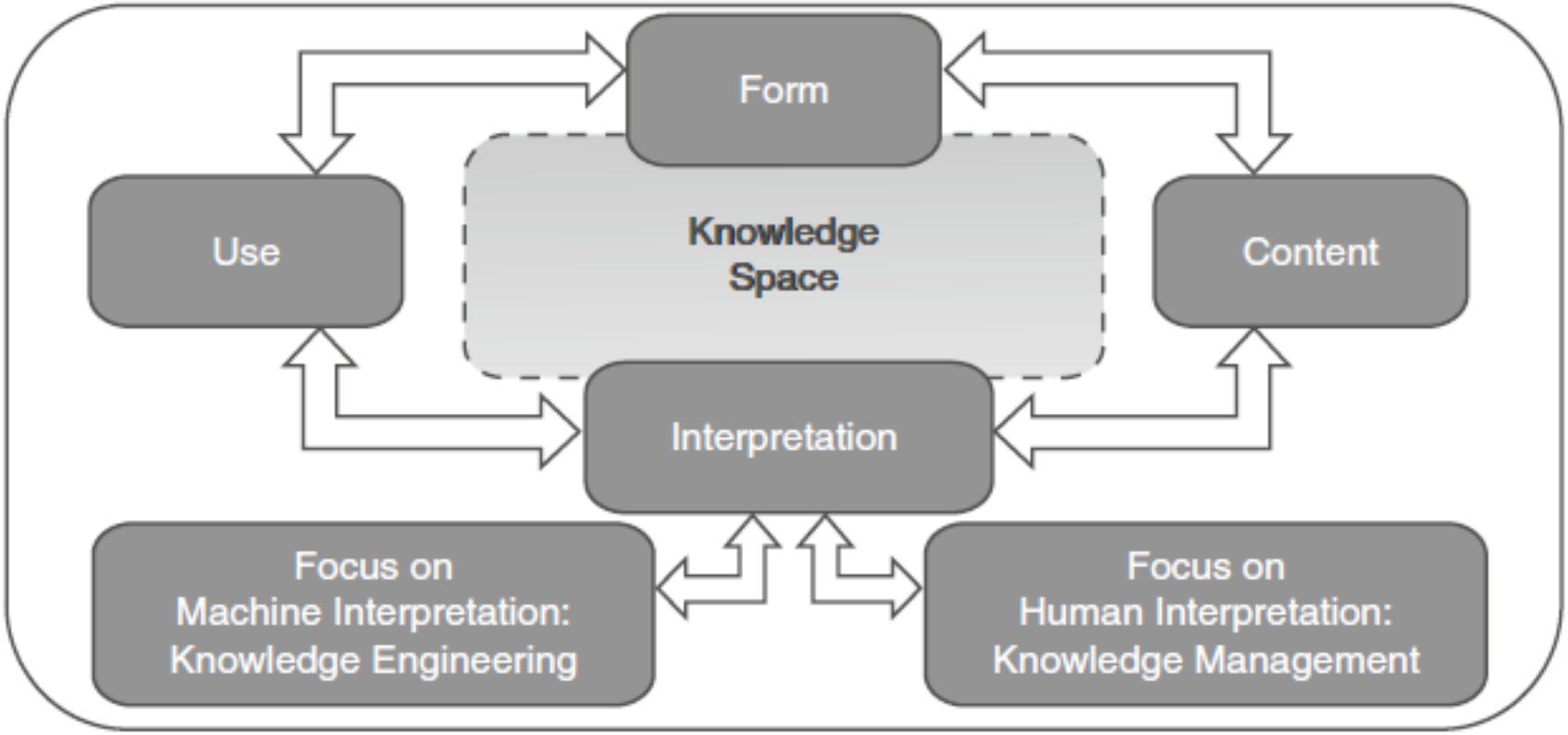
Ontology-based Modeling

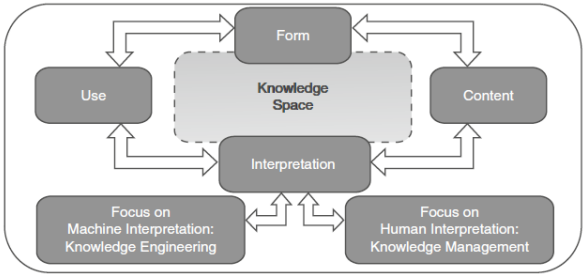
Knut Hinkelmann

Dimensions of a Knowledge Space



Karagiannis, D., & Woitsch, R. (2010). Knowledge Engineering in Business Process Management. In *Handbook on Business Process Management 2* (pp. 463–485). Springer.

Dimensions of the Knowledge Space



Use:

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

Form: modeling language

Content: Instantiation of concepts

- ***Use***: Stakeholders and their concerns determine the relevant subset of the knowledge
- ***Form***: Syntax and semantic of *meta model concepts*.
- ***Content***: *Instantiation* of meta model concepts for a specific *application* (represented in the labels)
- ***Interpretation***: Giving meaning to a model:
  - Graphical models are cognitively adequate for human
  - Machines need more formal representation

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

Content: Instantiation of Meta model + Application knowledge

- Humans «know» the meaning of the modeling objects.
  - Meta model: Concepts of the model language
  - Application: Labels/names of the model elements
- Examples:

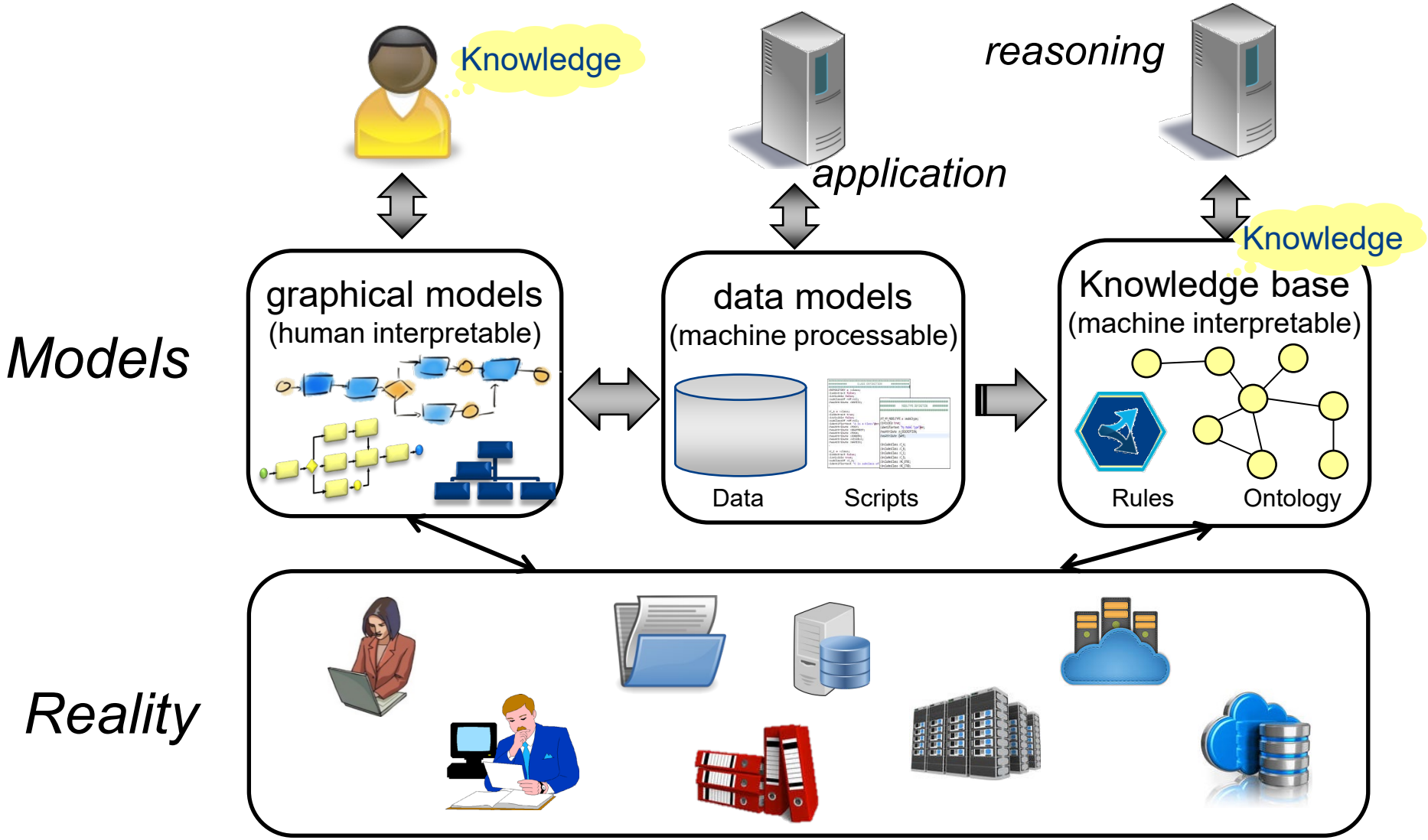
  - Meta model: Application Component
  - Application: «ERP System» is business software

  - Meta model: Task
  - Application: «Cook pasta» is 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

Semantic Lifting

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

Semantic Lifting: Map Models into an Ontology

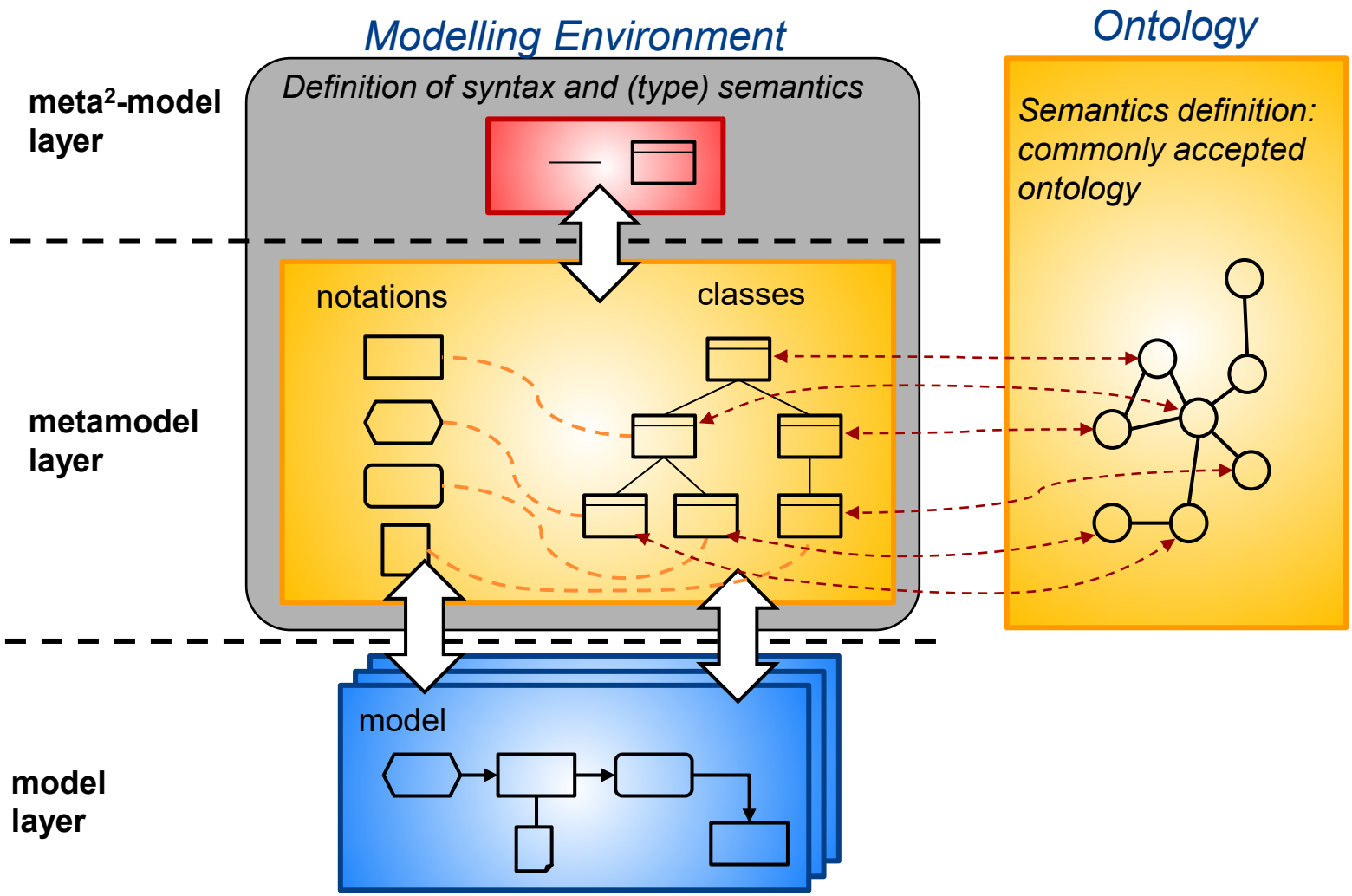


Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Semantic Lifting: Representing Content as Ontology

