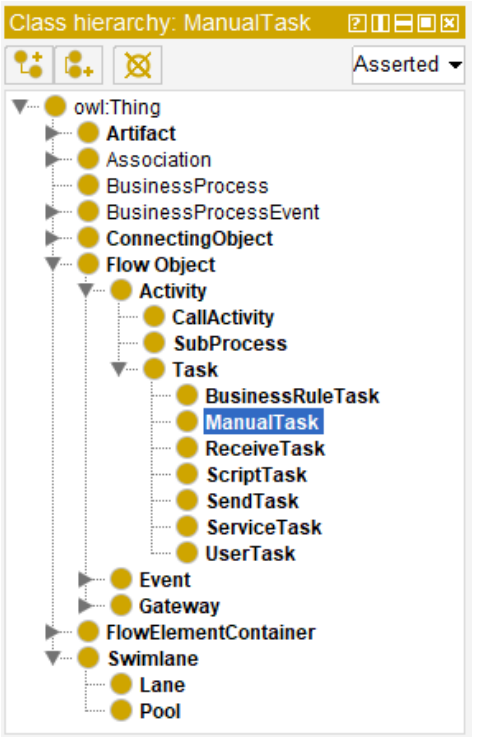
Palette Ontology (excerpt)



Ontology-Based Metamodel

- Single environment for modelling and ontology
- Model elements are directly created as instances in the ontology





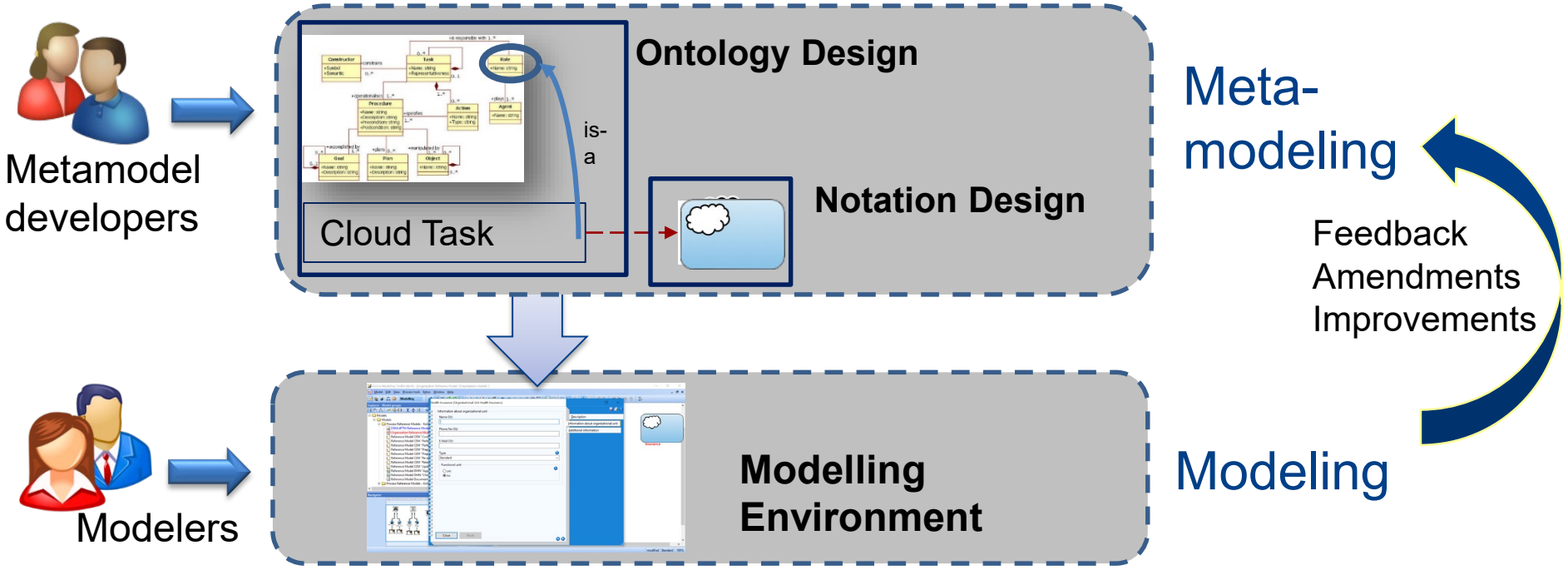
Agile Modelling

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Objective

Ensure a precise shared interpretation of new modeling constructs to both **humans and machines**

