UNICAM
Università di Camerino
1336

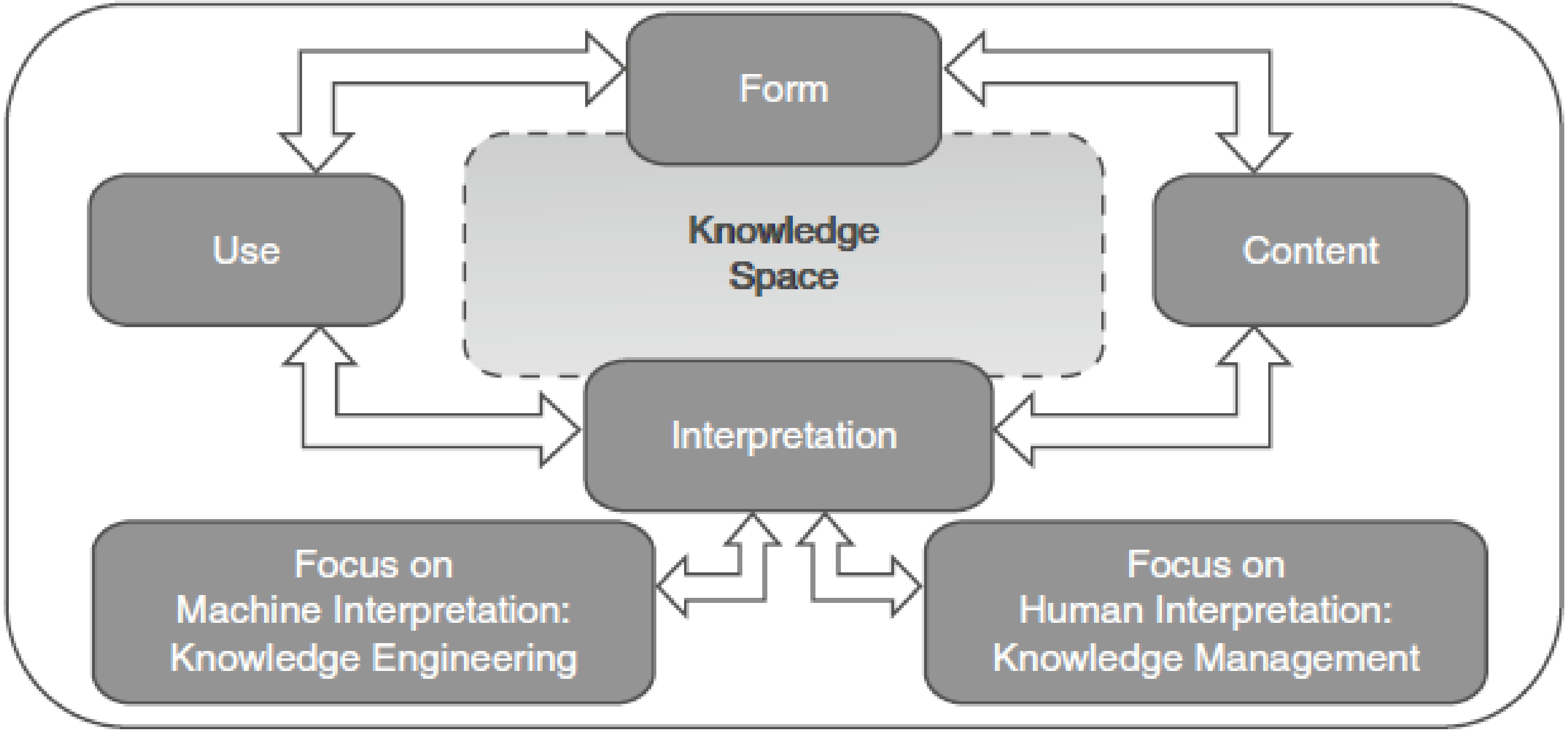
Ontology-based Modeling

Knut Hinkelmann

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

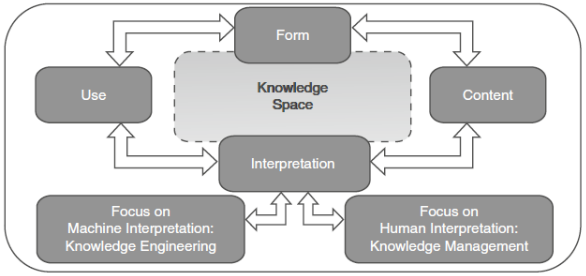
Models should allow automated analysis, decision making and digitalization

Dimensions of a Knowledge Space


Form
Use
Knowledge Space
Content
Interpretation
Focus on Machine Interpretation: Knowledge Engineering
Focus on Human Interpretation: Knowledge Management

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

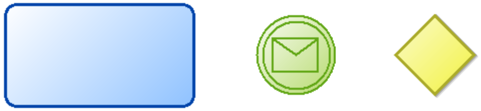
Dimensions of the Knowledge Space


Form
Use
Knowledge Space
Content
Interpretation
Focus on Machine Interpretation: Knowledge Engineering
Focus on Human Interpretation: Knowledge Management

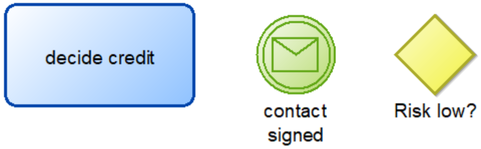
Use:

- 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

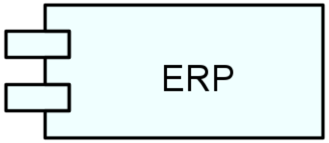


- ***Use***: Stakeholders and their concerns determine the relevant subset of the knowledge
- ***Form***: Syntax and semantic of *meta model concepts*.
- ***Content***: *Instantiation* of meta model concepts for a specific *application* (represented in the labels)
- ***Interpretation***: Giving meaning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

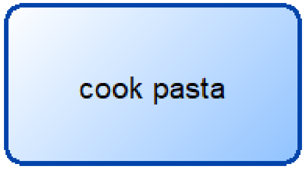
Content: Instantiation of Meta model + Application knowledge