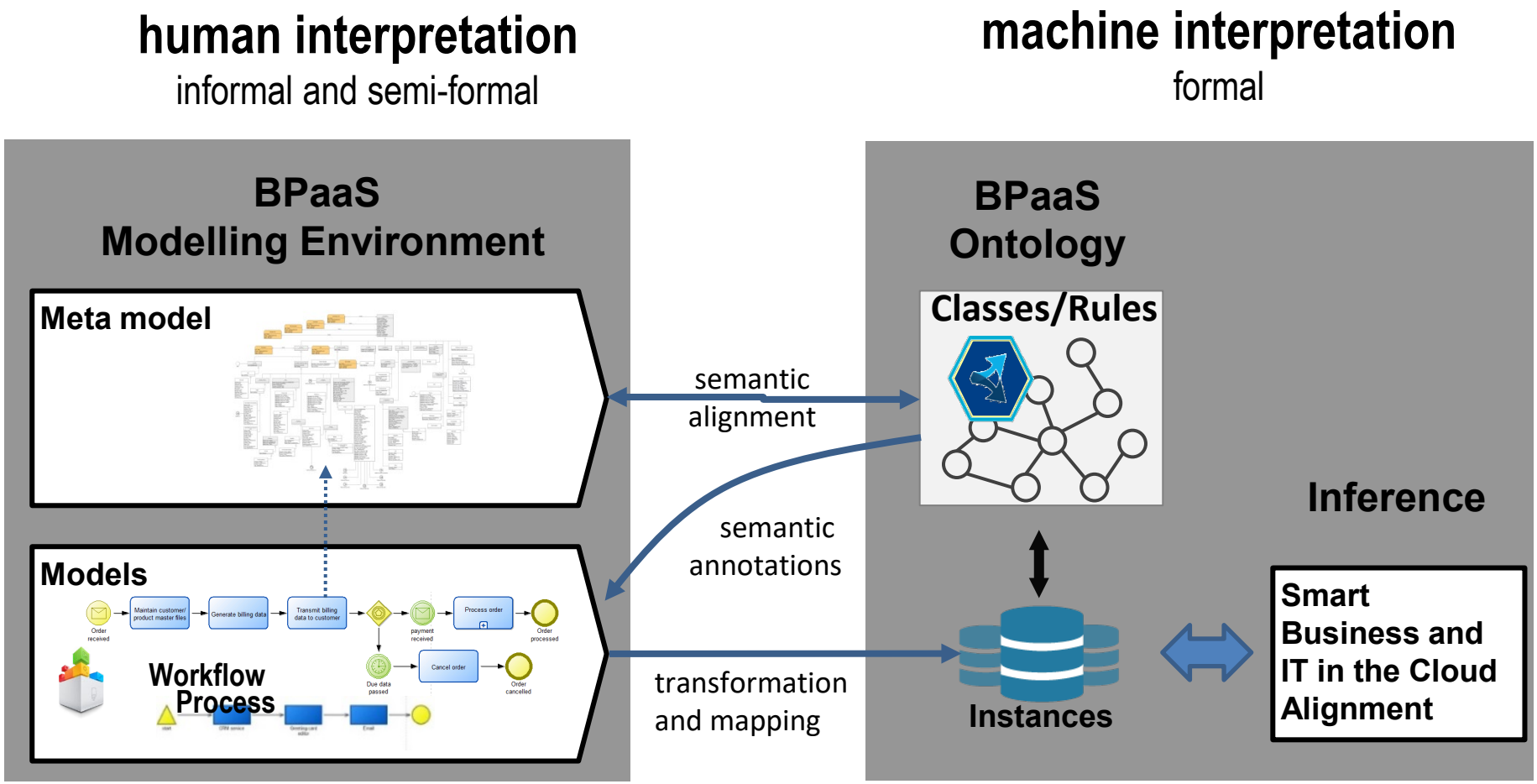
- **Meta model Knowledge**:
  - Concepts of the meta model have corresponding class in an ontology
  - For each element in a model an instance of the corresponding ontology class is created
- **Knowledge about application domain**:
  - Model elements are annotated with domain knowledge from application domain ontology
- Ontology reasoning can be applied to the content knowledge in the models

Semantic Lifting: Map Models into an Ontology



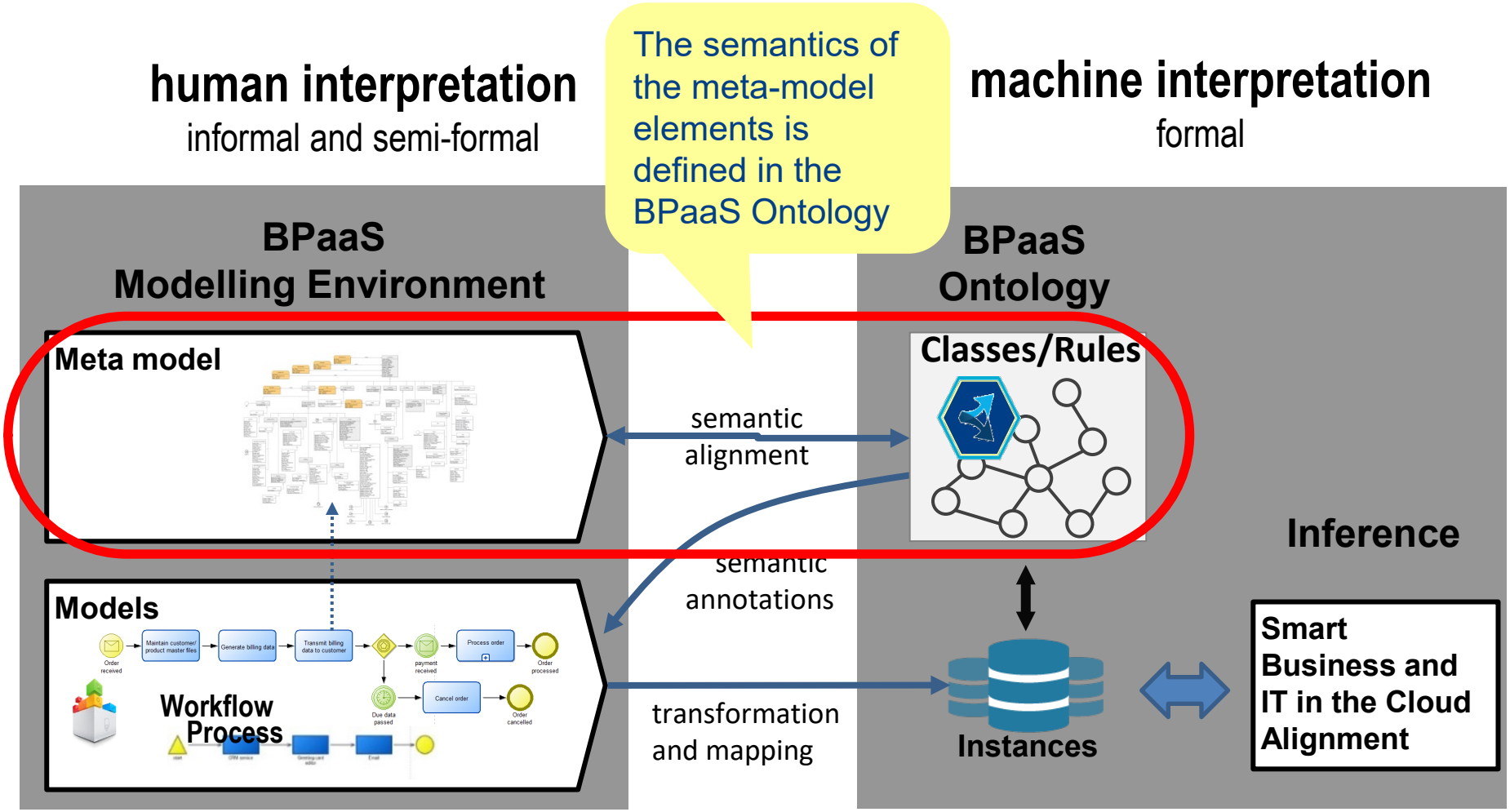
ontological metamodelling (lifting): *explication of type semantics*