Problem: Separation between Metamodel Developer and Modeler



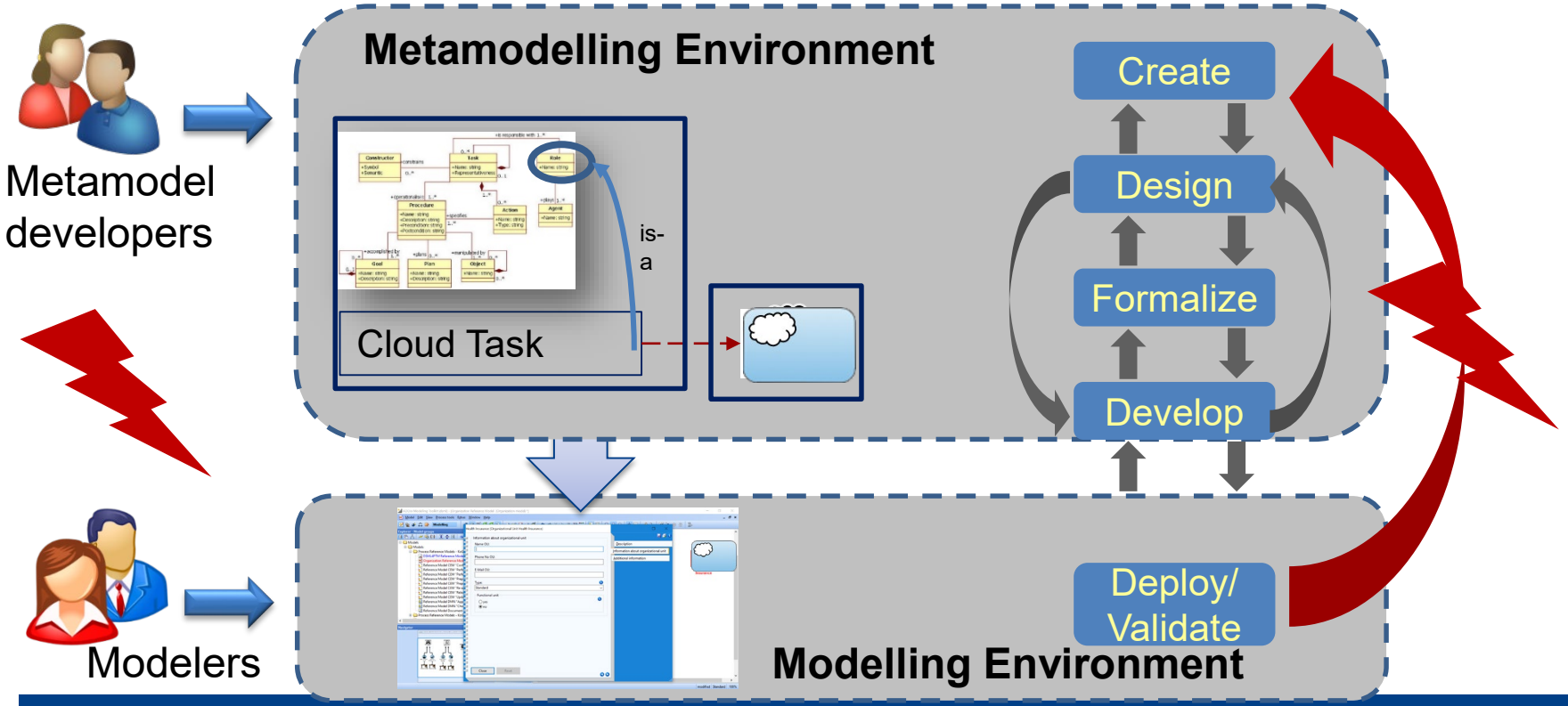
Challenge: Separation of metamodelling and modelling
(typically in separate environments, e.g. ADOxx Development and ADOxx Modelling Toolkits)