- 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

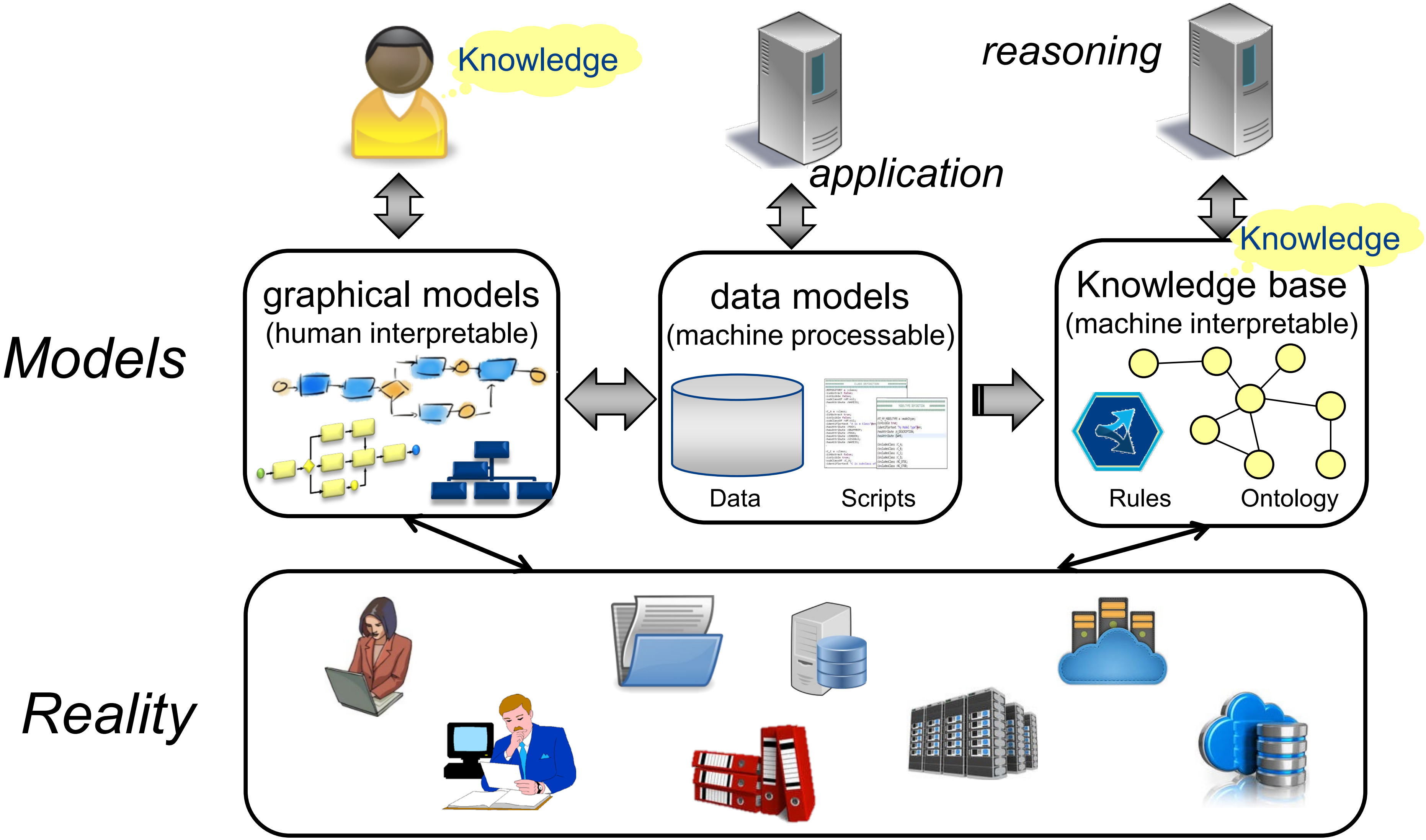


  - Meta model: Task
  - Application: «Cook pasta» is about preparing food

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

Semantic Lifting

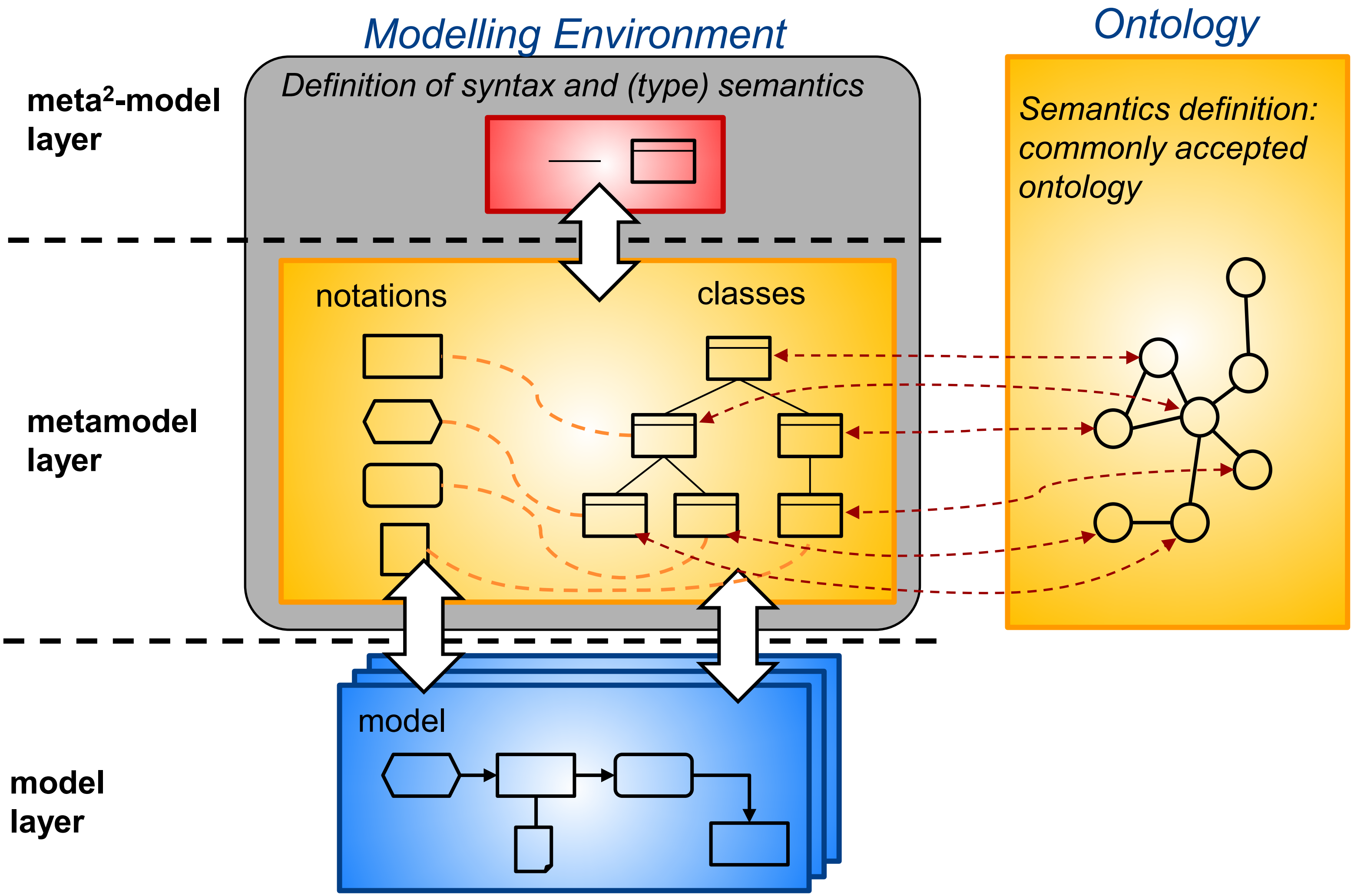
Semantic Lifting: Map Models into an Ontology



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

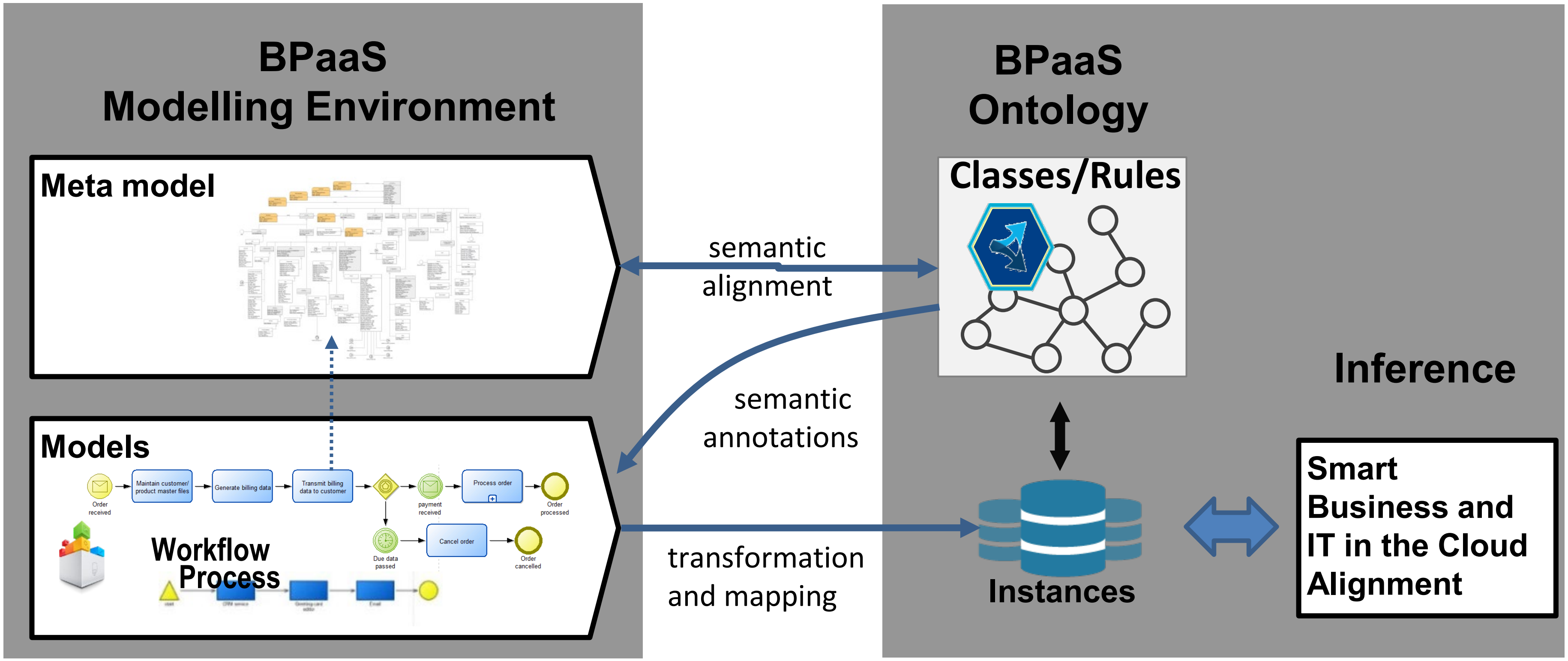


ontological metamodelling (lifting): *explication of type semantics*

Example: Business Process as a Service

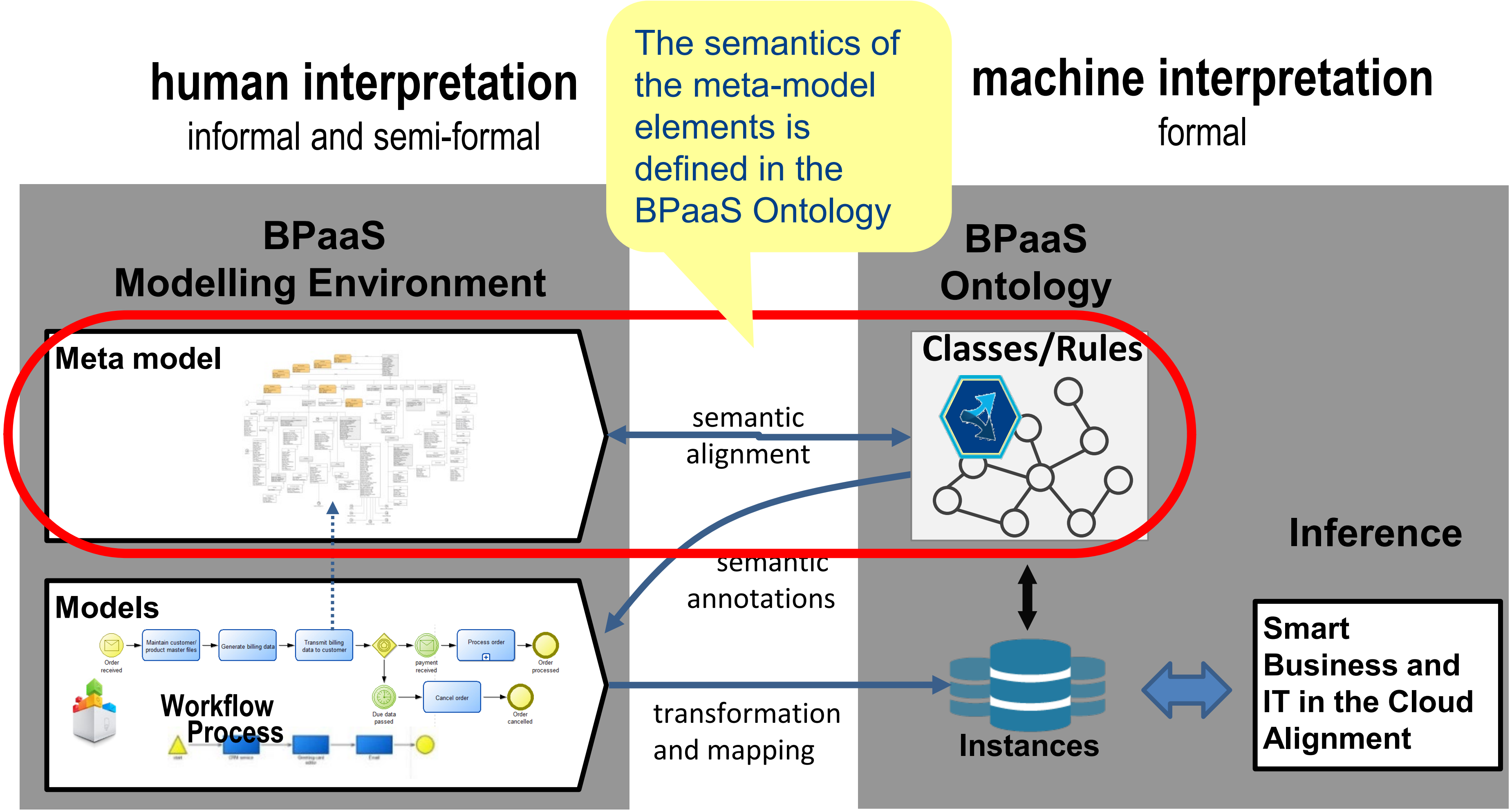
human interpretation
informal and semi-formal