Example: Business Process as a Service



From: CoudSocket Project

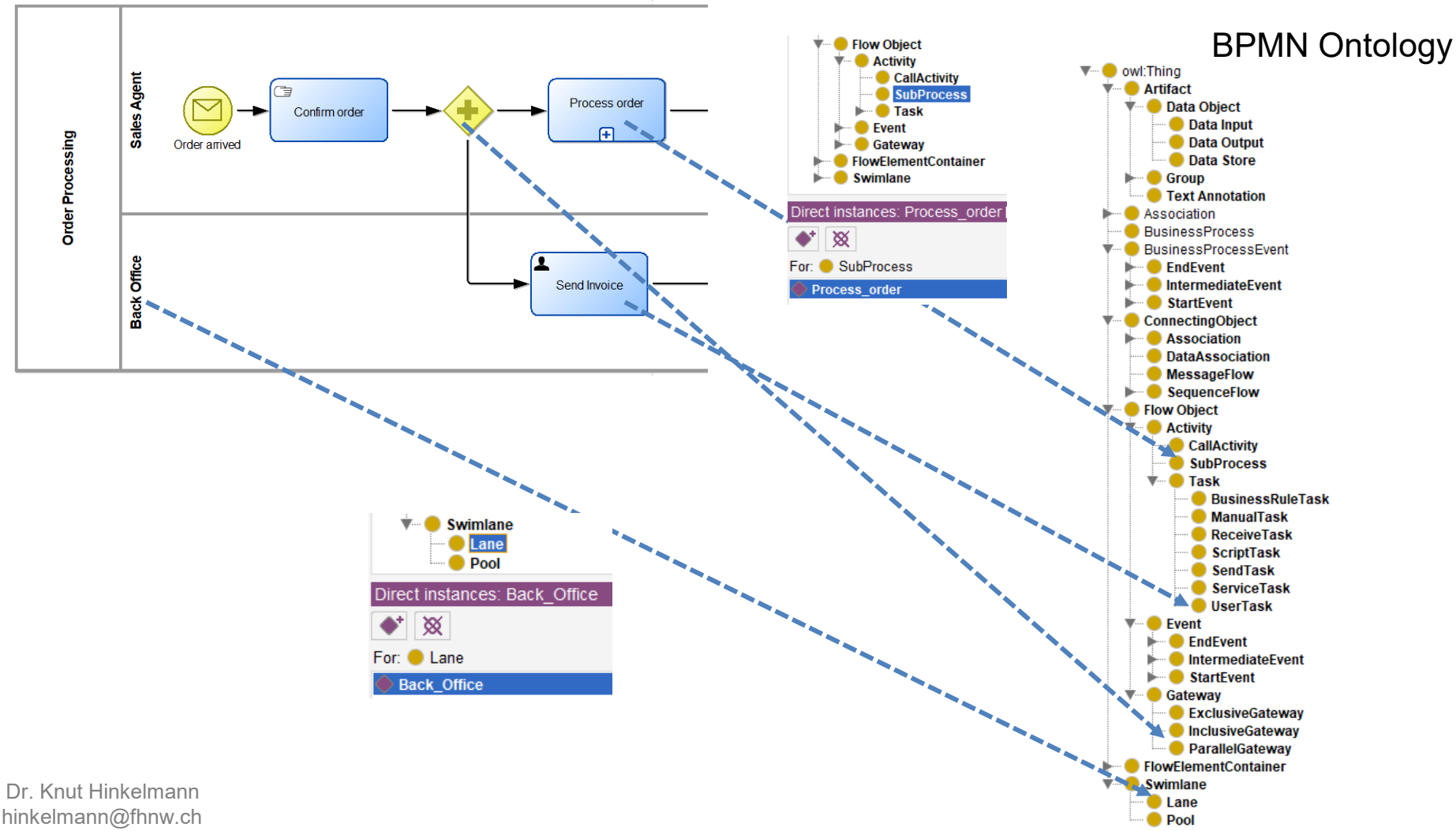
Example: Business Process as a Service



From: CoudSocket Project

Transformation and Mapping

The model elements are exported as instances ontology classes

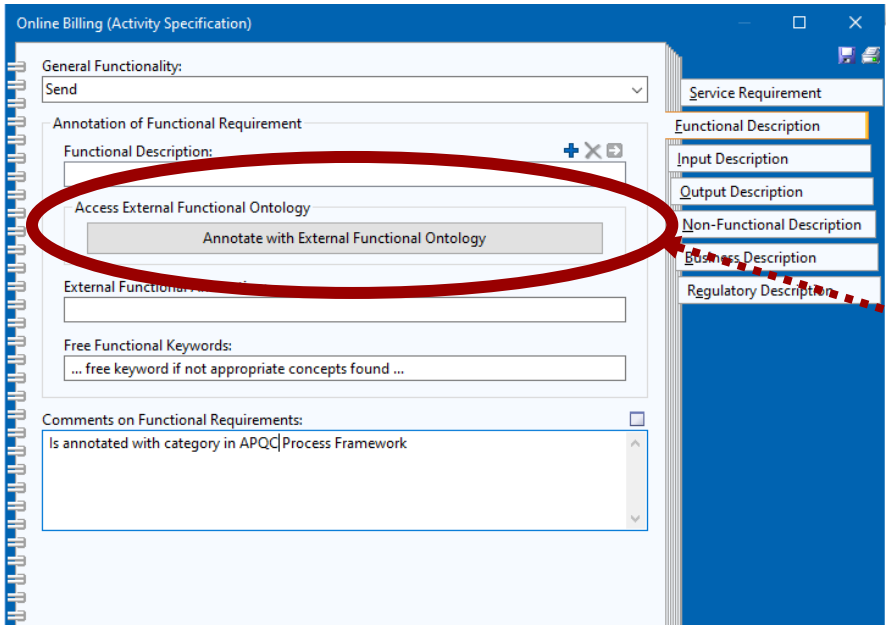


Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Semantic Annotations

Annotate modeling elements with classes from the domain ontology

Example: Functionality of a Service

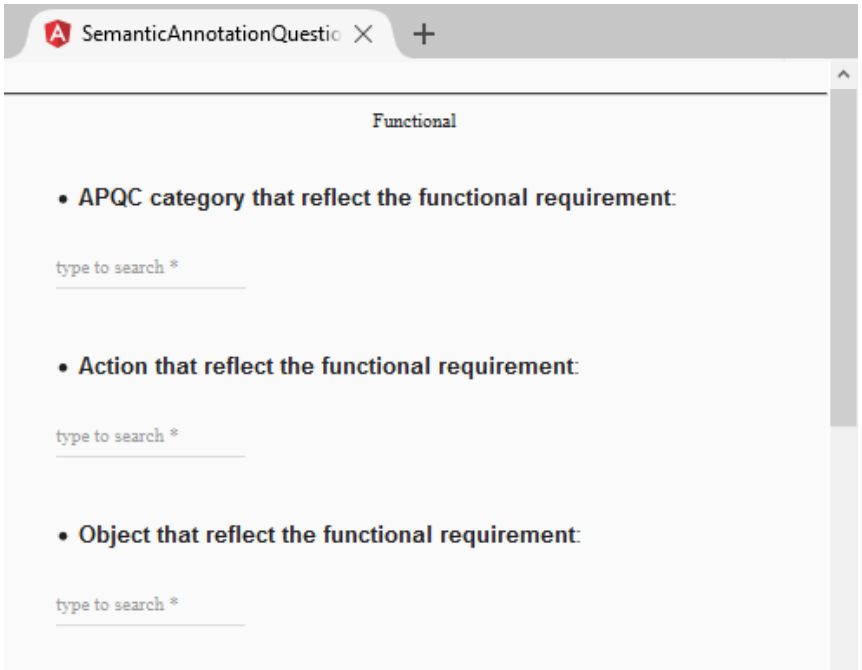


Domain Ontology:
APQC Process Classification Framework

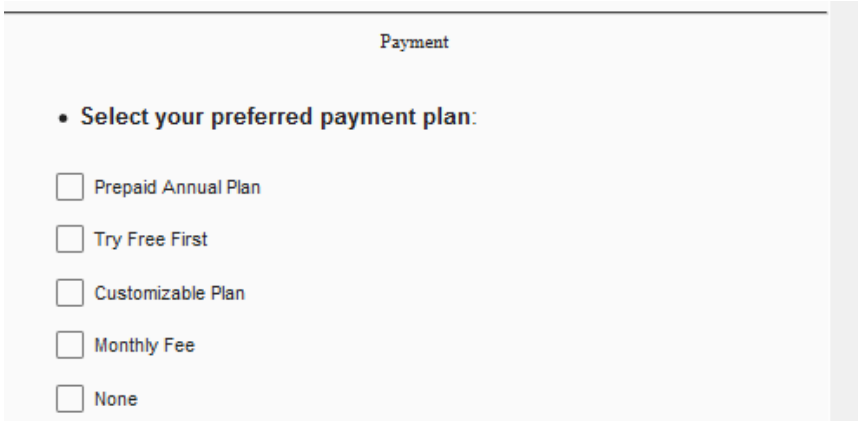
Inferencing: Cloud Service Selection

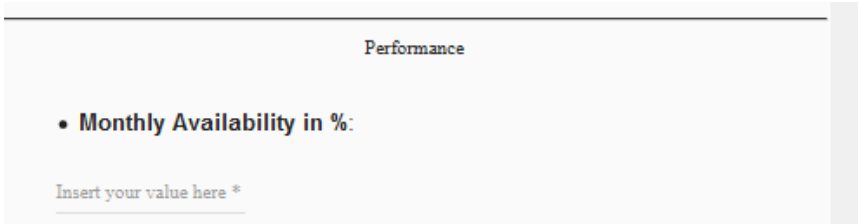
Cloud Service Selection

Functionality



Non-functional requirements





Thanks to Emanuele Laurenzi

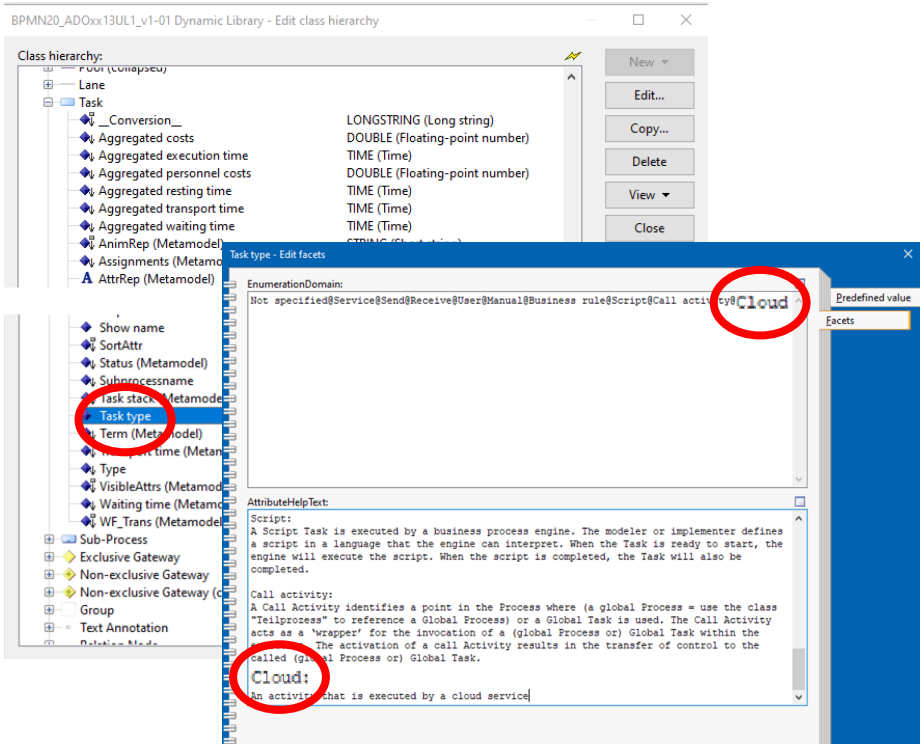
Drawbacks of Semantic Lifting

- 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 models must be translated again into the ontology instances

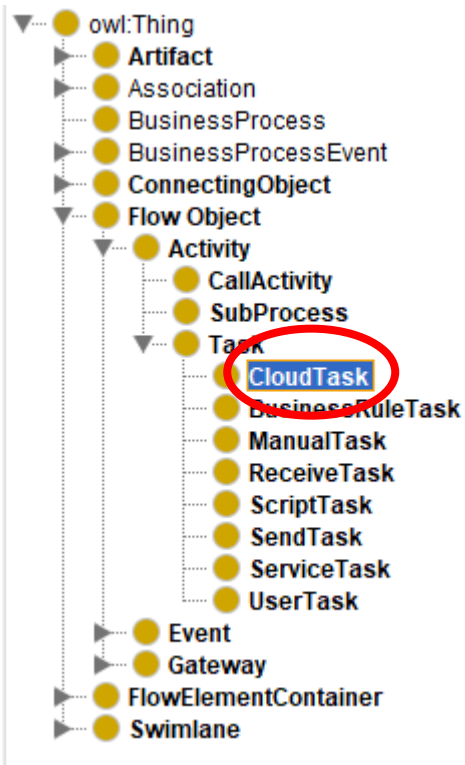
Example: New Model Element

- New task type: Cloud Task

Change in the meta model:

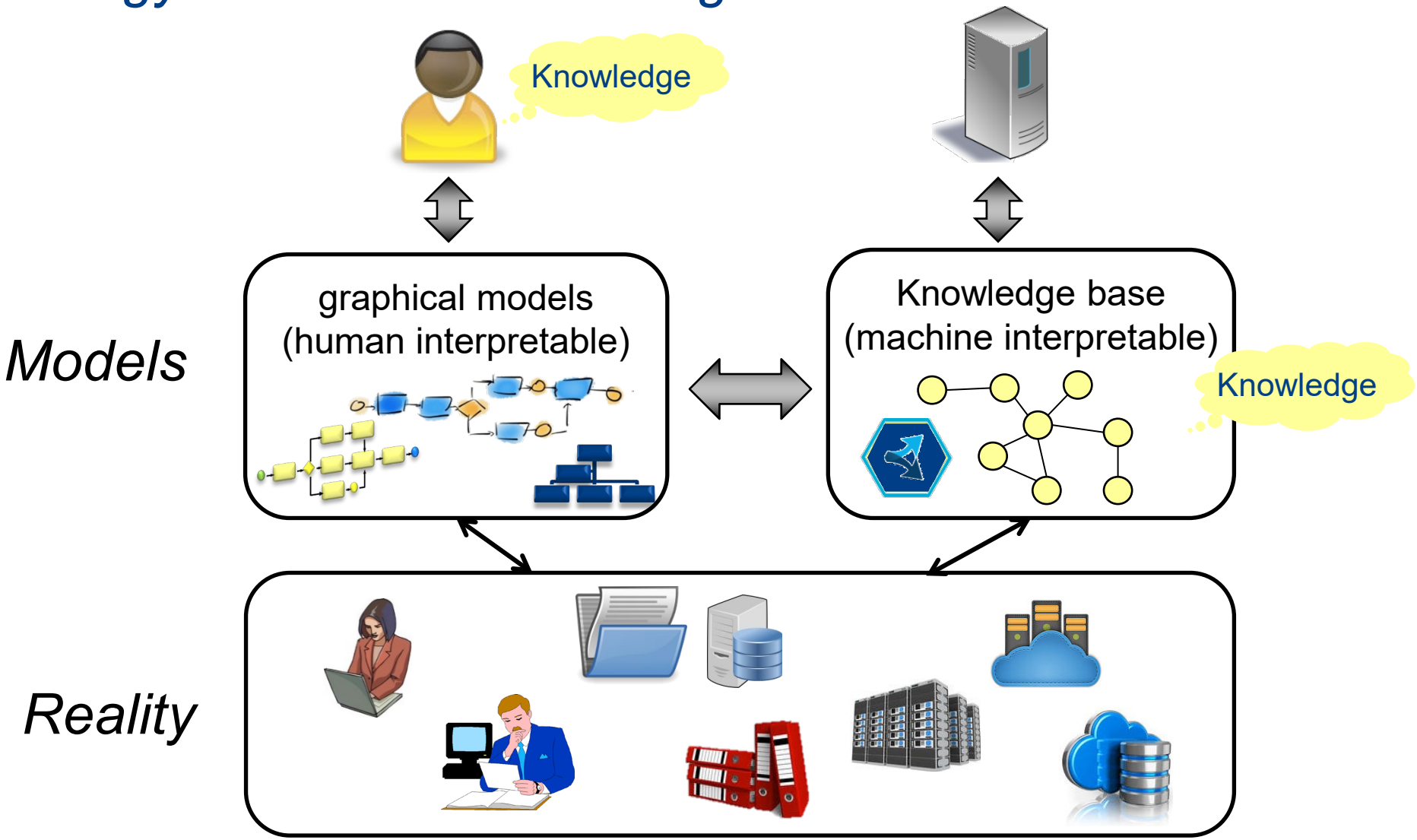


Change in the ontology:

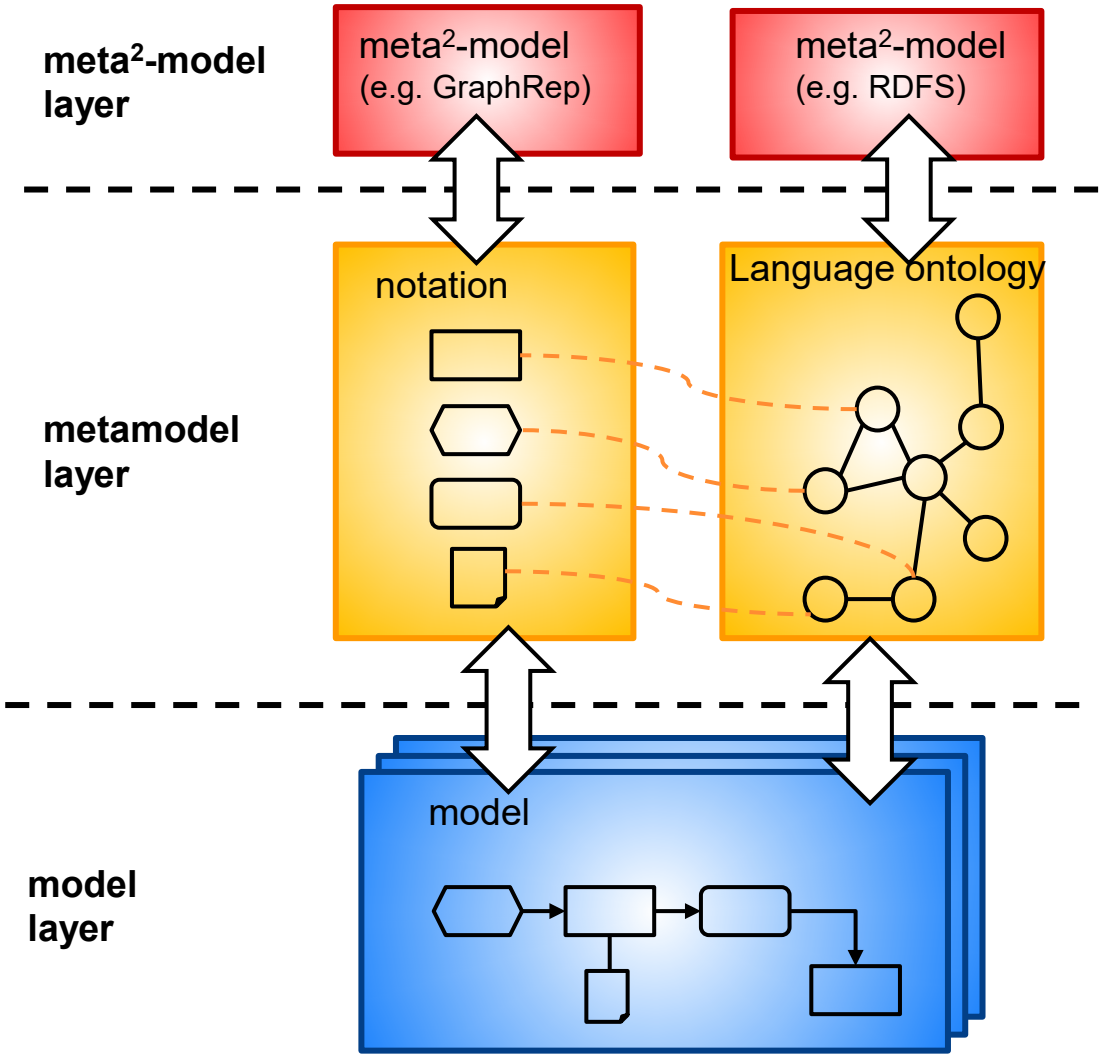


Ontology-based Metamodelling

Ontology-based Metamodeling



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

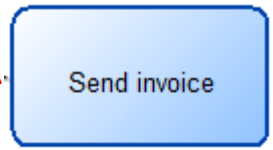


Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Modelling Language Ontologies

BPMN

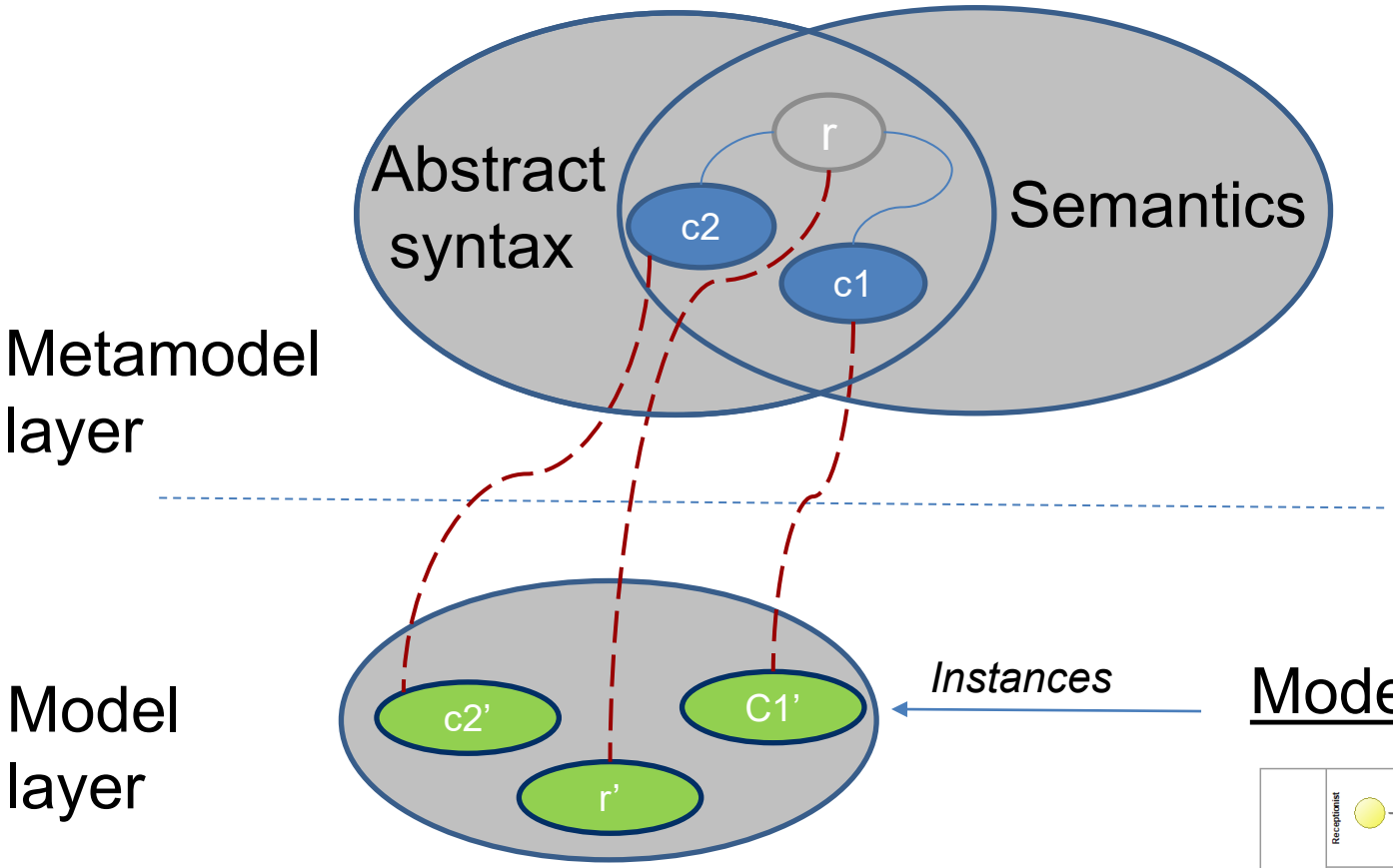
- owl:Thing
  - Artifact
    - Data Object
      - Data Input
      - Data Output
      - Data Store
    - Group
      - GeneratingCustomerBillingData
      - ManagingCustomerRelationship
      - TransmittingBillingData
      - Wholeprocess_sendinvoice
    - Text Annotation
  - Association
  - BusinessProcess
  - BusinessProcessEvent
  - ConnectingObject
    - Association
      - DataAssociation
    - DataAssociation
    - MessageFlow
    - SequenceFlow
      - Sequence Flow With Decision
  - Flow Object
    - Activity
      - CallActivity
      - SubProcess
      - Task
    - Event
      - EndEvent
      - IntermediateEvent
      - StartEvent
    - Gateway
  - FlowElementContainer
    - Process
  - Swimlane
    - Lane
    - Pool



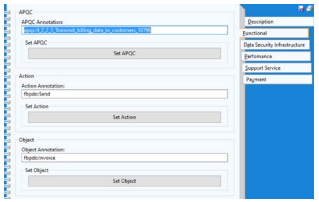
Archimate

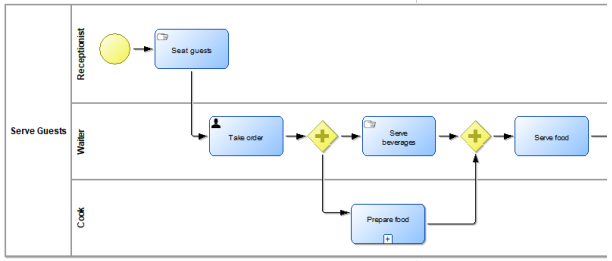
- owl:Thing
  - ADMetaModel
  - ArchitectureModel
  - ArchitectureViewpoint
  - EnterpriseObject
    - Action
      - ApplicationFunction
      - ApplicationInteraction
      - BusinessBehaviourElement
        - BusinessEvent
        - BusinessFunction
        - BusinessInteraction
        - BusinessProcess
      - InfrastructureFunction
      - SystemSoftware
    - ActiveStructureElement
      - ApplicationComponent
      - BusinessActor
      - BusinessRole
      - CommunicationPath
    - Interface
    - PassiveStructureElement
    - Service
  - InfrastructureElement
    - Network
    - Node
      - Device
      - SystemSoftware
  - Relationships
    - DynamicRelationships
    - OtherRelationships
    - StructuralRelationships
      - Access
      - Aggregation
      - Assignment
      - Association
      - Composition
      - Realisation
      - UsedBy
  - TopLevelElements
    - Event
    - Location
    - Time

Ontology



Model

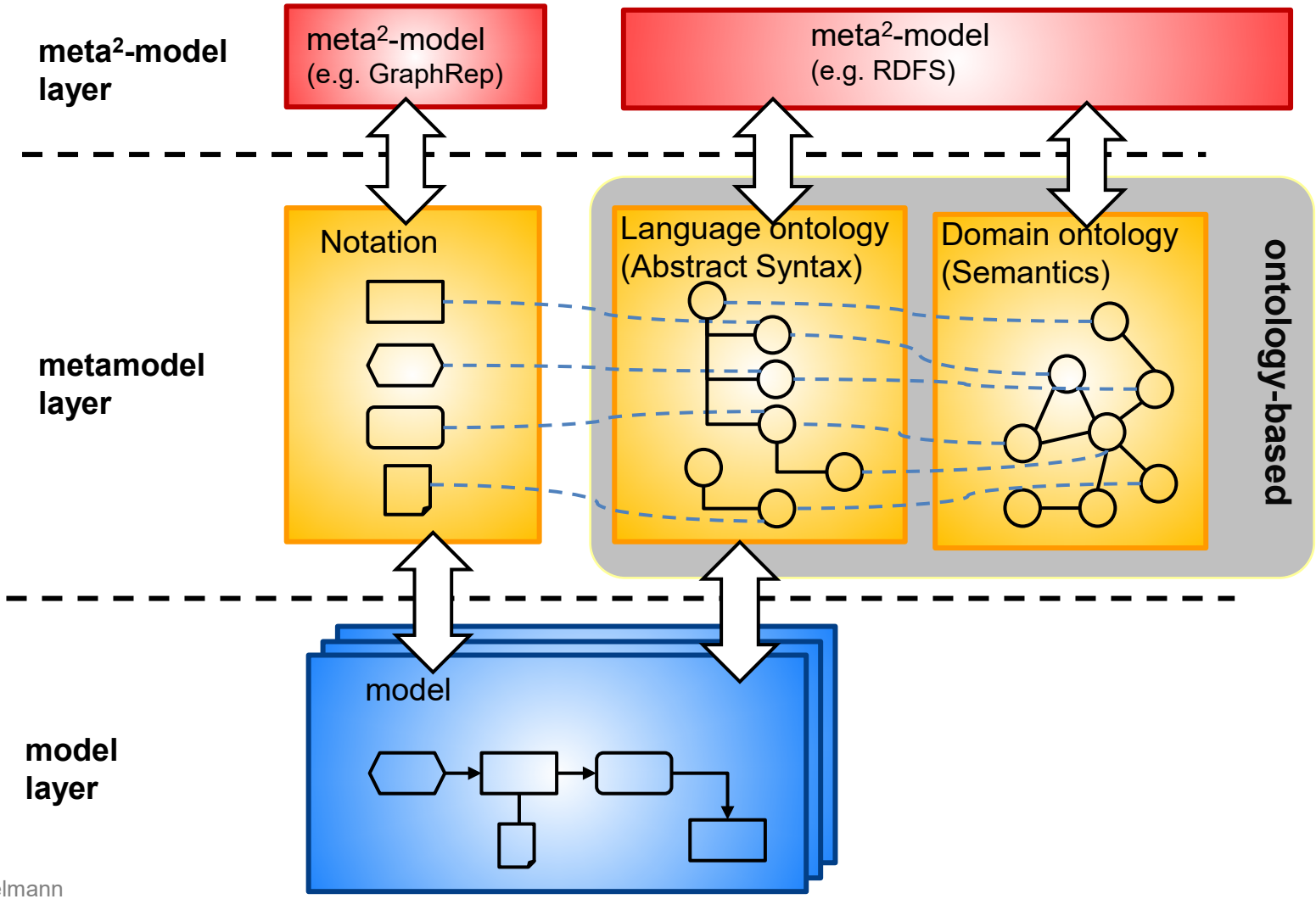




Thanks to Emanuele Laurenzi

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

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



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

Domain Ontologies

Enterprise Ontology (excerpt)