Objective: Integrate metamodeling and modeling in a single environment

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Thanks to Emanuele Laurenzi

Problem: Separation between Metamodel Developer and Modeler



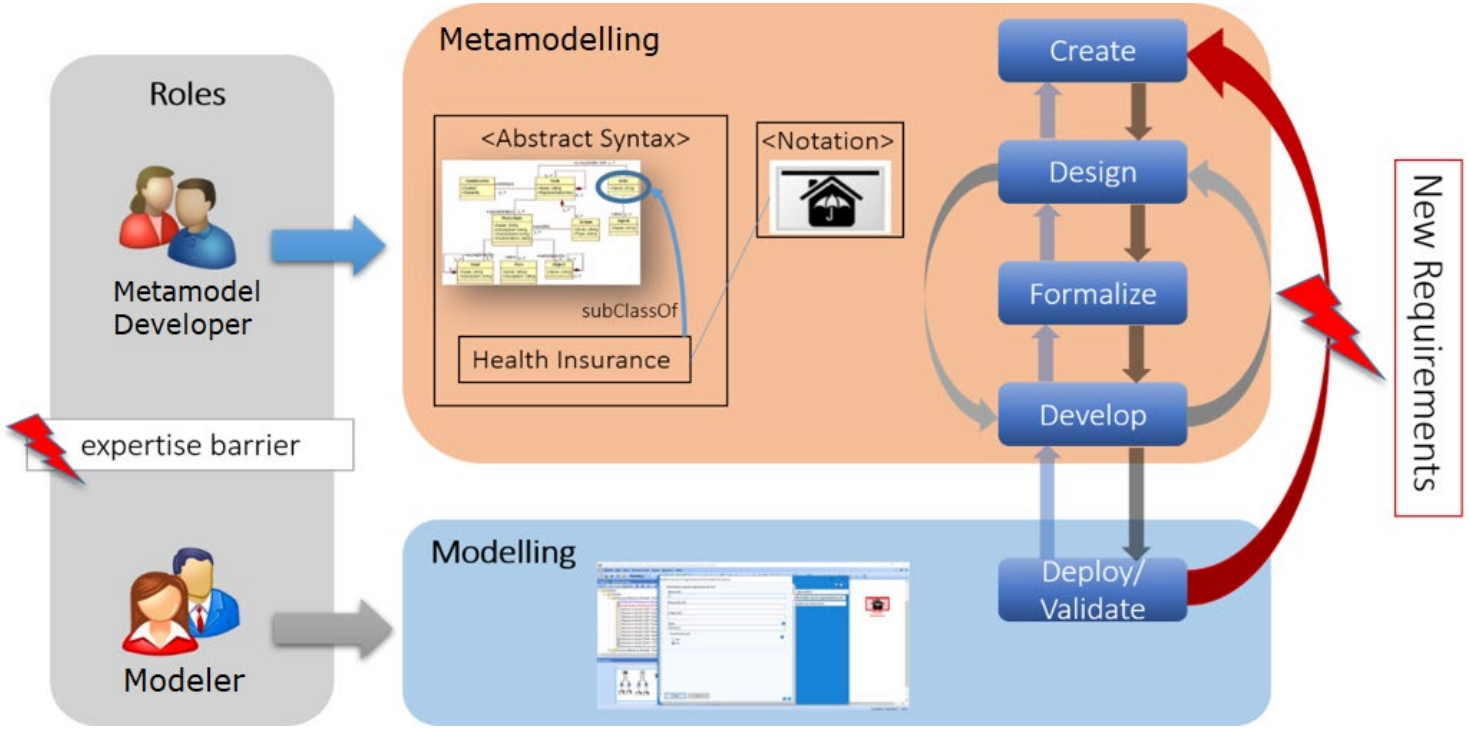
Challenge: Separation of the metamodelling and metamodelling
(typically in separate environments, e.g. ADOxx Development and ADOxx Modelling Toolkits)

Objective: Integrate metamodeling and modeling in a single environment

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Thanks to Emanuele Laurenzi

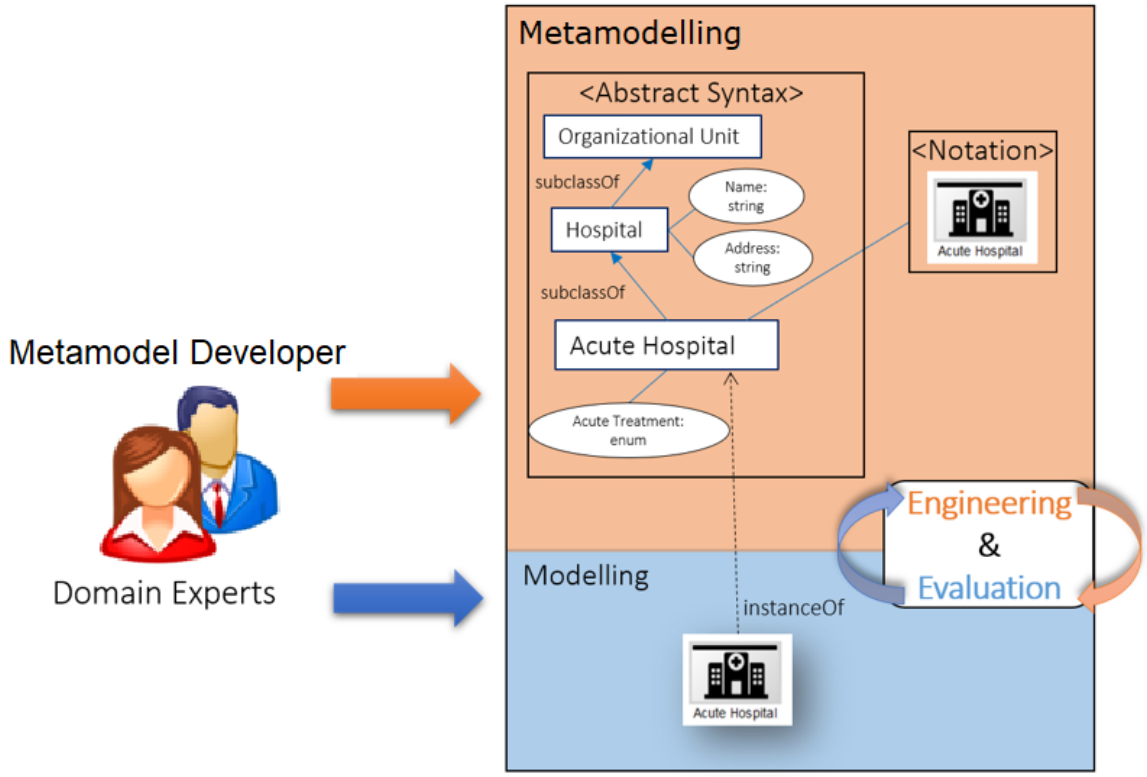
Challenge: Separation of metamodelling and modelling