machine interpretation
formal



From: CoudSocket Project

Example: Business Process as a Service



human interpretation

informal and semi-formal

The semantics of the meta-model elements is defined in the BPaaS Ontology

machine interpretation

formal

BPaaS Modelling Environment

Meta model

Models

Workflow Process

semantic alignment

semantic annotations

transformation and mapping

BPaaS Ontology

Classes/Rules

Instances

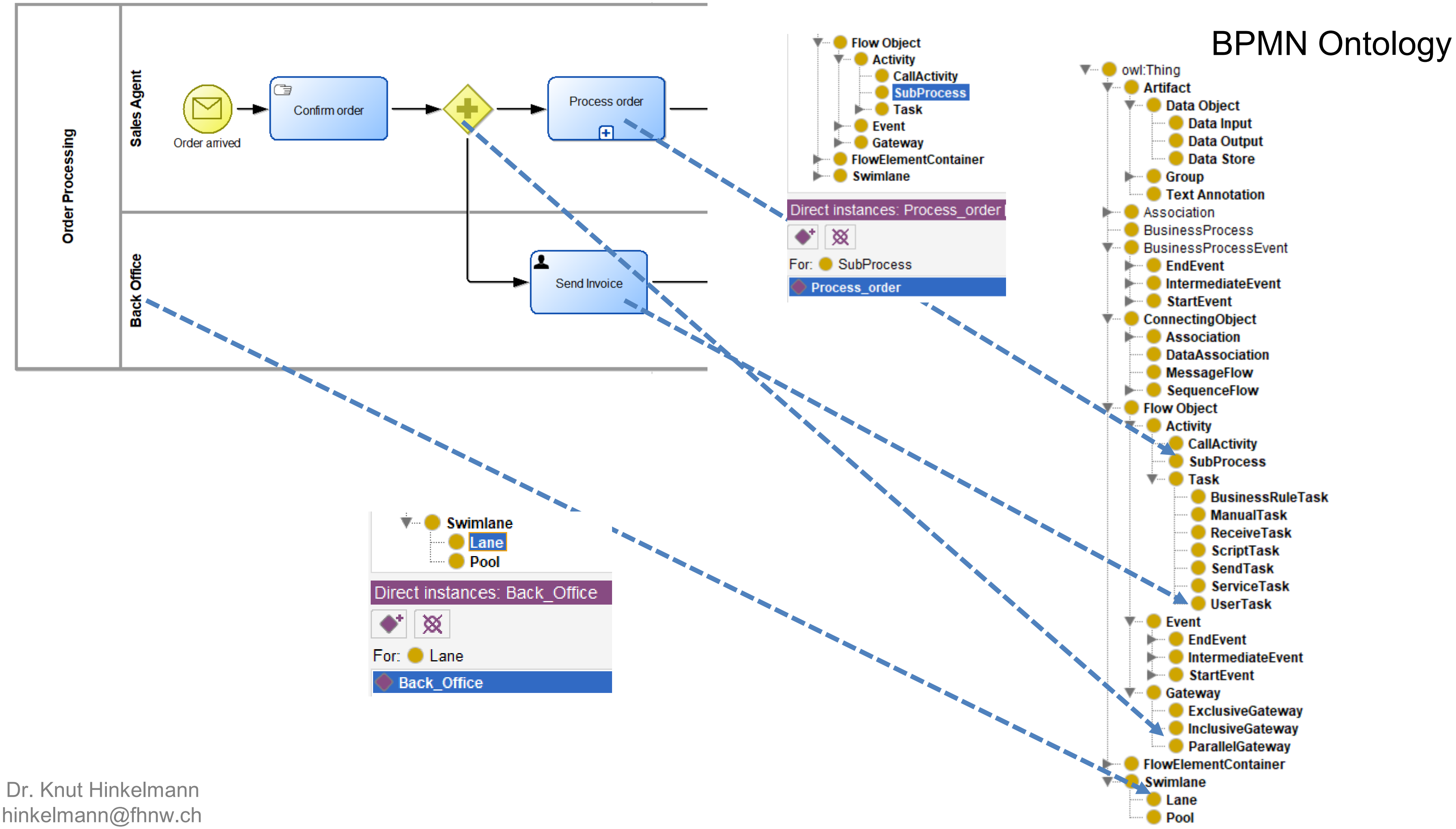
Inference

Smart Business and IT in the Cloud Alignment

From: CoudSocket Project

Transformation and Mapping

The model elements are exported as instances ontology classes



- Flow Object
  - Activity
    - CallActivity
    - SubProcess
    - Task
  - Event
  - Gateway
- FlowElementContainer
- Swimlane

Direct instances: Process_order

For: SubProcess

Process_order

- Swimlane
  - Lane
  - Pool

Direct instances: Back_Office

For: Lane

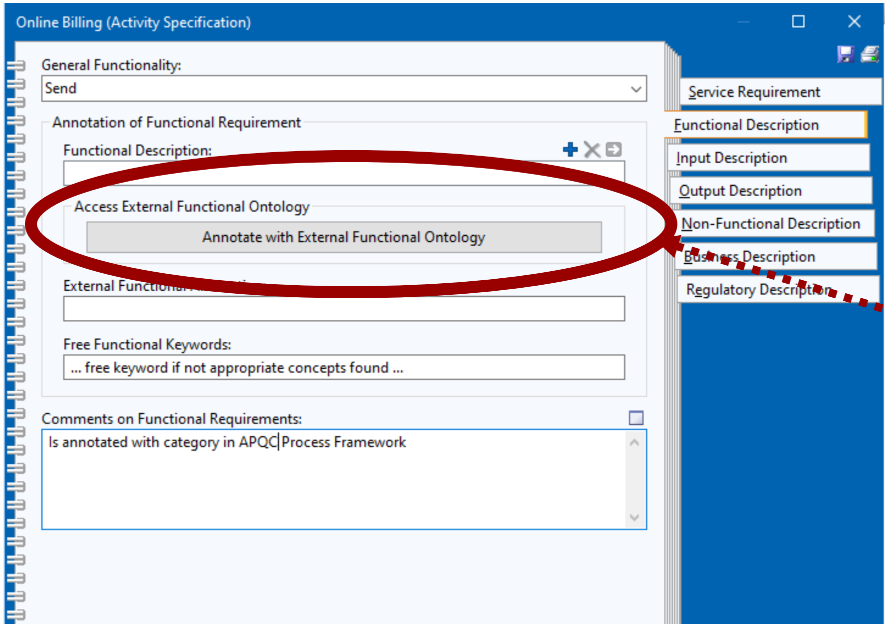
Back_Office

- owl:Thing
  - Artifact
    - Data Object
      - Data Input
      - Data Output
      - Data Store
    - Group
    - Text Annotation
  - Association
  - BusinessProcess
  - BusinessProcessEvent
    - EndEvent
    - IntermediateEvent
    - StartEvent
  - ConnectingObject
    - Association
    - DataAssociation
    - MessageFlow
    - SequenceFlow
  - Flow Object
    - Activity
      - CallActivity
      - SubProcess
      - Task
        - BusinessRuleTask
        - ManualTask
        - ReceiveTask
        - ScriptTask
        - SendTask
        - ServiceTask
        - UserTask
    - Event
      - EndEvent
      - IntermediateEvent
      - StartEvent
    - Gateway
      - ExclusiveGateway
      - InclusiveGateway
      - ParallelGateway
  - FlowElementContainer
  - Swimlane
    - Lane
    - Pool

Semantic Annotations

Annotate modeling elements with classes from the domain ontology

Example: Functionality of a Service



Online Billing (Activity Specification)

General Functionality:
Send

Annotation of Functional Requirement

Functional Description:

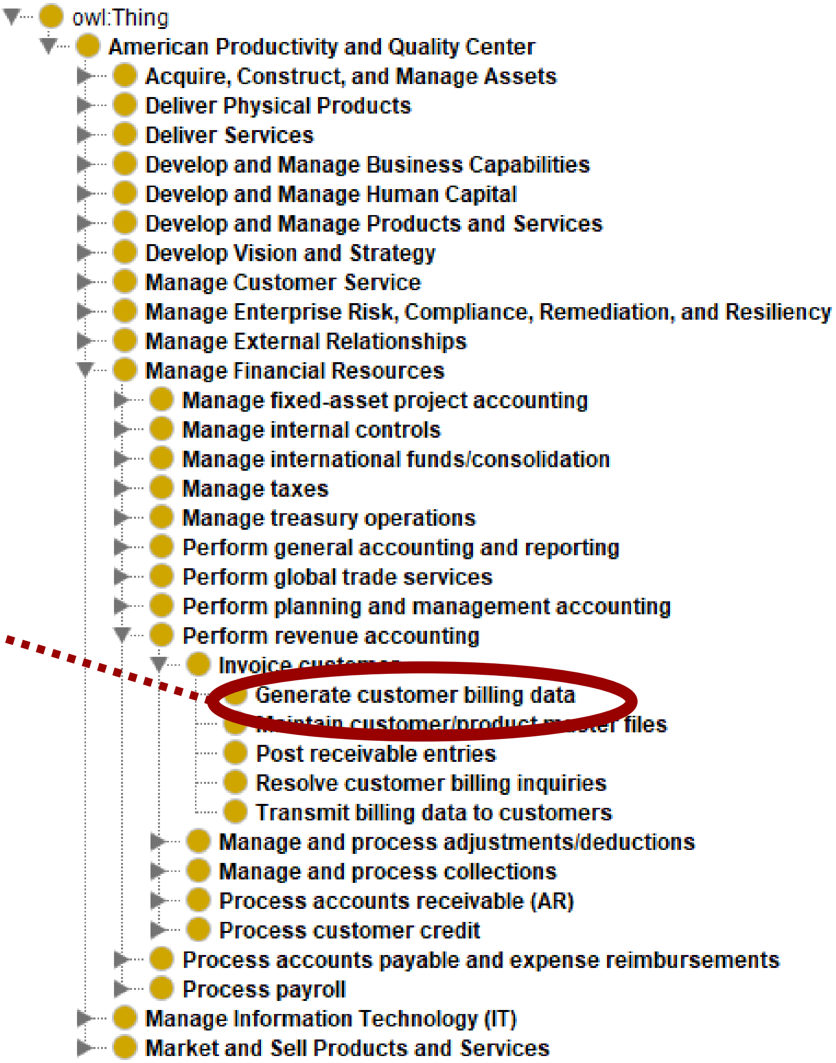
Access External Functional Ontology

External Functional Annotation:

Free Functional Keywords:
... free keyword if not appropriate concepts found ...