- owl:Thing
  - Action
    - Collaborate
    - Create
    - Send
  - ApplicationService
  - BackupType
  - BusinessObject
    - Invoice
    - Order
    - Pricing Model
  - ActiveStructureElement
    - ApplicationComponent
    - BusinessActor
      - LegalEntity
      - OrganisationalUnit
      - Person
    - BusinessRole
      - BusinessCollaboration
      - BusinessPartner
        - Client
        - Consultant
        - Consumer
        - Contractor
        - ContractPartner
        - Customer
        - Influencer
        - Supplier
      - Employee
        - ClerkRole
        - ExpertRole
        - ManagerRole

Action type

Object

Send invoice

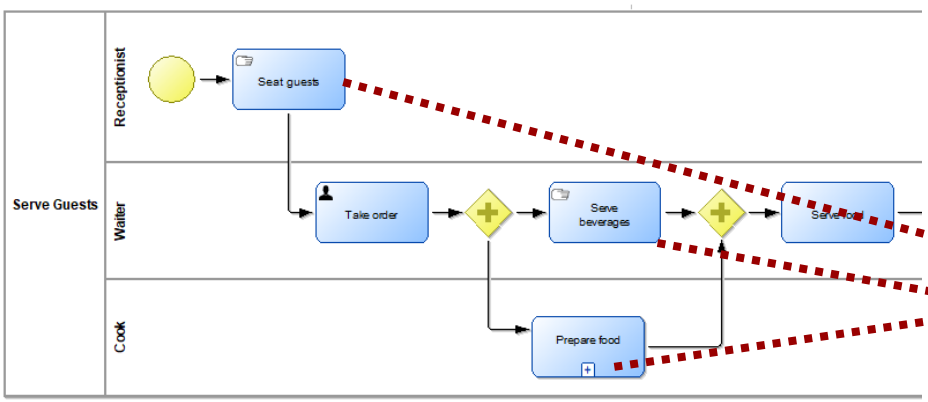
APQC Class

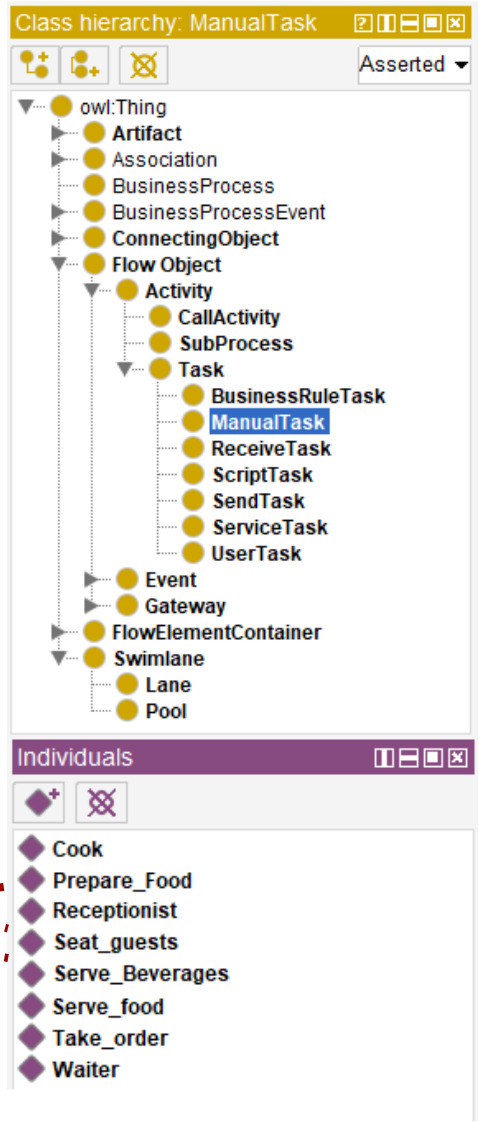
Domain Ontology:
APQC Process Classification Framework

- 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 customer
          - 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

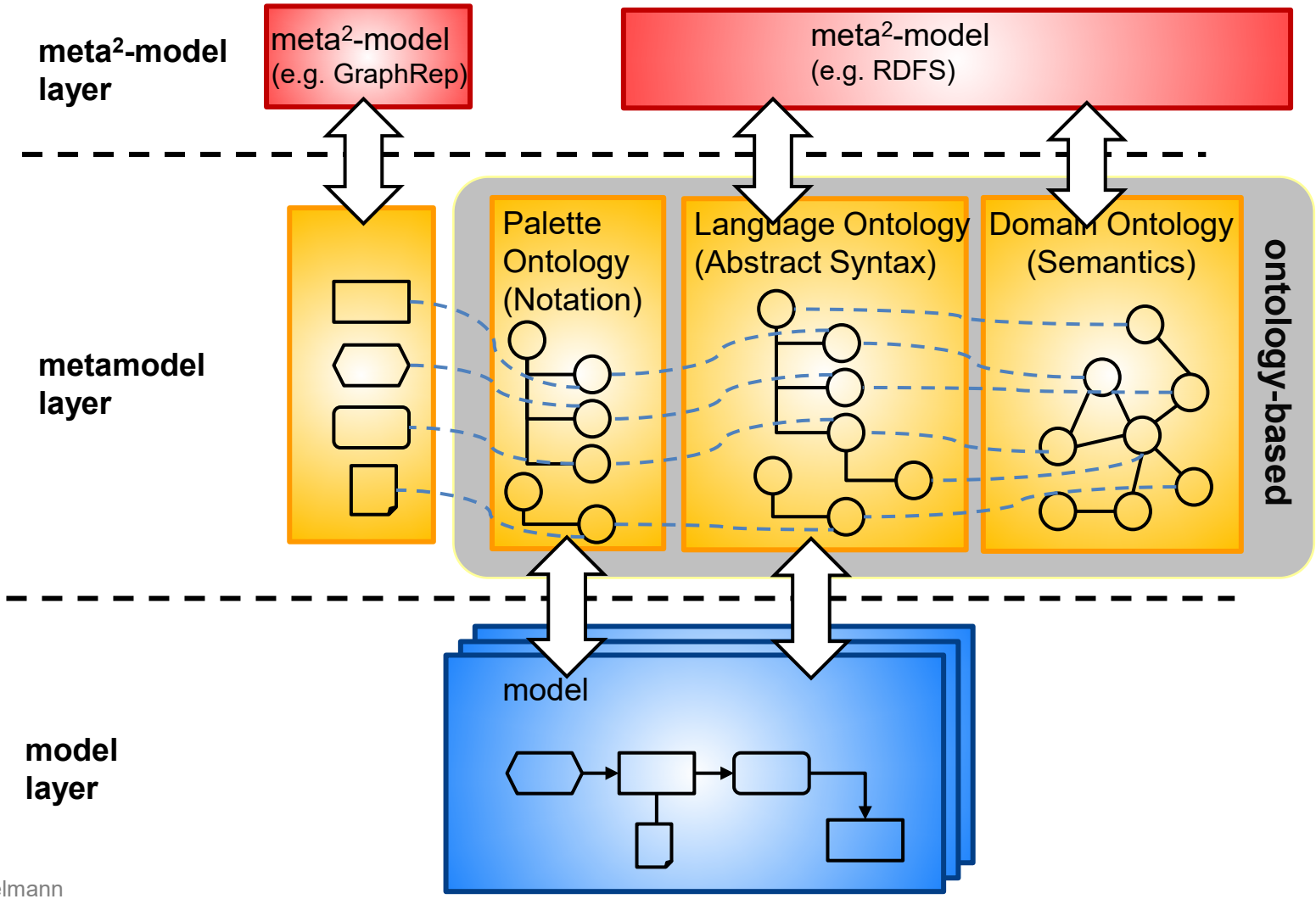
Ontology-Based Modeling

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





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



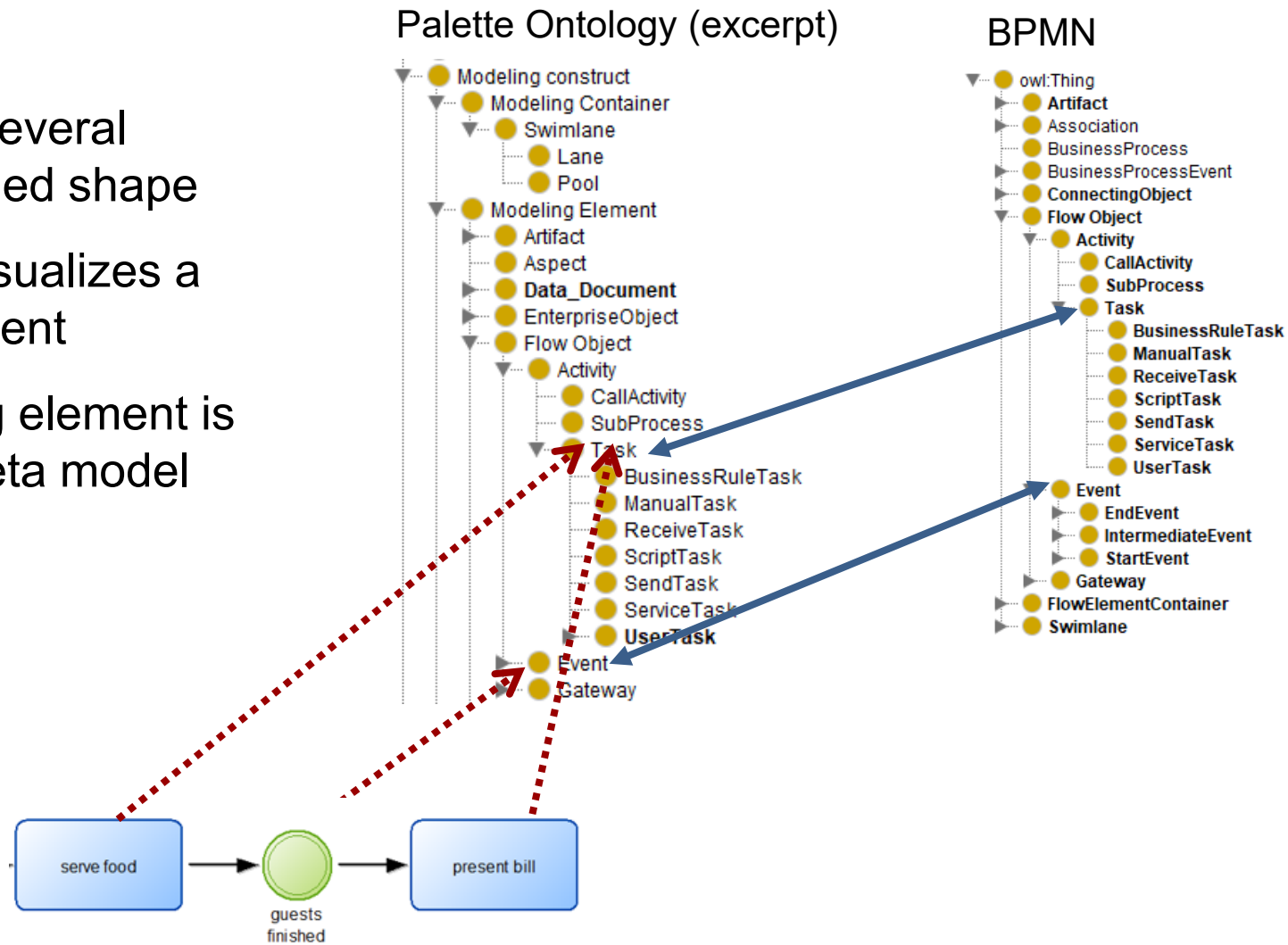
Palette Ontology

Palette Ontology (excerpt)

- Modeling construct
  - Modeling Container
    - Swimlane
      - Lane
      - Pool
  - Modeling Element
    - Artifact
    - Aspect
    - **Data_Document**
    - EnterpriseObject
    - Flow Object
      - Activity
        - CallActivity
        - SubProcess
        - Task
          - BusinessRuleTask
          - ManualTask
          - ReceiveTask
          - ScriptTask
          - SendTask
          - ServiceTask
          - **UserTask**
      - Event
      - Gateway
    - OrganizationalModelElement
    - Perspective
    - Sentry
    - Swimlane
    - Task
  - Modeling Relation
    - Belongs_To
    - ConnectingObject
      - Association
      - DataAssociation
      - MessageFlow
      - SequenceFlow
    - FlowElementContainer
    - OrganizationalModelConnector
    - Relationships