- Challenge 1: Metamodeling is a joint effort between metamodel experts and domain experts
- Challenge 2: Sequentialization of metamodeling and modeling is time consuming

Integration Modeling and Metamodeling in a Single Environment

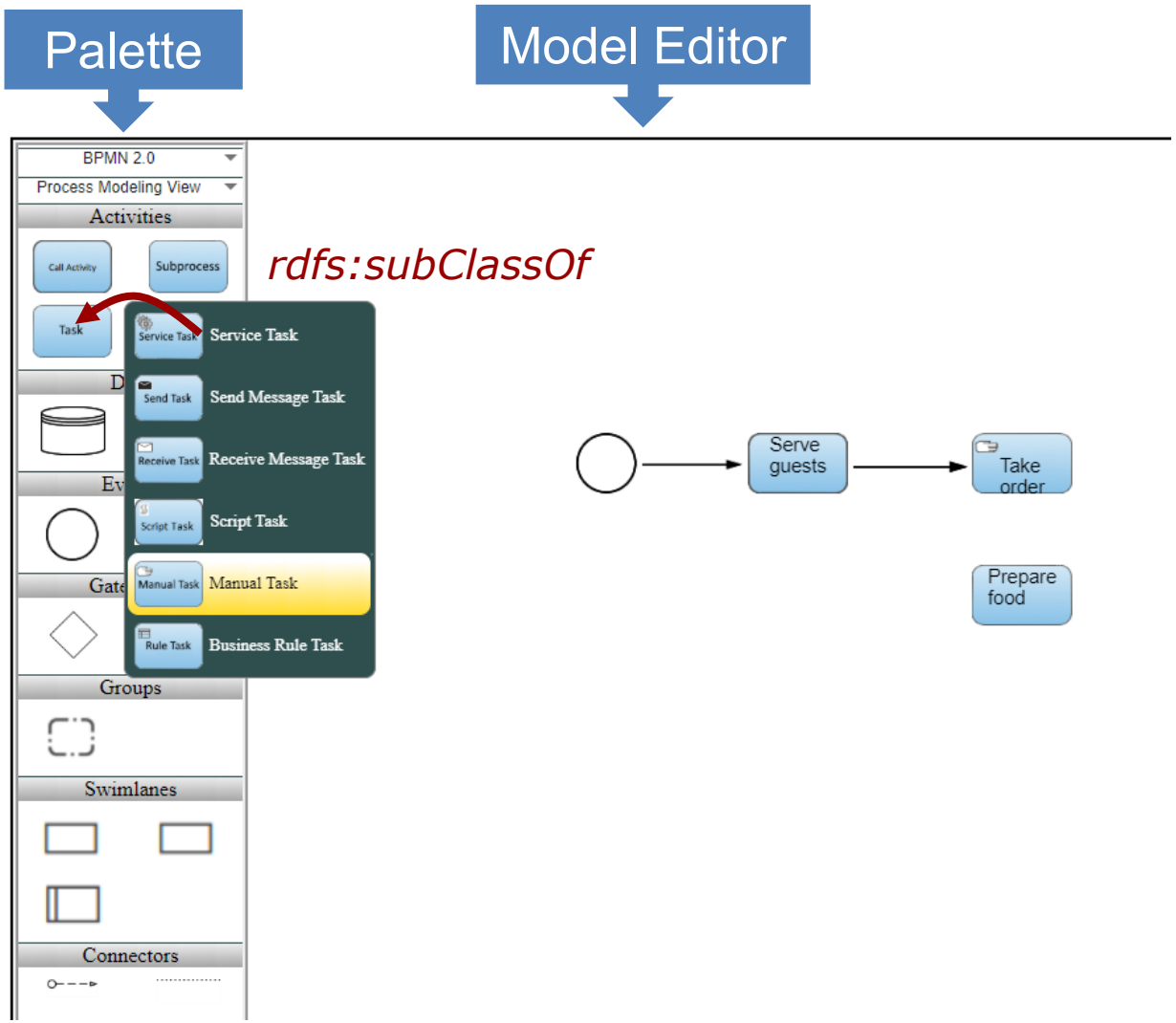
- Tight collaboration between metamodel developer and modeler
- Modeler can also take the role of metamodel developer