Comments on Functional Requirements:
Is annotated with category in APQC Process Framework

Domain Ontology:
APQC Process Classification Framework

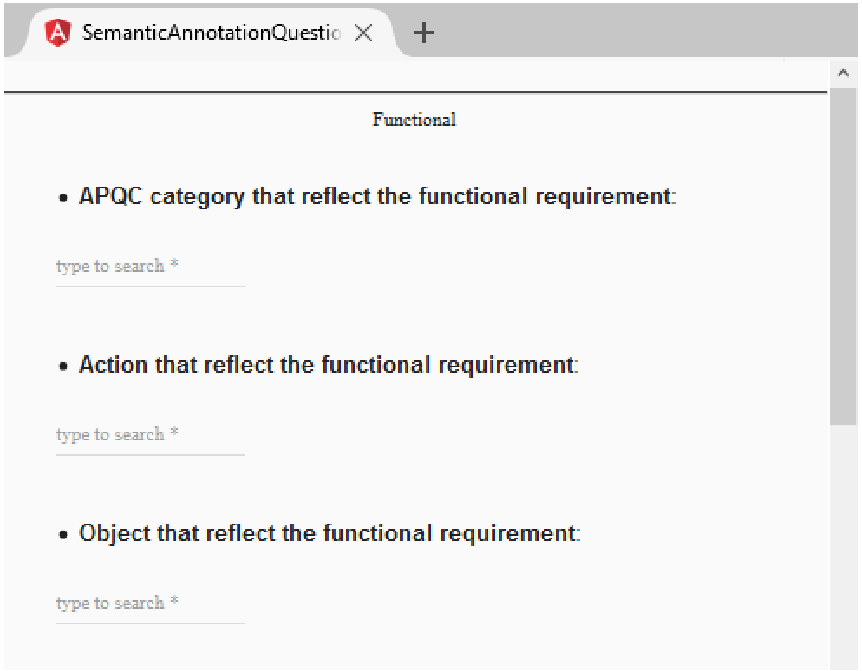


- owl:Thing
  - American Productivity and Quality Center
    - Acquire, Construct, and Manage Assets
    - Deliver Physical Products
    - Deliver Services
    - Develop and Manage Business Capabilities
    - Develop and Manage Human Capital
    - Develop and Manage Products and Services
    - Develop 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

Inferencing: Cloud Service Selection

Cloud Service Selection

Functionality



Functional

- **APQC category that reflect the functional requirement:**

type to search *

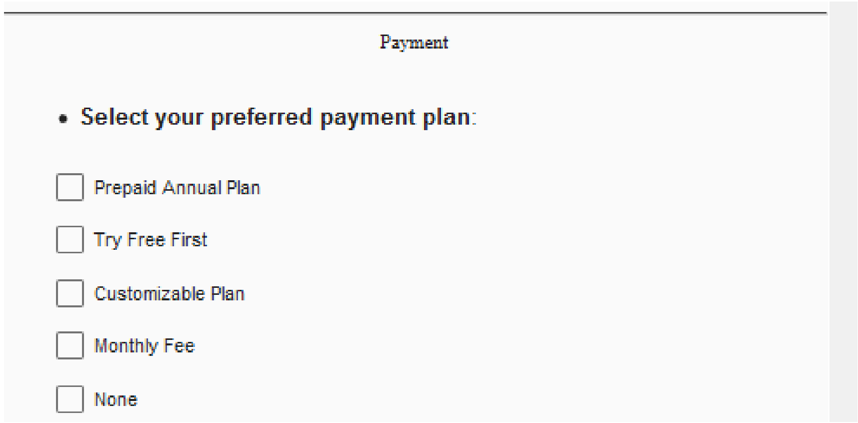
- **Action that reflect the functional requirement:**

type to search *

- **Object that reflect the functional requirement:**

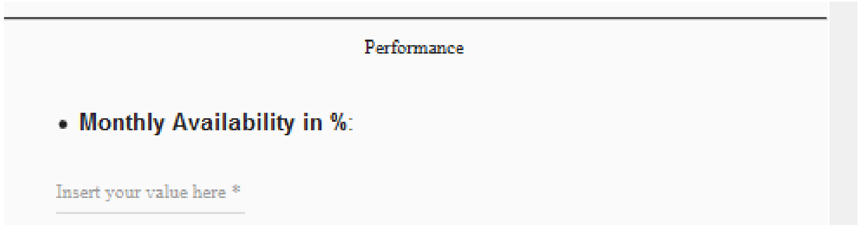
type to search *

Non-functional requirements


Payment

- **Select your preferred payment plan:**

□ Prepaid Annual Plan
□ Try Free First
□ Customizable Plan
□ Monthly Fee
□ None


Performance

- **Monthly Availability in %:**

Insert your value here *

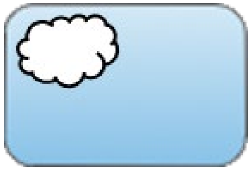
Thanks to Emanuele Laurenzi

Drawbacks of Semantic Lifting

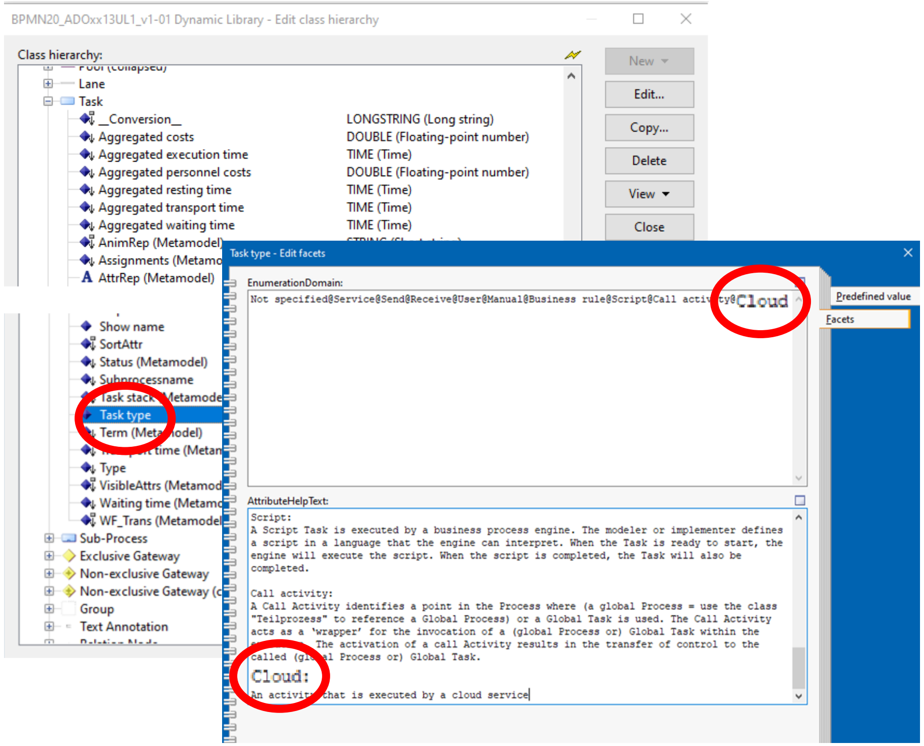
- Separate Environments for
  - Modelling
  - Knowledge Base (Inferencing)
- Inconsistency: Both metamodel and ontology must be aligned but are maintained independently:
  - Metamodel and ontology must represent the same semantics
  - Each change in metamodel must be reproduced in the ontology and vice 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:



BPMN20_ADOxx13UL1_v1-01 Dynamic Library - Edit class hierarchy

Class hierarchy:

- Pool (collapsed)
- Lane
- Task
  - __Conversion__ LONGSTRING (Long string)
  - Aggregated costs DOUBLE (Floating-point number)
  - Aggregated execution time TIME (Time)
  - Aggregated personnel costs DOUBLE (Floating-point number)
  - Aggregated resting time TIME (Time)
  - Aggregated transport time TIME (Time)
  - Aggregated waiting time TIME (Time)
  - AnimRep (Metamodel)
  - Assignments (Metamo
  - AttrRep (Metamodel)
  - Show name
  - SortAttr
  - Status (Metamodel)
  - Subprocessname
  - Task stack (Metamode
  - Task type
  - Term (Metamodel)
  - Transport time (Metam
  - Type
  - VisibleAttrs (Metamod
  - Waiting time (Metamo
  - WF_Trans (Metamodel
- Sub-Process
- Exclusive Gateway
- Non-exclusive Gateway
- Non-exclusive Gateway (c
- Group
- Text Annotation
- Relation Node

Task type - Edit facets

EnumerationDomain:

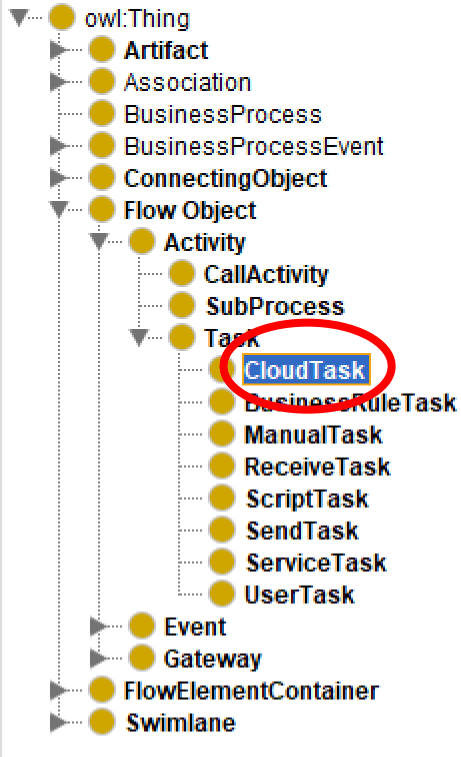
```
Not specified@Service@Send@Receive@User@Manual@Business rule@Script@Call activity@Cloud
```

AttributeHelpText:

```
Script:
A Script Task is executed by a business process engine. The modeler or implementer defines
a script in a language that the engine can interpret. When the Task is ready to start, the
engine will execute the script. When the script is completed, the Task will also be
completed.

Call activity:
A Call Activity identifies a point in the Process where (a global Process = use the class
"Teilprozess" to reference a Global Process) or a Global Task is used. The Call Activity
acts as a 'wrapper' for the invocation of a (global Process or) Global Task within the
execution. The activation of a call Activity results in the transfer of control to the
called (global Process or) Global Task.
Cloud:
An activity that is executed by a cloud service
```