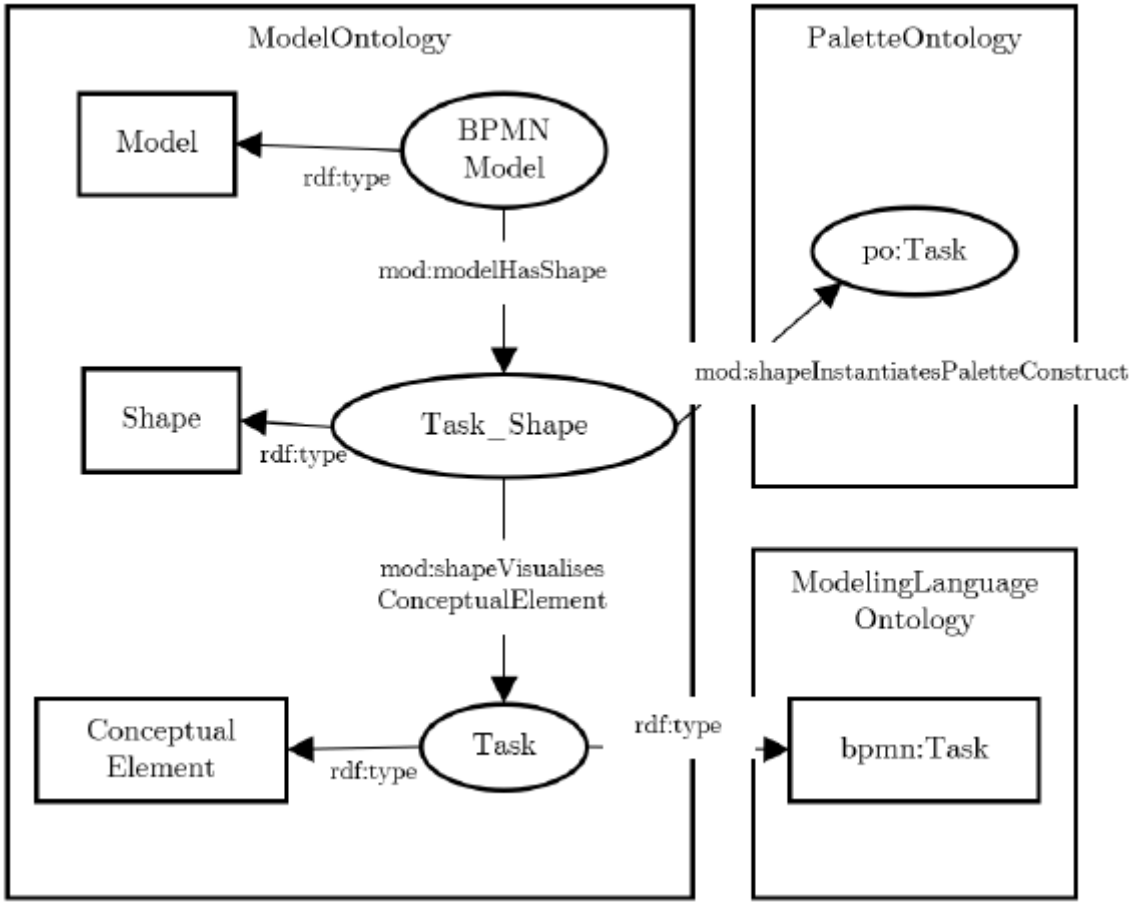
Representing Models in AOAME

- Models have several elements, named shape
- Each shape visualizes a modeling element
- Each modeling element is related to a meta model construct



Representing Models in AOAME

- Models have several elements, named shape
- Each shape visualizes a modeling element
- Each modeling element is related to a meta model construct
- Semantic alignment is built-in to the environment, because triples can be added for each conceptual element



Example Query

«Which task elements are in the model Serve Guests»?

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX mod: <http://fhnw.ch/modelingEnvironment/ModelOntology#>
PREFIX lo: <http://fhnw.ch/modelingEnvironment/LanguageOntology#>
PREFIX po: <http://fhnw.ch/modelingEnvironment/PaletteOntology#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX bpmn: <http://ikm-group.ch/archiMEO/BPMN#>