AOAME:
Agile and Ontology-Aided Modeling Environment

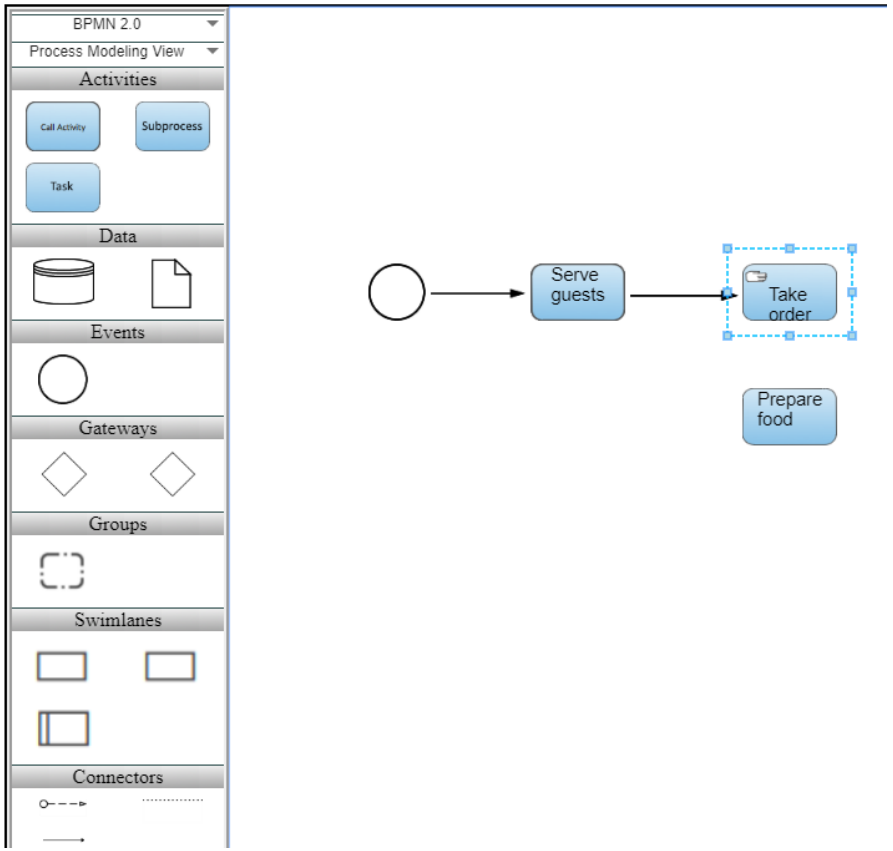
- AOAME is a a prototypical implementation for Agile and Ontology-Aided Modeling
- It is based on the PhD Thesis of Emanuele Laurenzi
- Implementation of the current version by
  - Emanuele Laurenzi
  - Charuta Pande
  - Devid Montecchiari