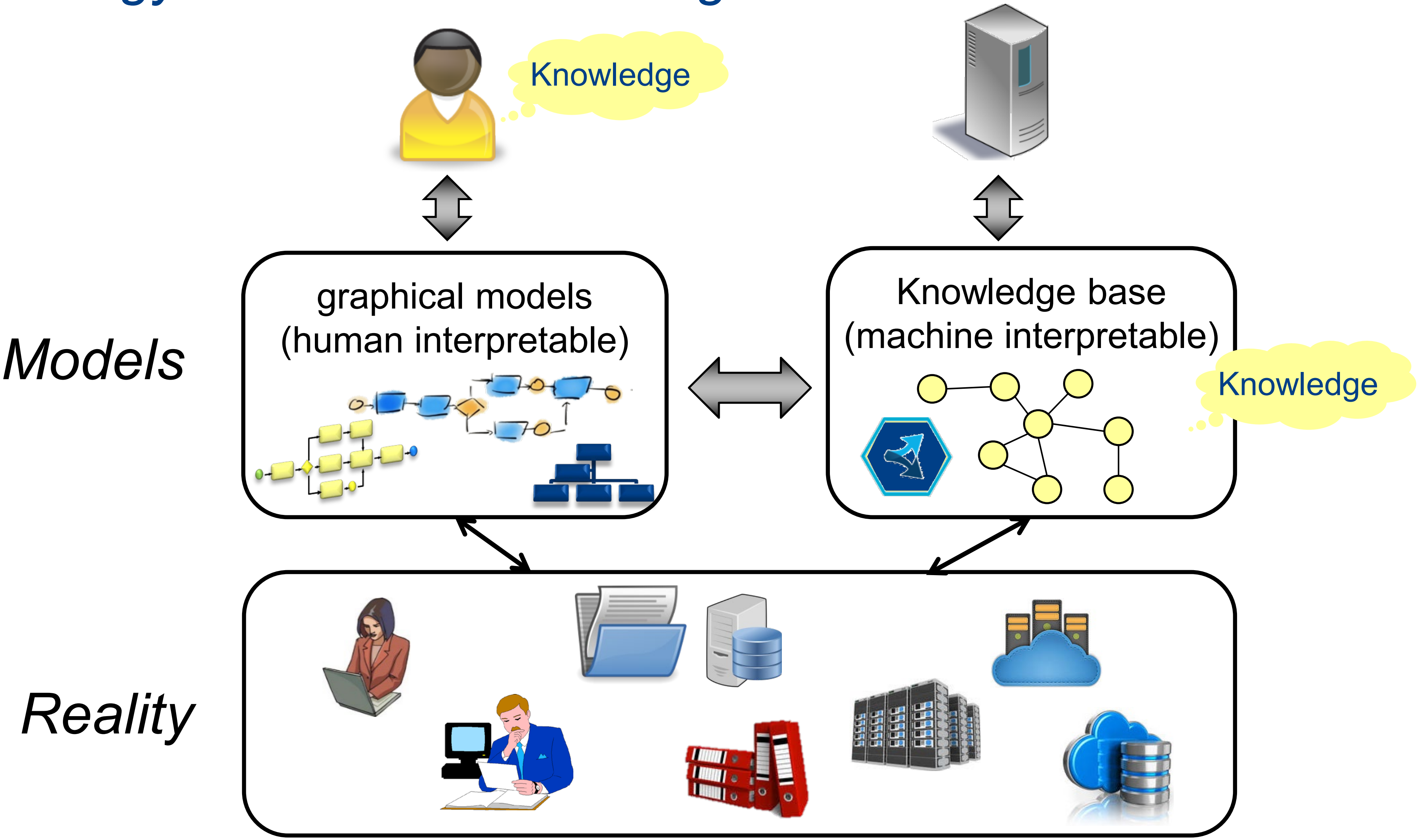
Change in the ontology:



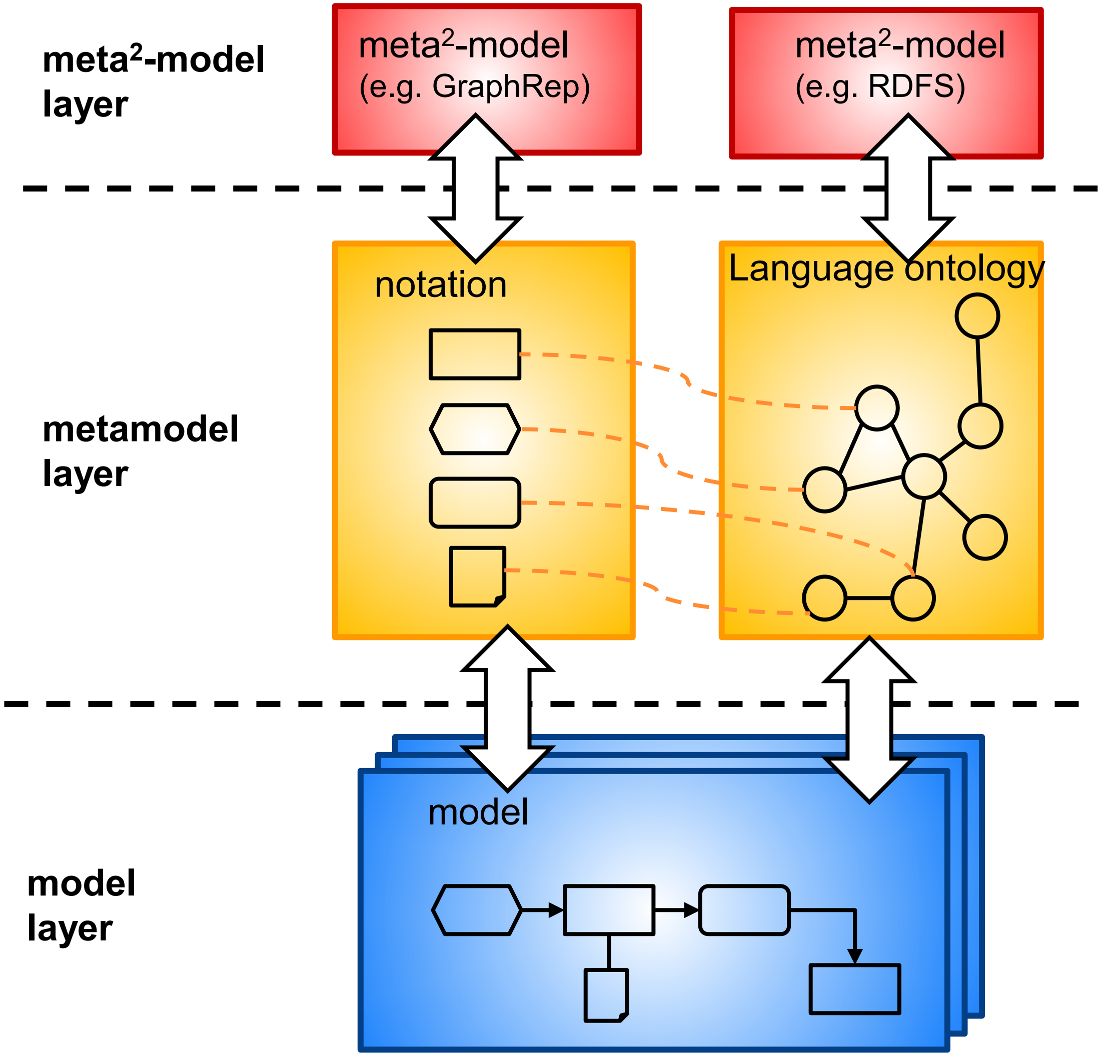
- owl:Thing
  - Artifact
  - Association
  - BusinessProcess
  - BusinessProcessEvent
  - ConnectingObject
  - Flow Object
    - Activity
      - CallActivity
      - SubProcess
      - Task
        - CloudTask
        - BusinessRuleTask
        - ManualTask
        - ReceiveTask
        - ScriptTask
        - SendTask
        - ServiceTask
        - UserTask
    - Event
    - Gateway
  - FlowElementContainer
  - Swimlane