SELECT ?model ?shape ?task ?l
WHERE {
  ?model rdfs:label «Serve Guests".
  ?model mod:modelHasShape ?shape.
  ?shape mod:shapeVisualisesConceptualElement ?task.
  ?task rdf:type bpmn:Task .
  ?shape rdfs:label ?l.
}
```

Select the elements (named shapes) in the model

For the shapes find the conceptual elements

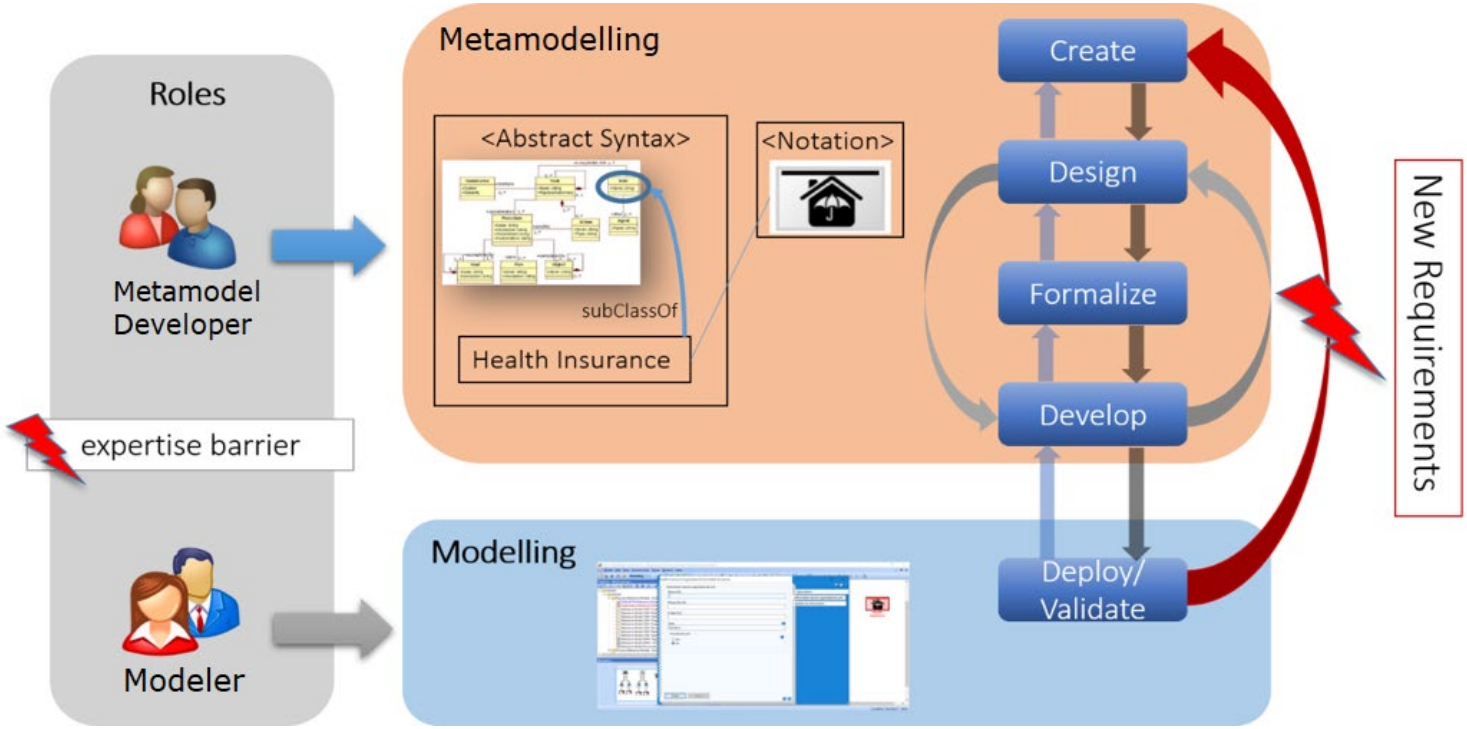
Filter the elements for BPMN Tasks and show the labels

Agile Modelling

Objective

Adapt modeling languages and ensure a precise shared interpretation of new modeling constructs to both **humans and machines**

Challenge: Separation of metamodelling and modelling

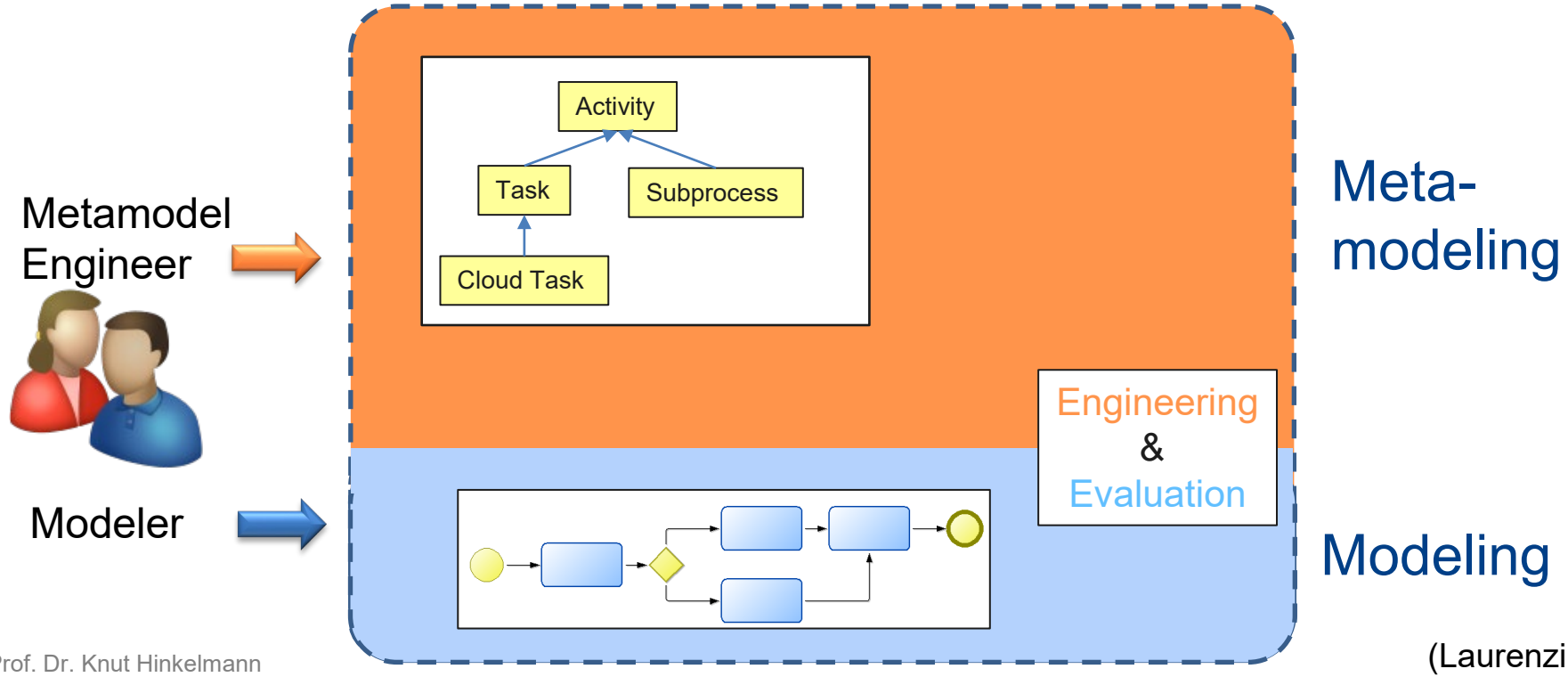


- Challenge 1: Metamodeling is a joint effort between metamodel experts and domain experts
- Challenge 2: Sequentialization of metamodeling and modeling is time consuming

Thanks to Emanuele Laurenzi

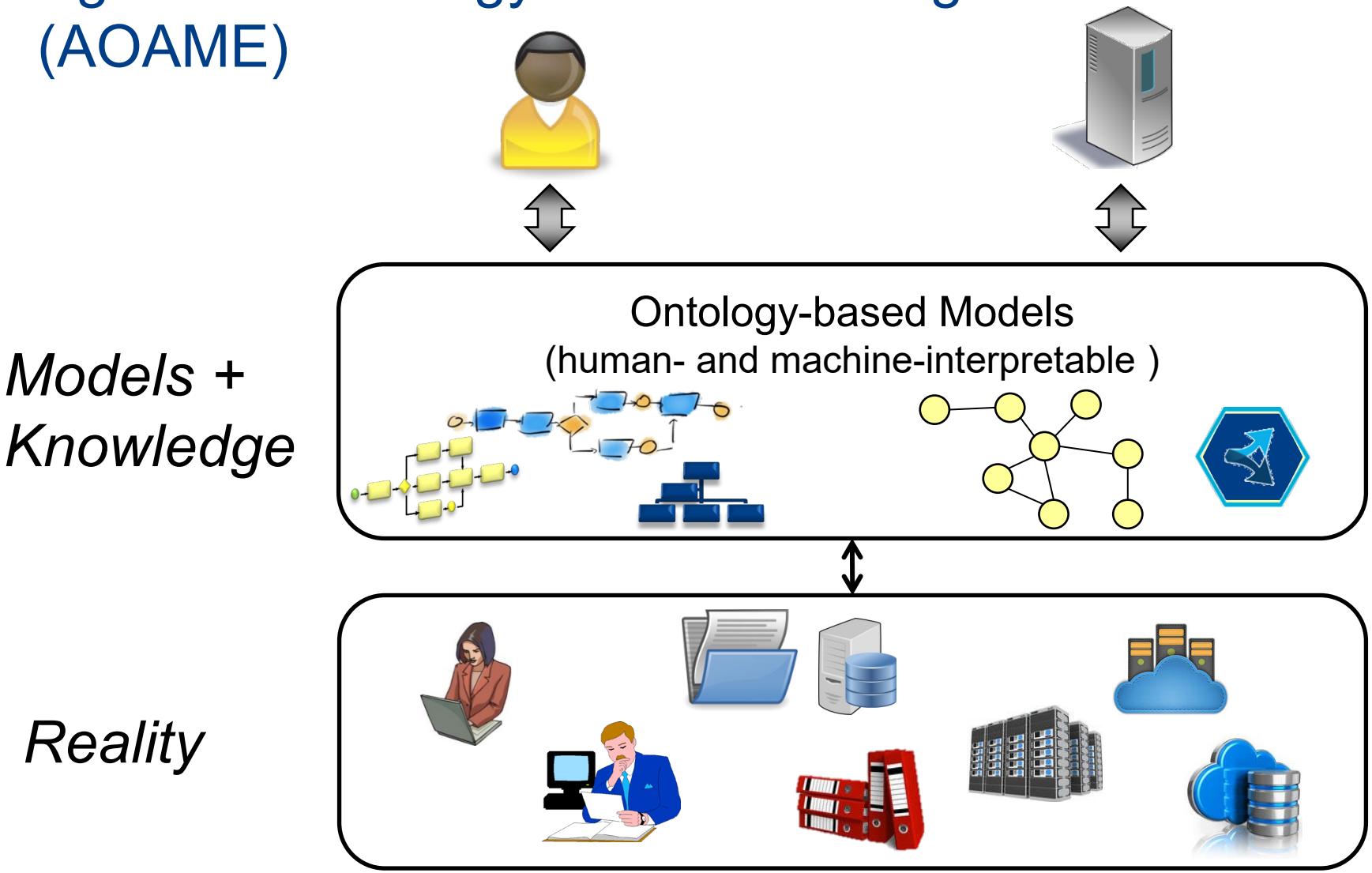
Integration Modeling and Metamodeling in a Single Environment

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



(Laurenzi et al. 2018)

Agile and Ontology-Aided Modeling Environment (AOAME)

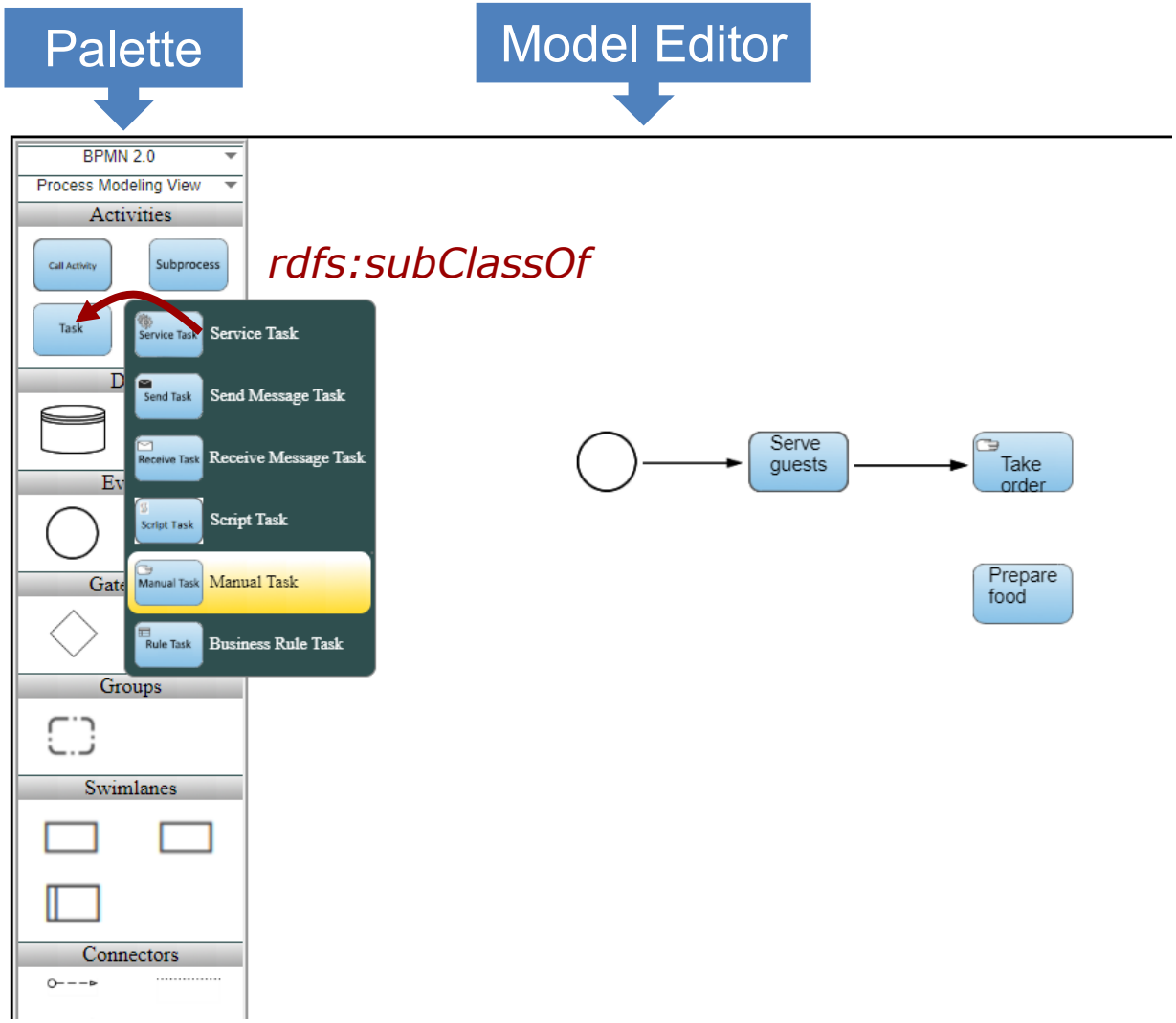


(Laurenzi et al. 2018)

AOAME: Agile and Ontology-Aided Modeling Environment