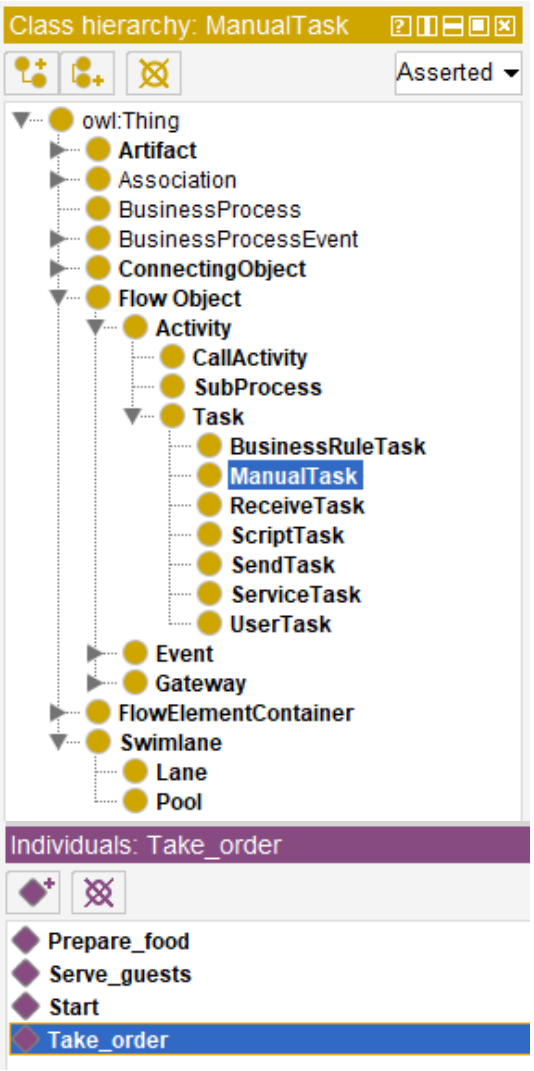
Ontology-Based Modeling in AOAME



Ontology-Based Modelling

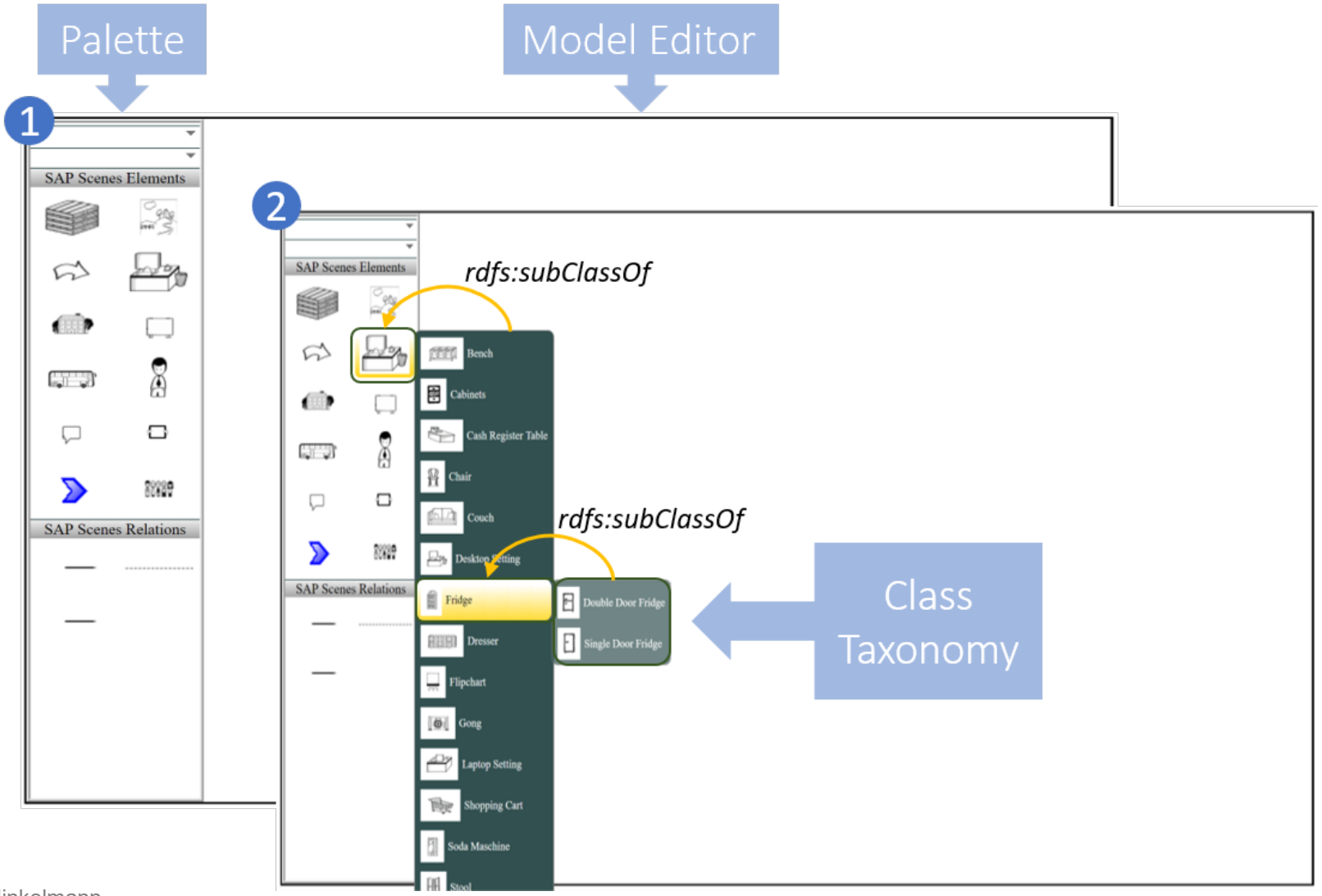
Model elements are directly created as instances in the ontology
Modelling and ontology in a single environment