Ontology-based Metamodelling

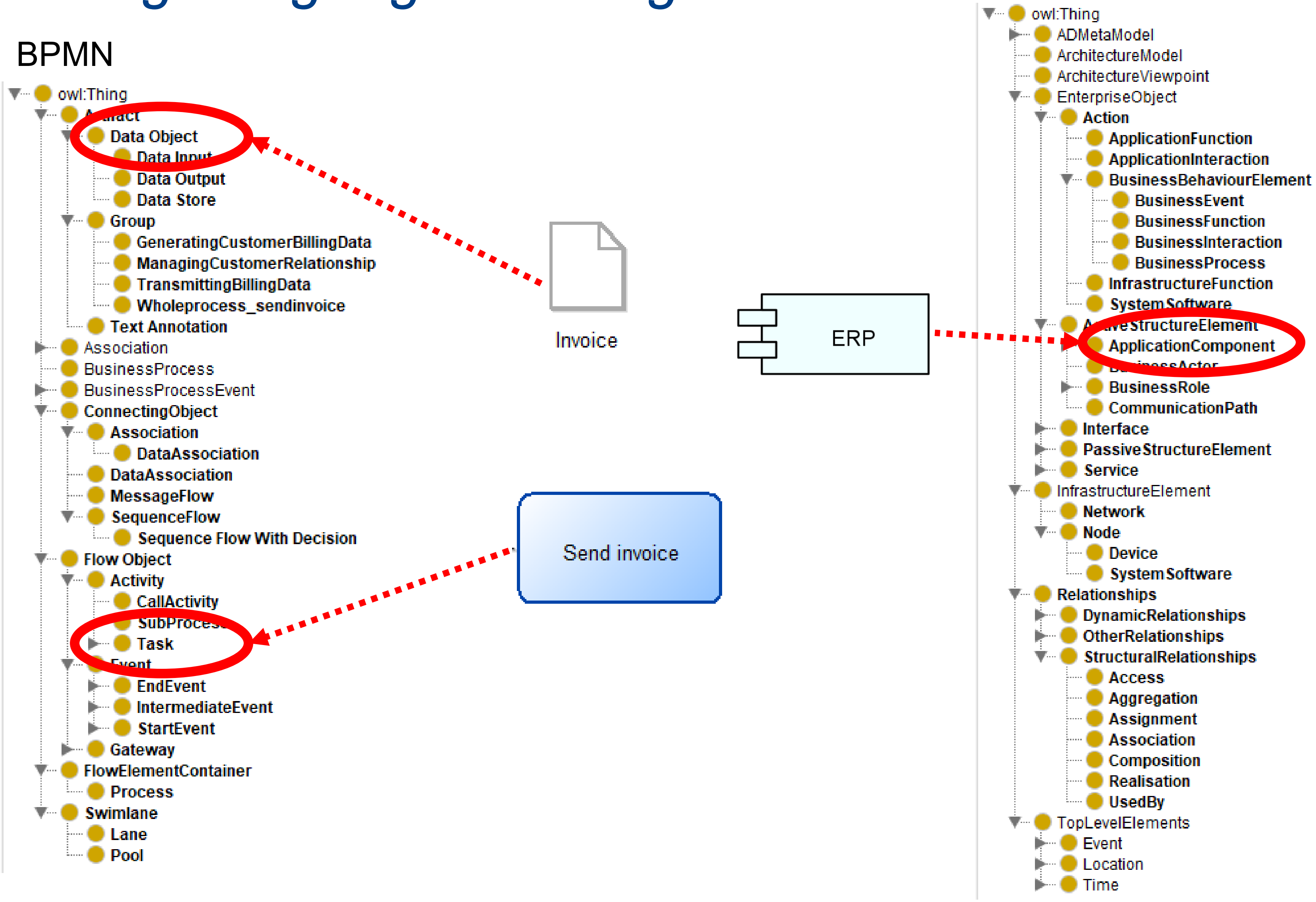
Ontology-based Metamodeling



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



Modelling Language Ontologies

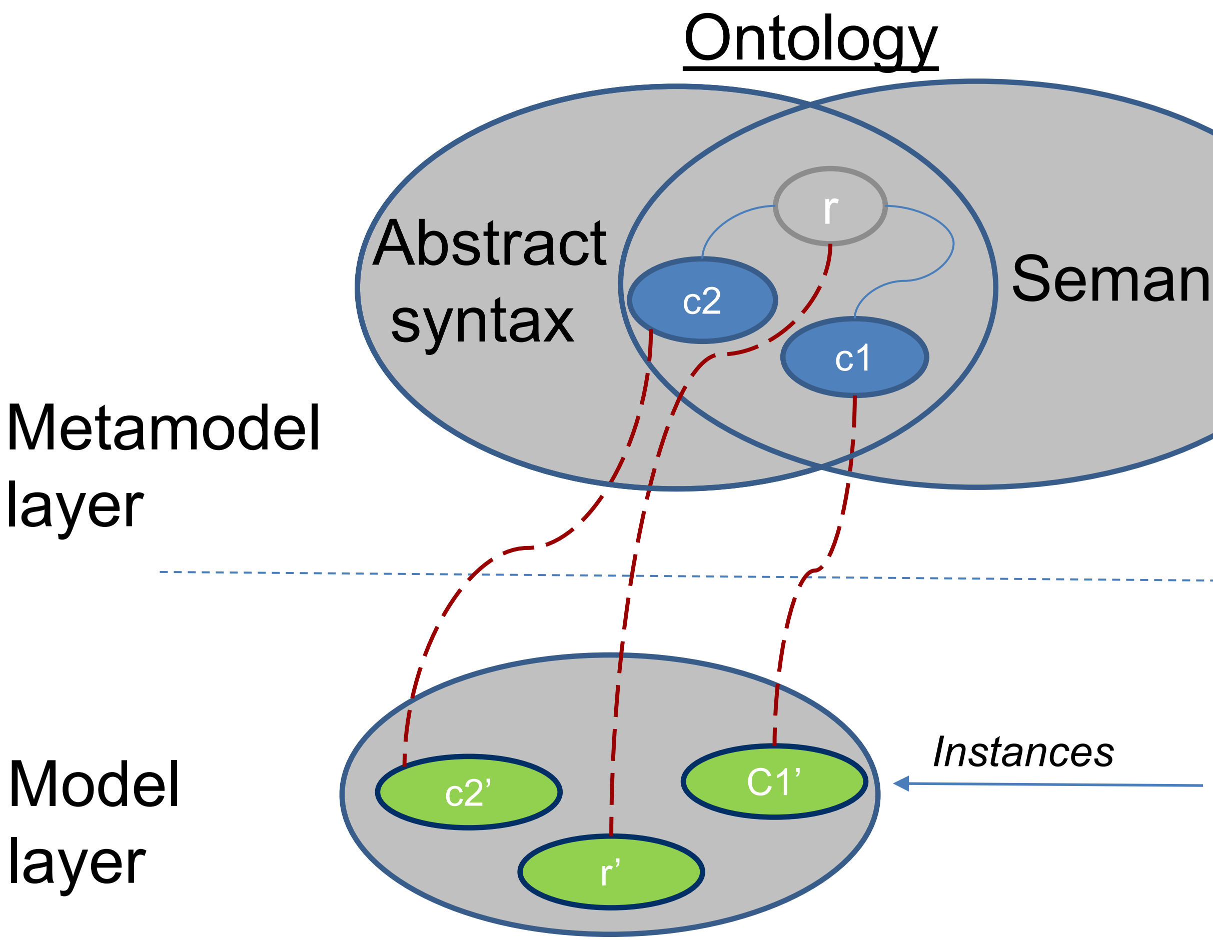


BPMN

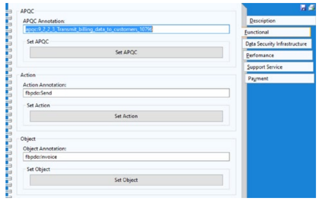
- owl:Thing
  - Artifact
    - Data Object
      - Data Input
      - Data Output
      - Data Store
    - Group
      - GeneratingCustomerBillingData
      - ManagingCustomerRelationship
      - TransmittingBillingData
      - Wholeprocess_sendinvoice
    - Text Annotation
  - Association
  - BusinessProcess
  - 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
      - System Software
    - 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