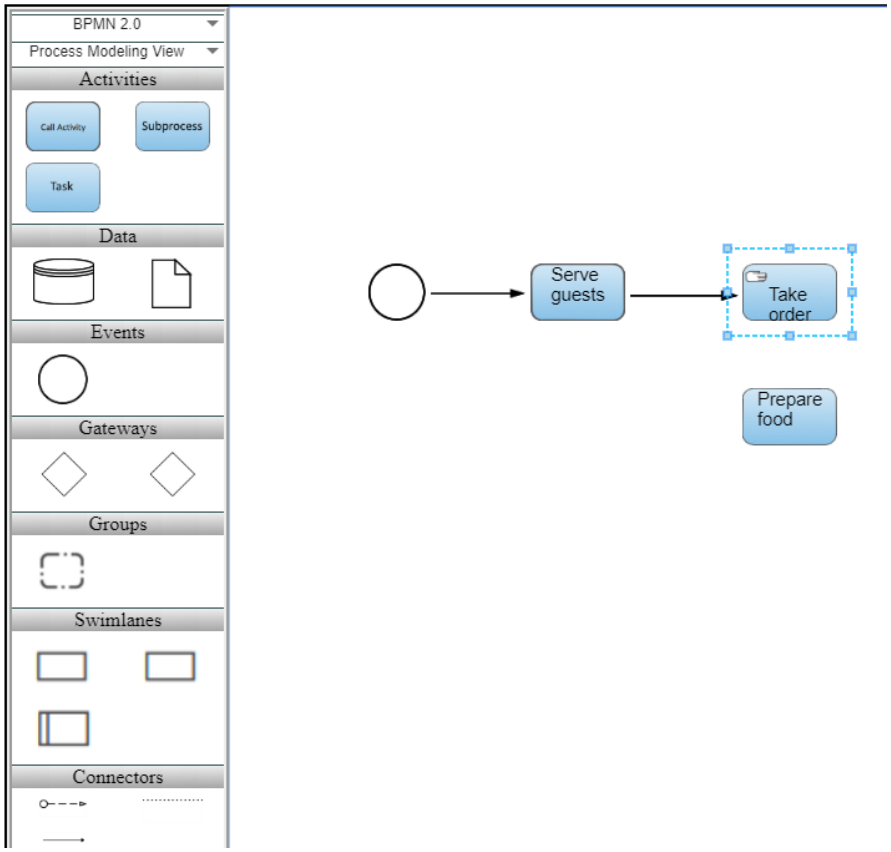
- AOAME is a a prototypical implementation for Agile and Ontology-Aided Modeling
- It is based on the PhD Thesis of Emanuele Laurenzi
- Implementation of the current version by
  - Emanuele Laurenzi
  - Charuta Pande
  - Devid Montecchiari
  - Egemen Kaba

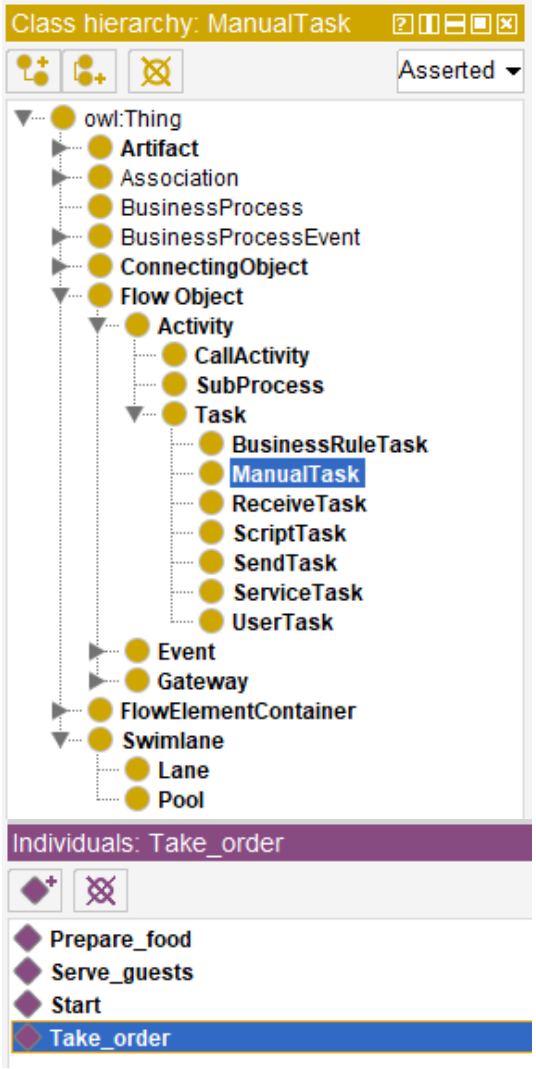
Ontology-Based Modeling in AOAME



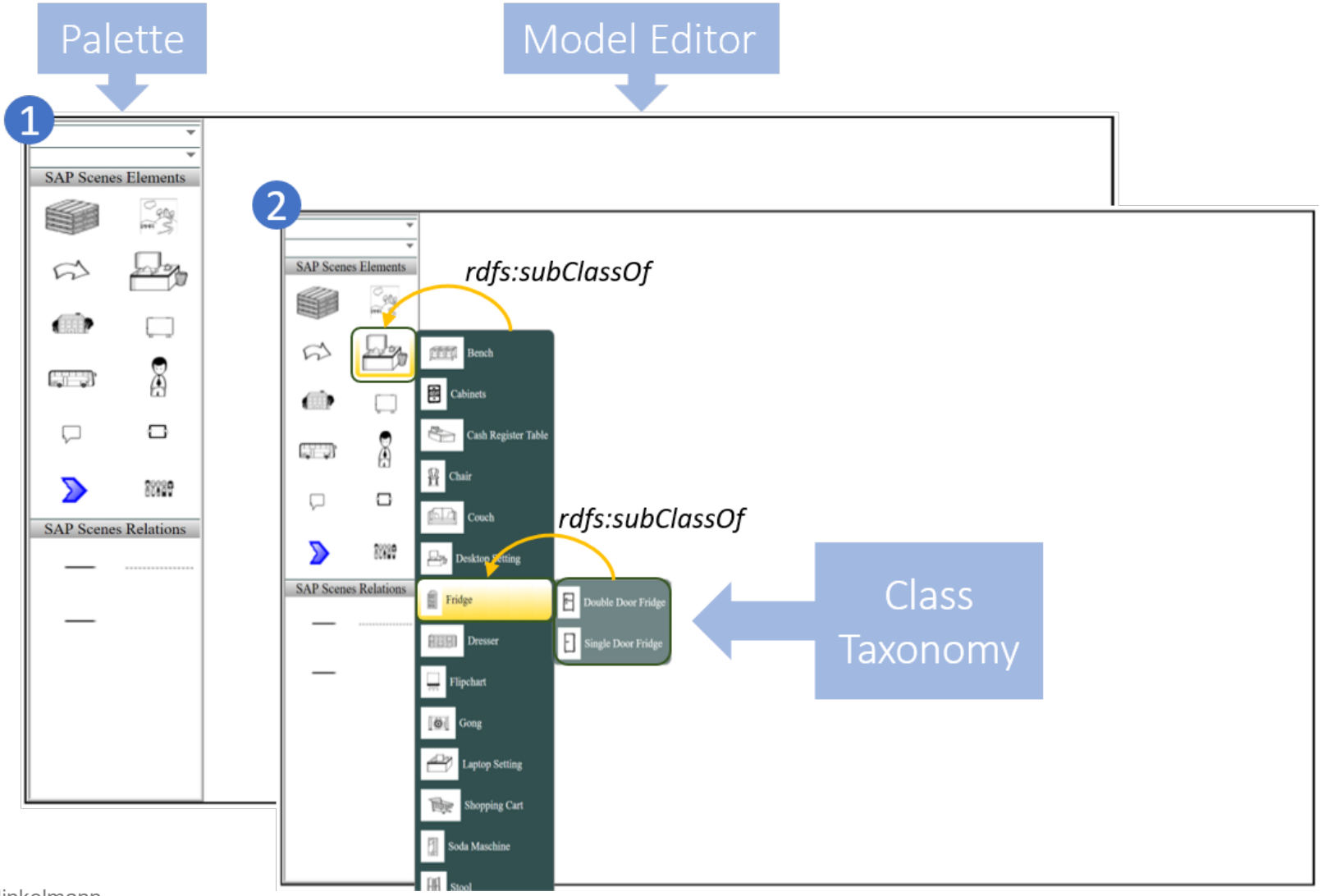
Ontology-Based Modelling

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

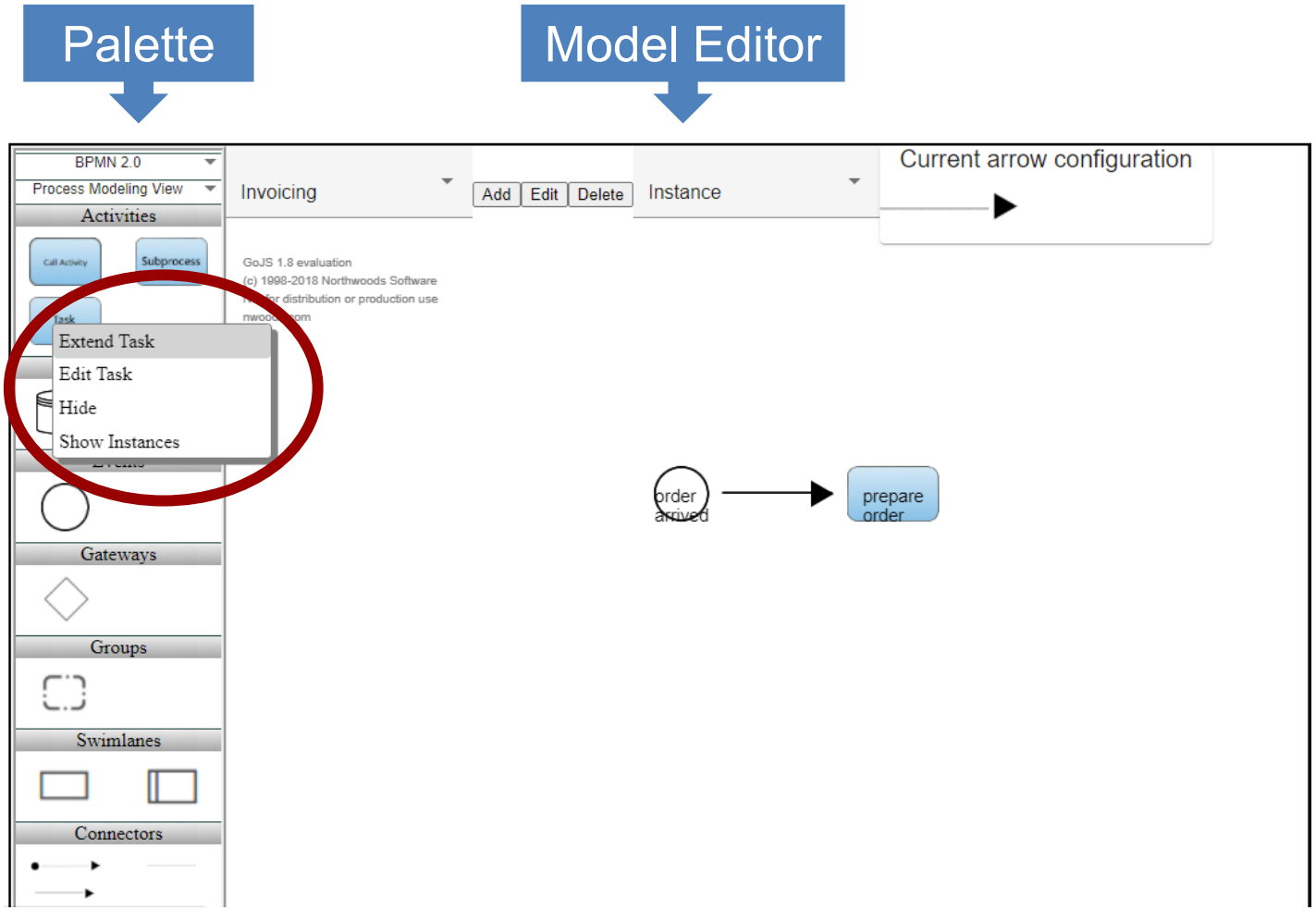




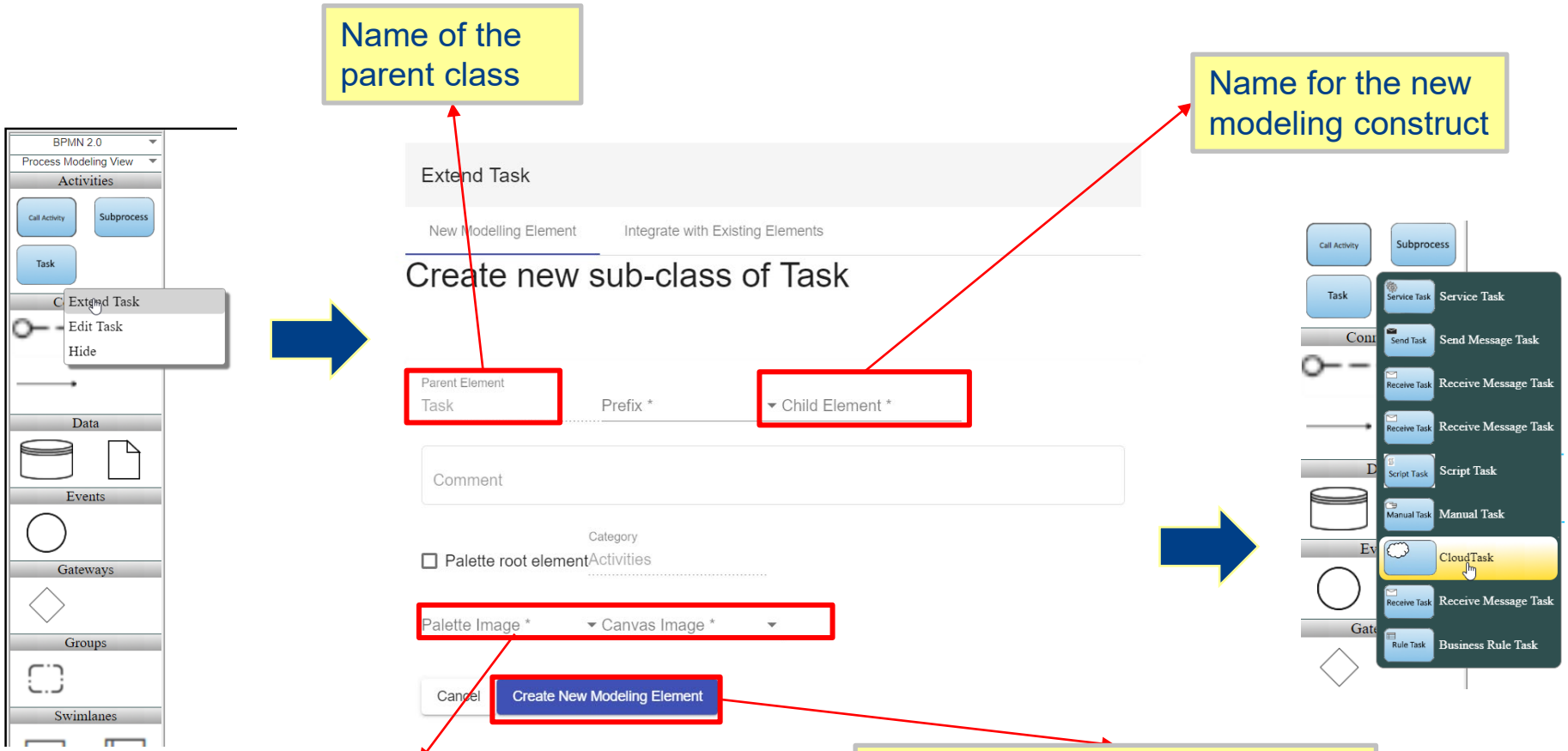
Modeling Elements are represented in a Class Hierarchy



Extending AOAME Modeling Languages – on the fly



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



Name of the parent class

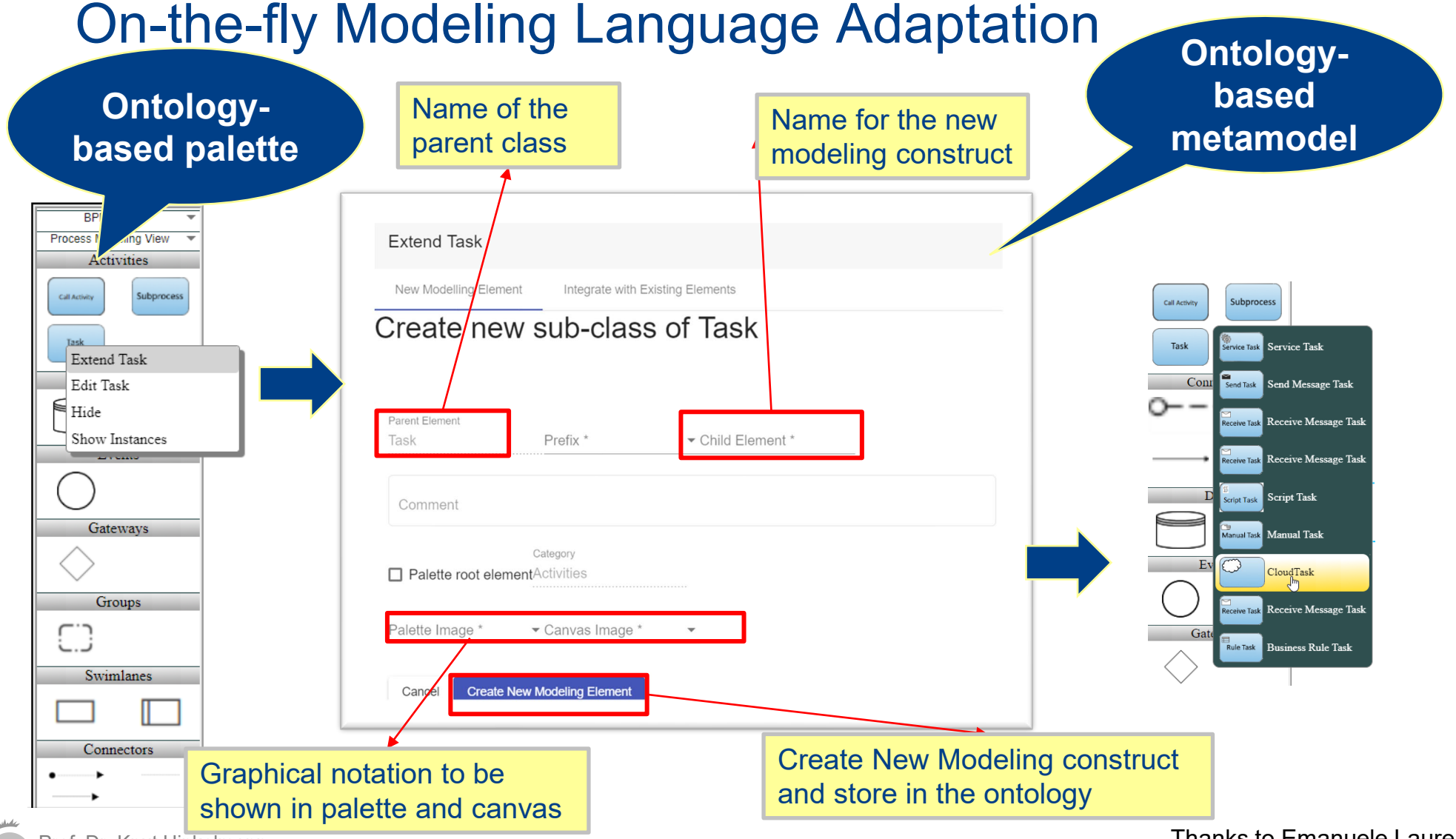
Name for the new modeling construct

Graphical notation to be shown in palette and canvas

Create New Modeling construct and store in the ontology

(Laurenzi et al. 2018)

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



Thanks to Emanuele Laurenzi

Semantic Alignment in AOAME

- With Semantic Mapping modeling elements can be connected to domain ontology