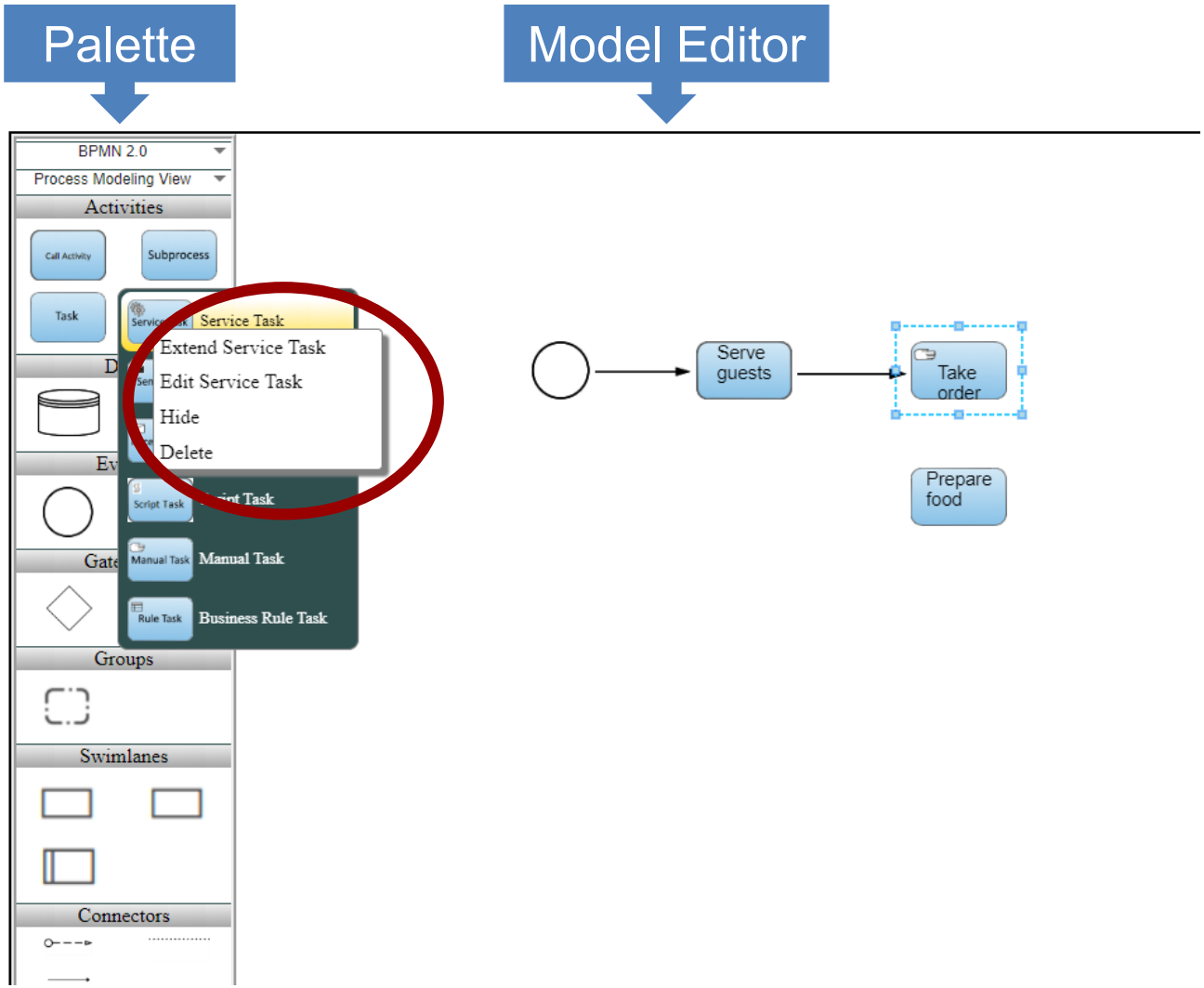


Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Modeling Elements are represented in a Class Hierarchy



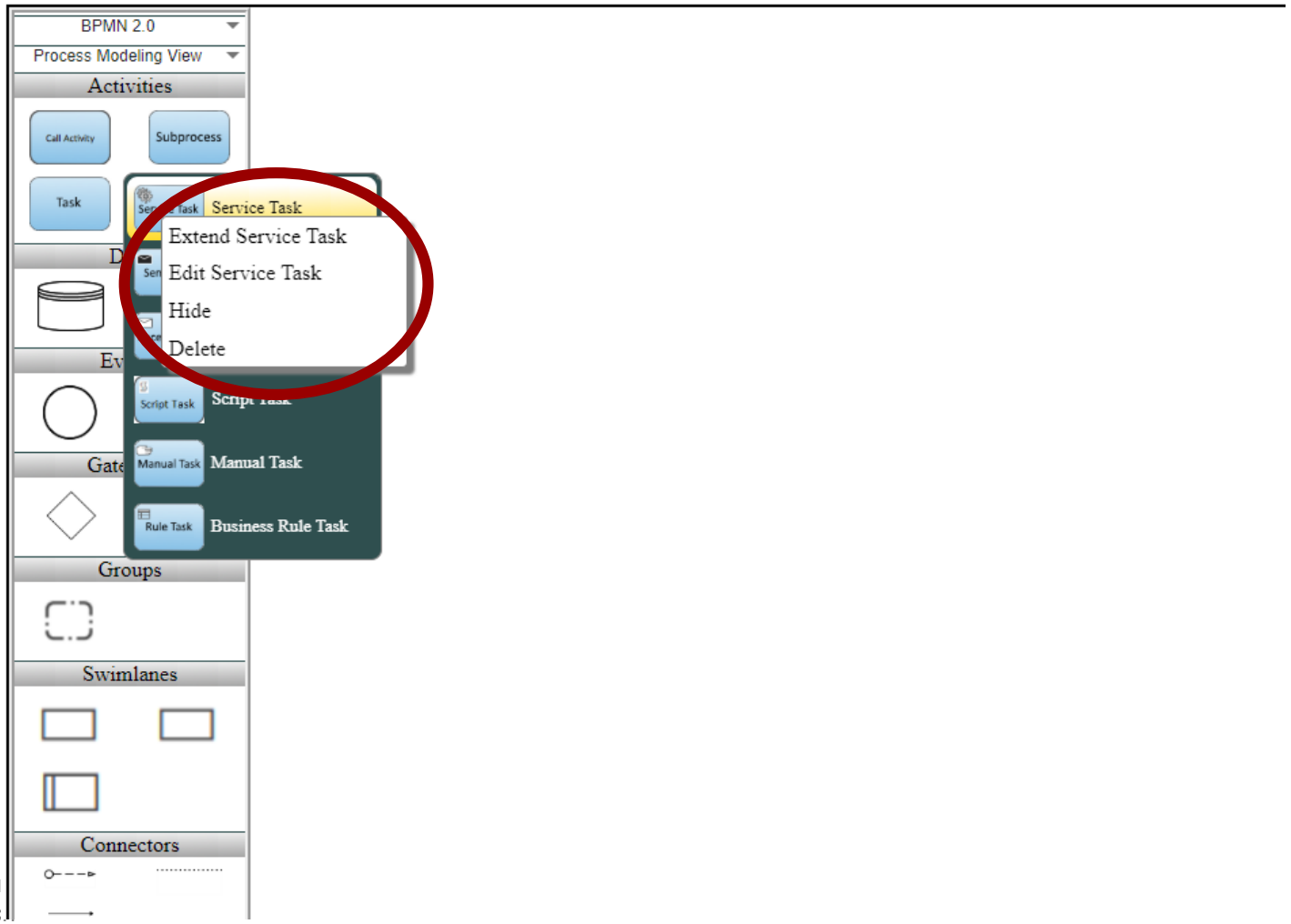
Extending AOAME Modeling Languages – on the fly