Model


APQC

APQC Annotation: apqc:9_2_2_3_Transmit_billing_data_to_customers_10796

Set APQC

Action

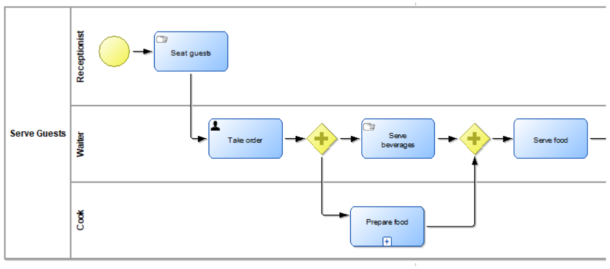
Action Annotation: fbpdo:Send

Set Action

Object

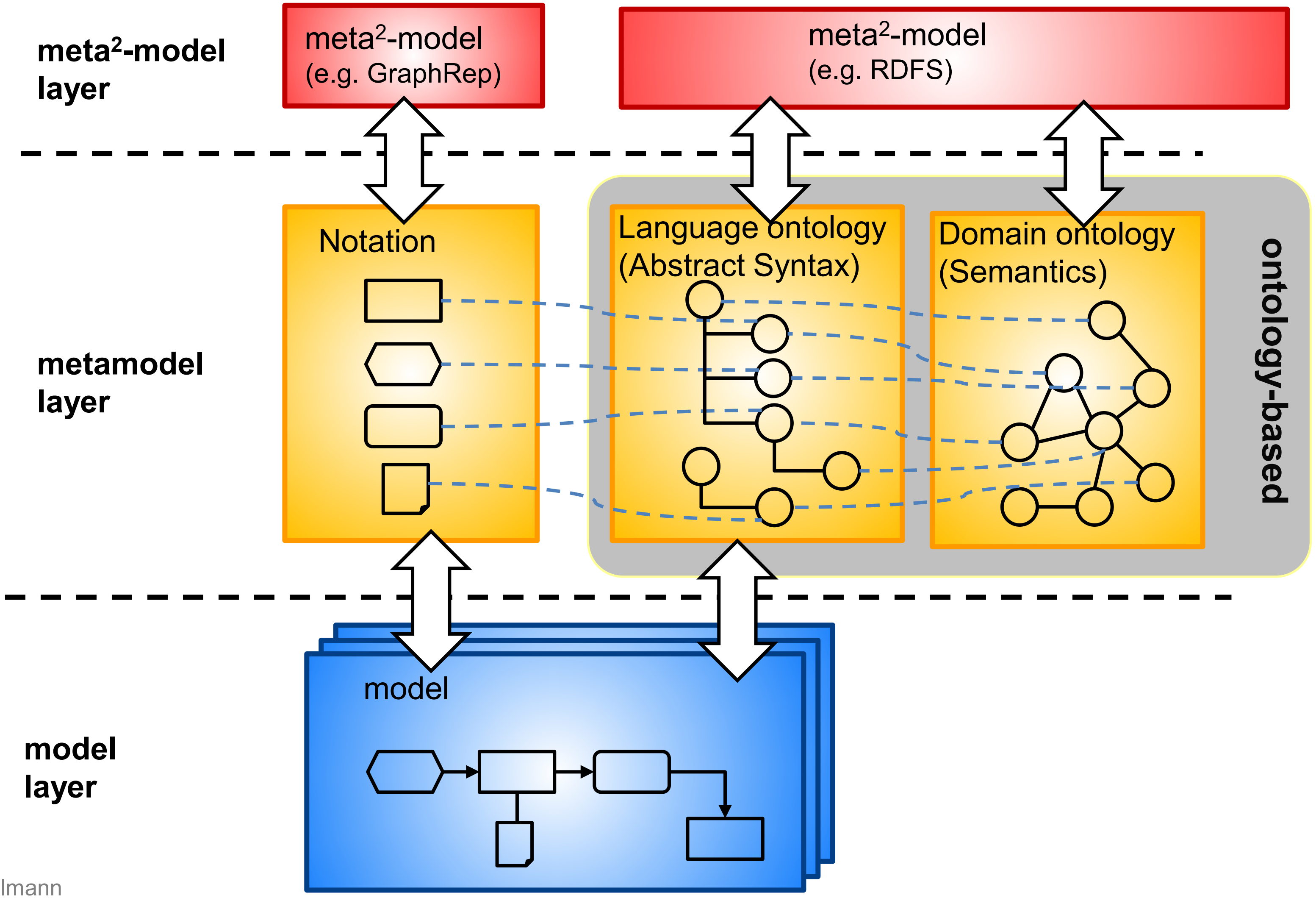
Object Annotation: fbpdo:Invoice

Set Object



Thanks to Emanuele Laurenzi

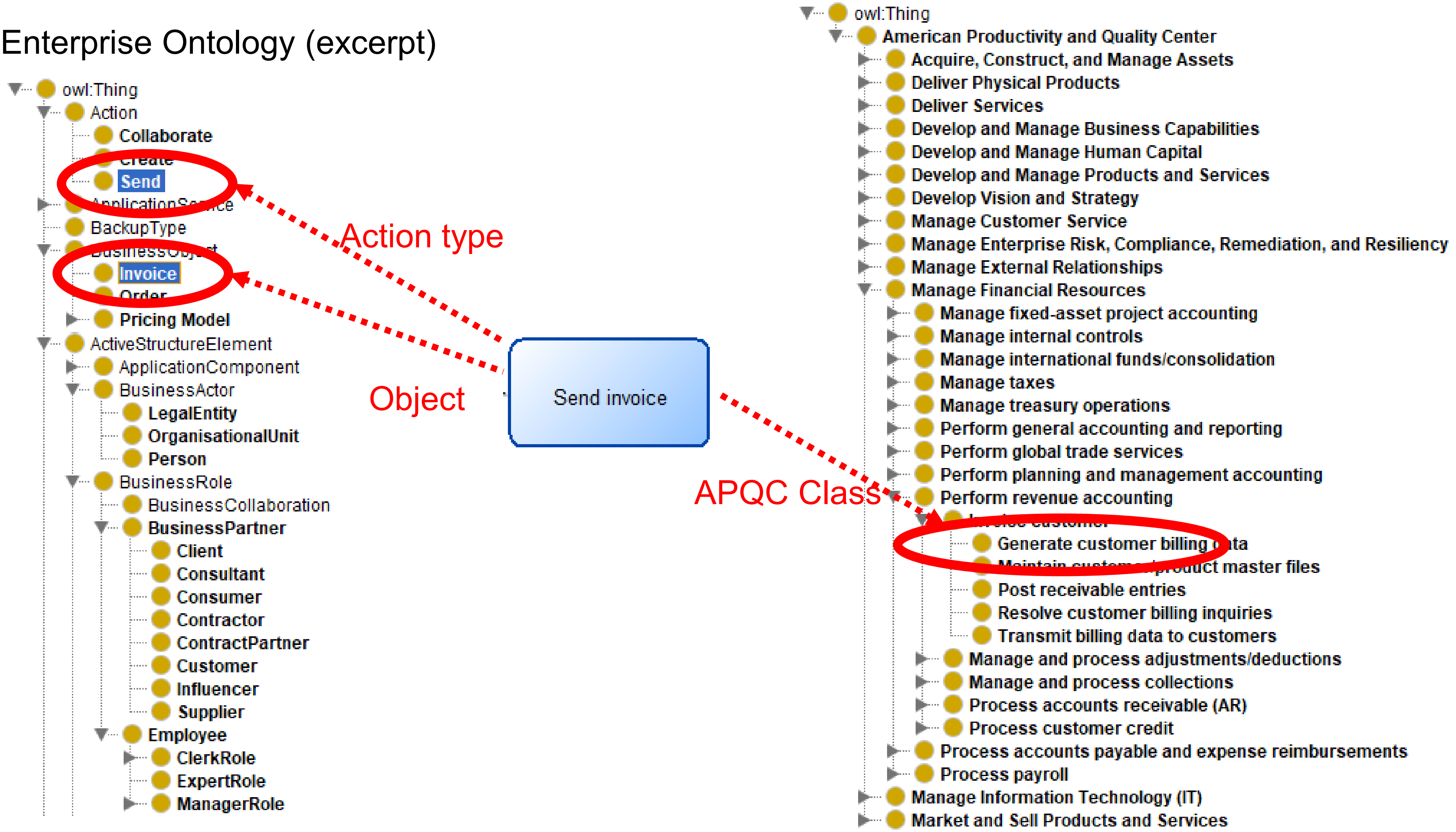
Ontology-based Metamodeling (2): Ontologies for Metamodel and Content



Domain Ontologies

Domain Ontology:
APQC Process Classification Framework

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

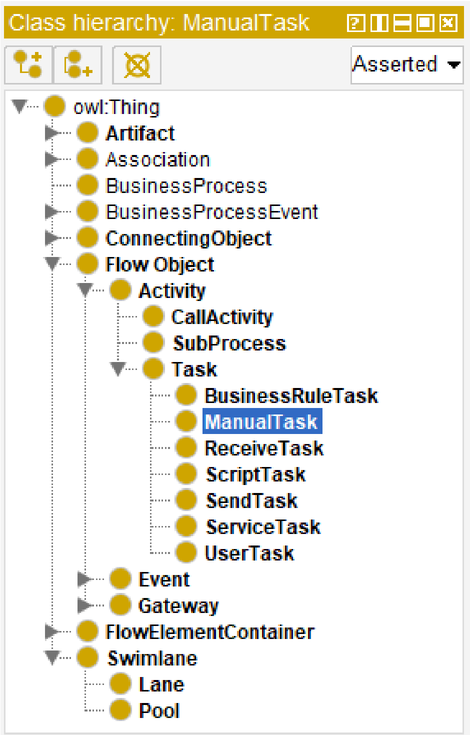
Send invoice

APQC Class

- 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

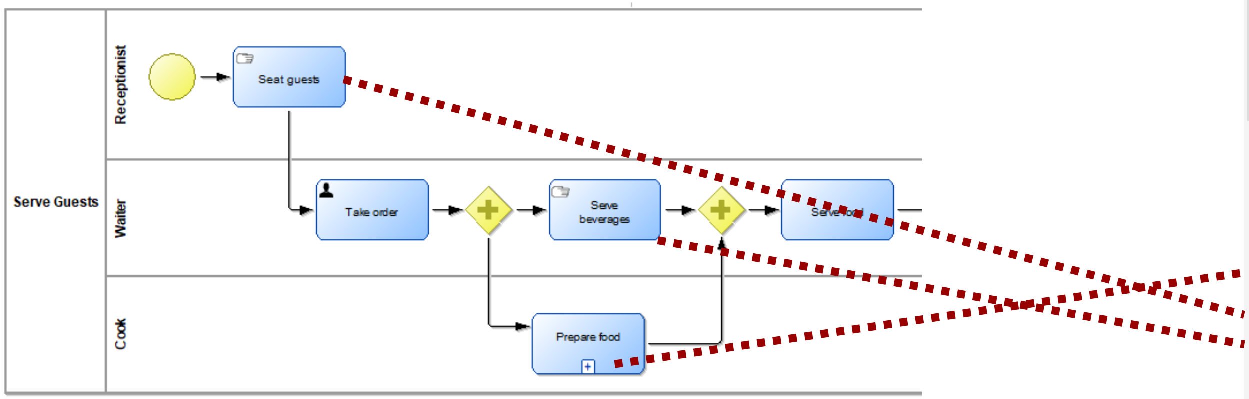
- Single environment for modelling and ontology
- Model elements are directly created as instances in the ontology

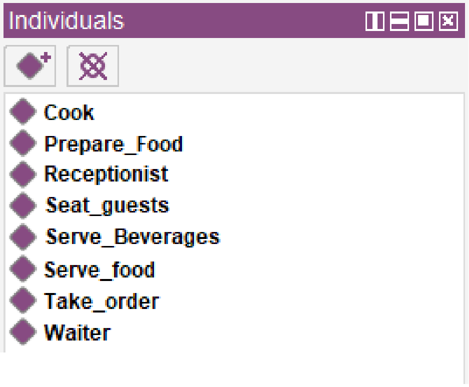


Class hierarchy: ManualTask

Asserted

- owl:Thing
  - Artifact
  - Association
  - BusinessProcess
  - BusinessProcessEvent
  - ConnectingObject
  - Flow Object
    - Activity
      - CallActivity
      - SubProcess
      - Task
        - BusinessRuleTask
        - ManualTask
        - ReceiveTask
        - ScriptTask
        - SendTask
        - ServiceTask
        - UserTask
    - Event
    - Gateway
  - FlowElementContainer
  - Swimlane
    - Lane
    - Pool

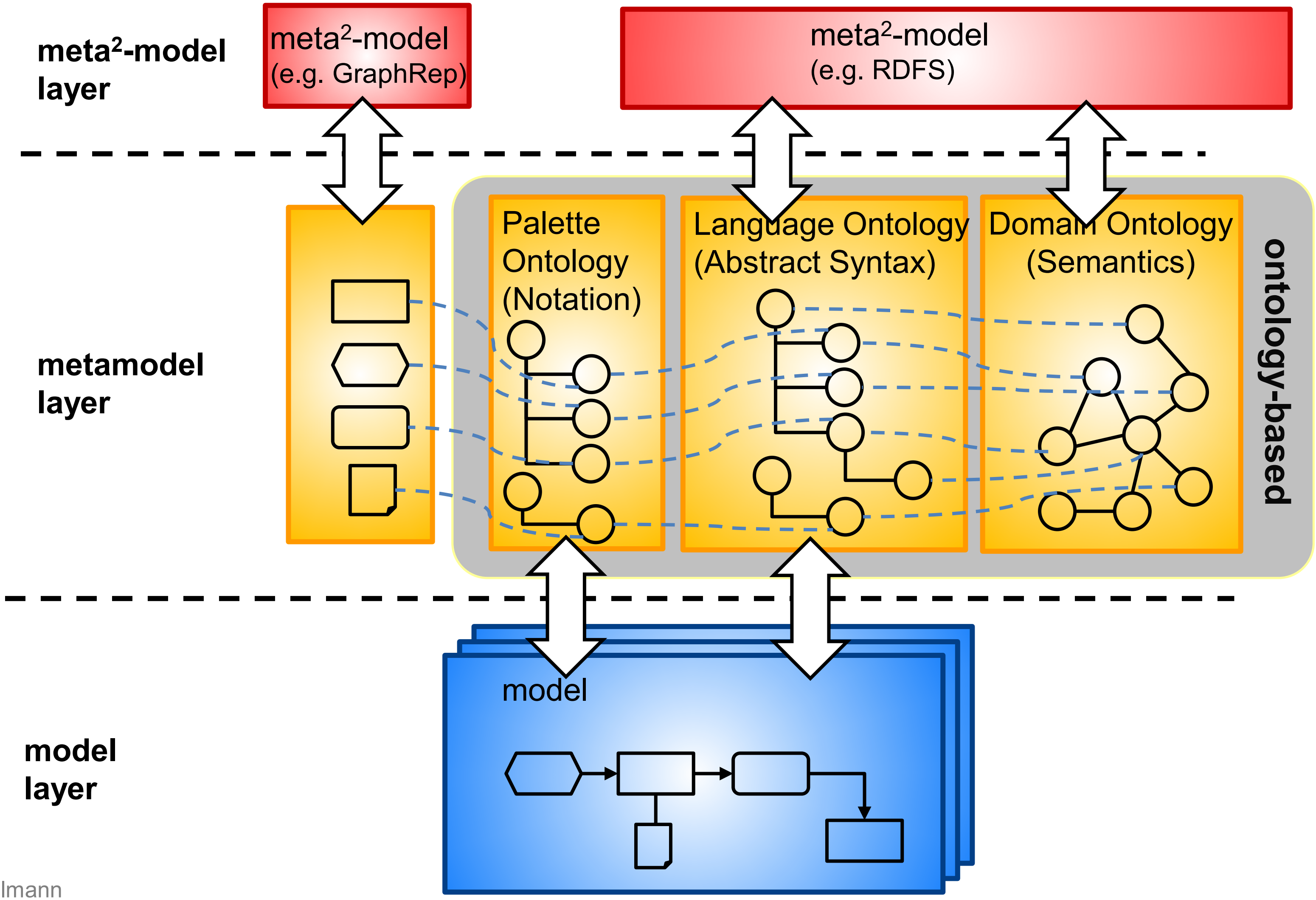




Individuals

- Cook
- Prepare_Food
- Receptionist
- Seat_guests
- Serve_Beverages
- Serve_food
- Take_order
- Waiter