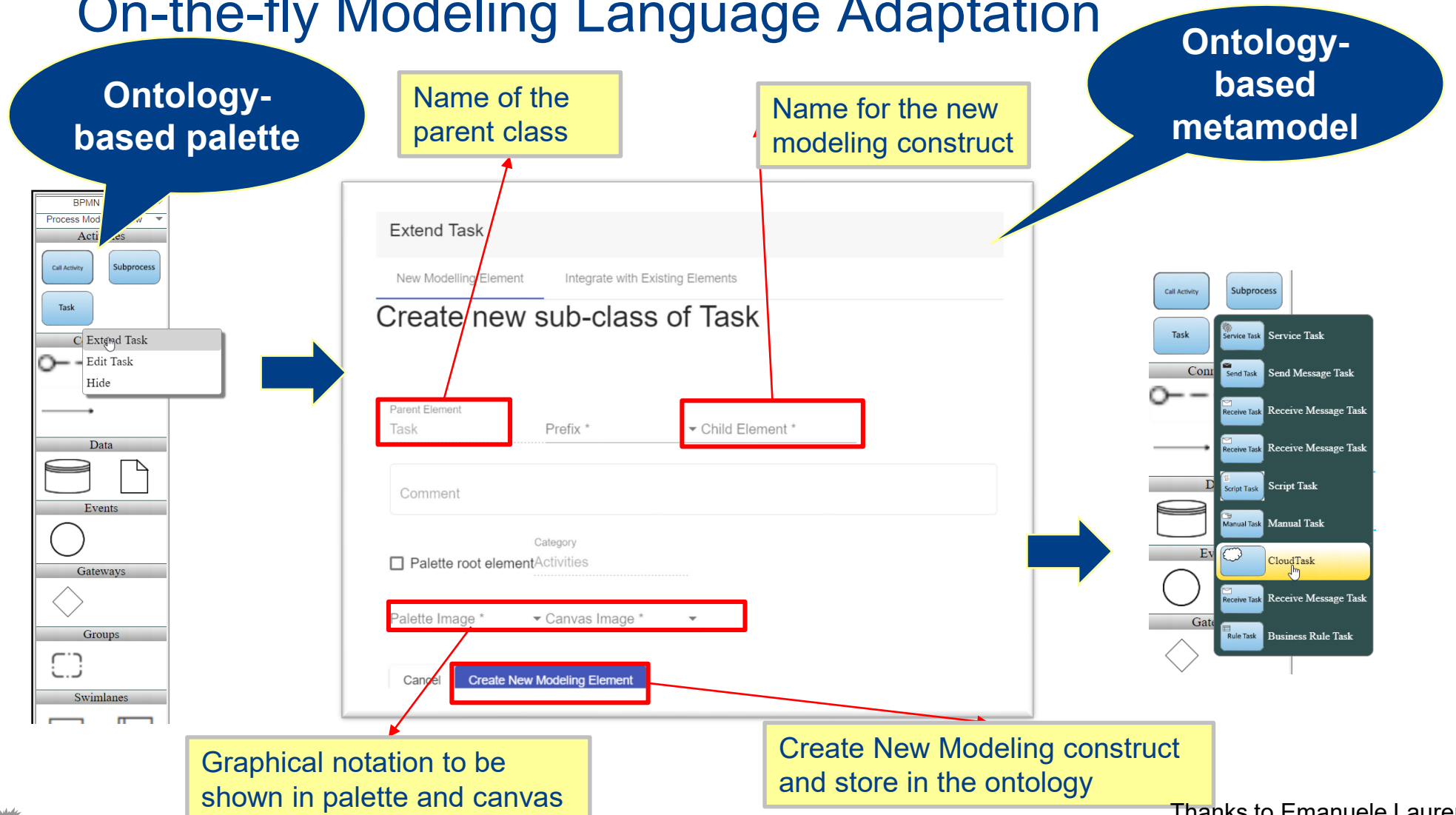
Extending AOAME Modeling Languages

Palette

Model Editor



Integration of Meta-modeling and Modeling: On-the-fly Modeling Language Adaptation



Thanks to Emanuele Laurenzi

Hands-on
Agile and Ontology-Aided Modeling in AOAME