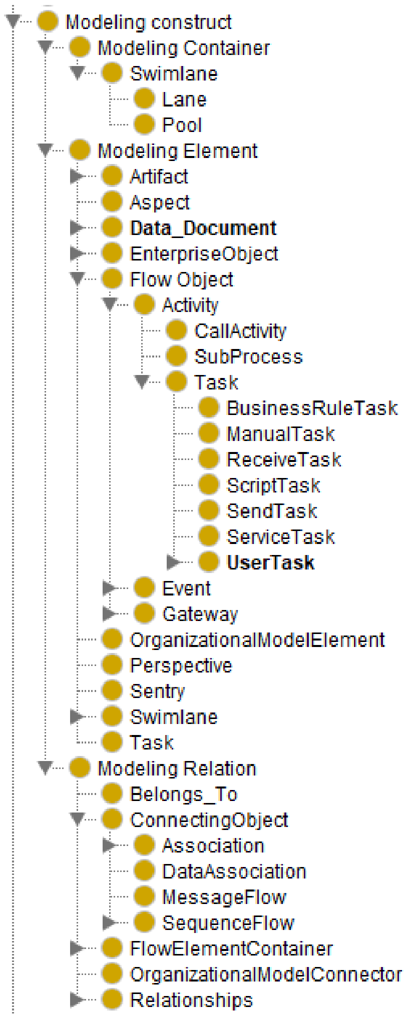
Ontology-based Metamodeling (3): Ontologies for Language, Metamodel and Content



meta²-model layer
meta²-model (e.g. GraphRep)
meta²-model (e.g. RDFS)
metamodel layer
Palette Ontology (Notation)
Language Ontology (Abstract Syntax)
Domain Ontology (Semantics)
ontology-based
model layer
model

Palette Ontology

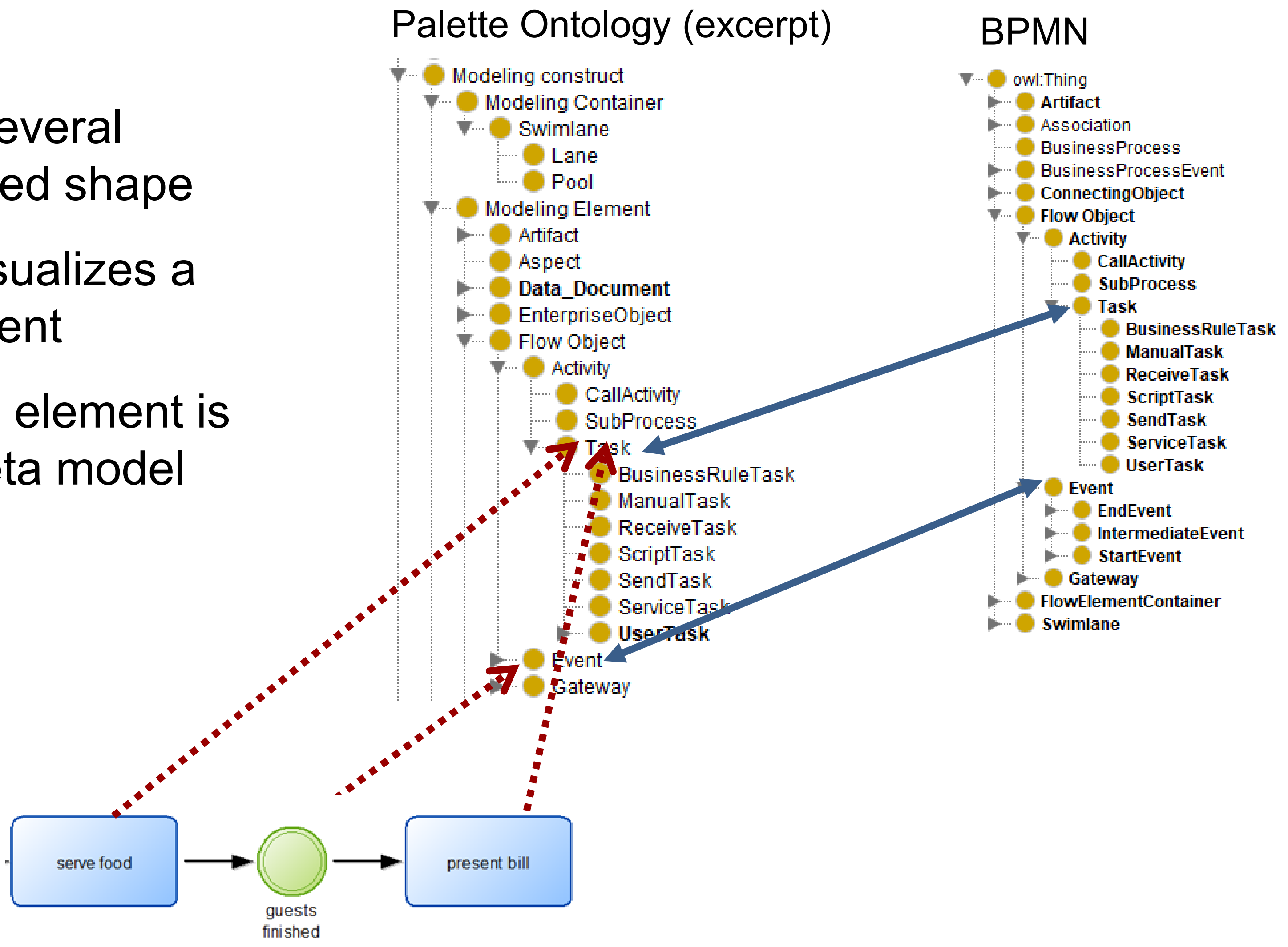
Palette Ontology (excerpt)



- 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



Palette Ontology (excerpt)

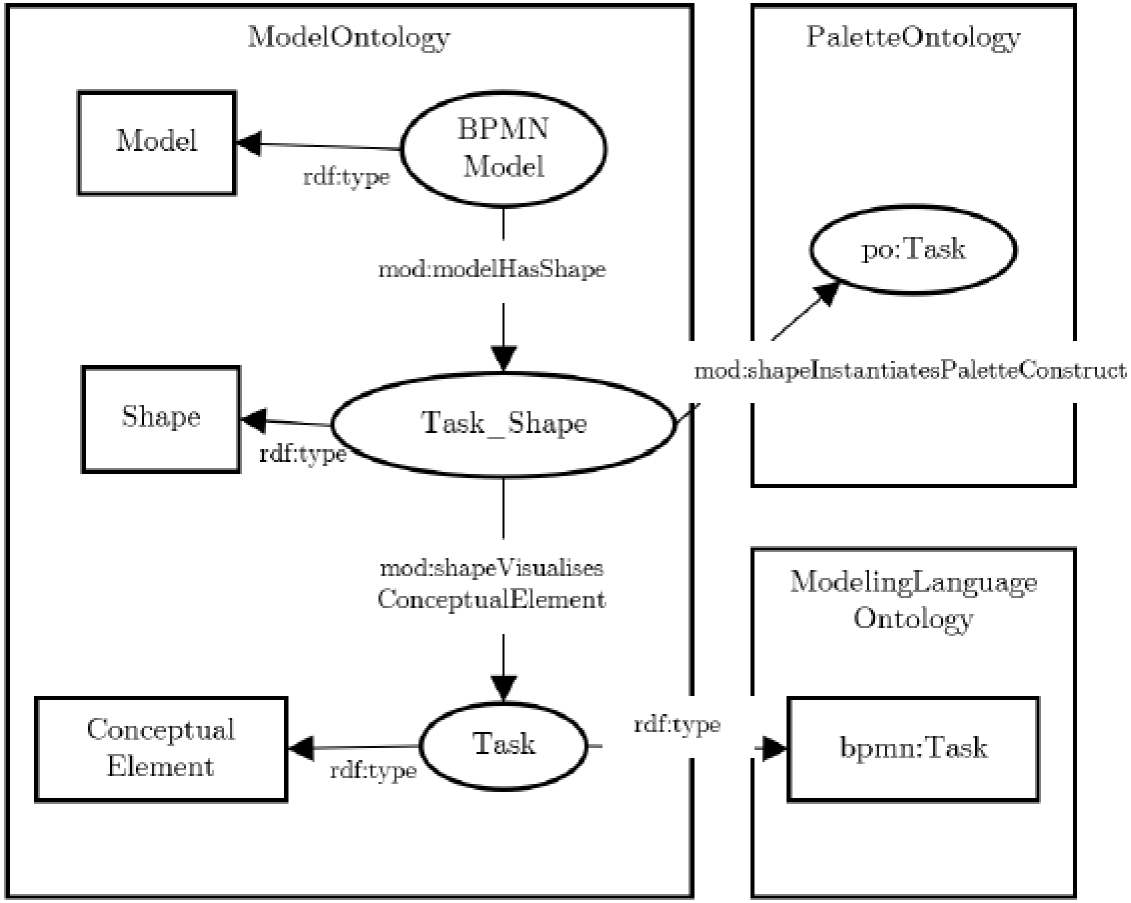
- Modeling construct
  - Modeling Container
    - Swimlane
      - Lane
      - Pool
  - Modeling Element
    - Artifact
    - Aspect
    - **Data_Document**
    - EnterpriseObject
    - Flow Object
      - Activity
        - CallActivity
        - SubProcess
        - Task
          - BusinessRuleTask
          - ManualTask
          - ReceiveTask
          - ScriptTask
          - SendTask
          - ServiceTask
          - **UserTask**
      - Event
      - Gateway

BPMN

- owl:Thing
  - **Artifact**
  - Association
  - BusinessProcess
  - BusinessProcessEvent
  - **ConnectingObject**
  - **Flow Object**
    - **Activity**
      - **CallActivity**
      - **SubProcess**
      - **Task**
        - **BusinessRuleTask**
        - **ManualTask**
        - **ReceiveTask**
        - **ScriptTask**
        - **SendTask**
        - **ServiceTask**
        - **UserTask**
    - **Event**
      - **EndEvent**
      - **IntermediateEvent**
      - **StartEvent**
    - **Gateway**
  - **FlowElementContainer**
  - **Swimlane**

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

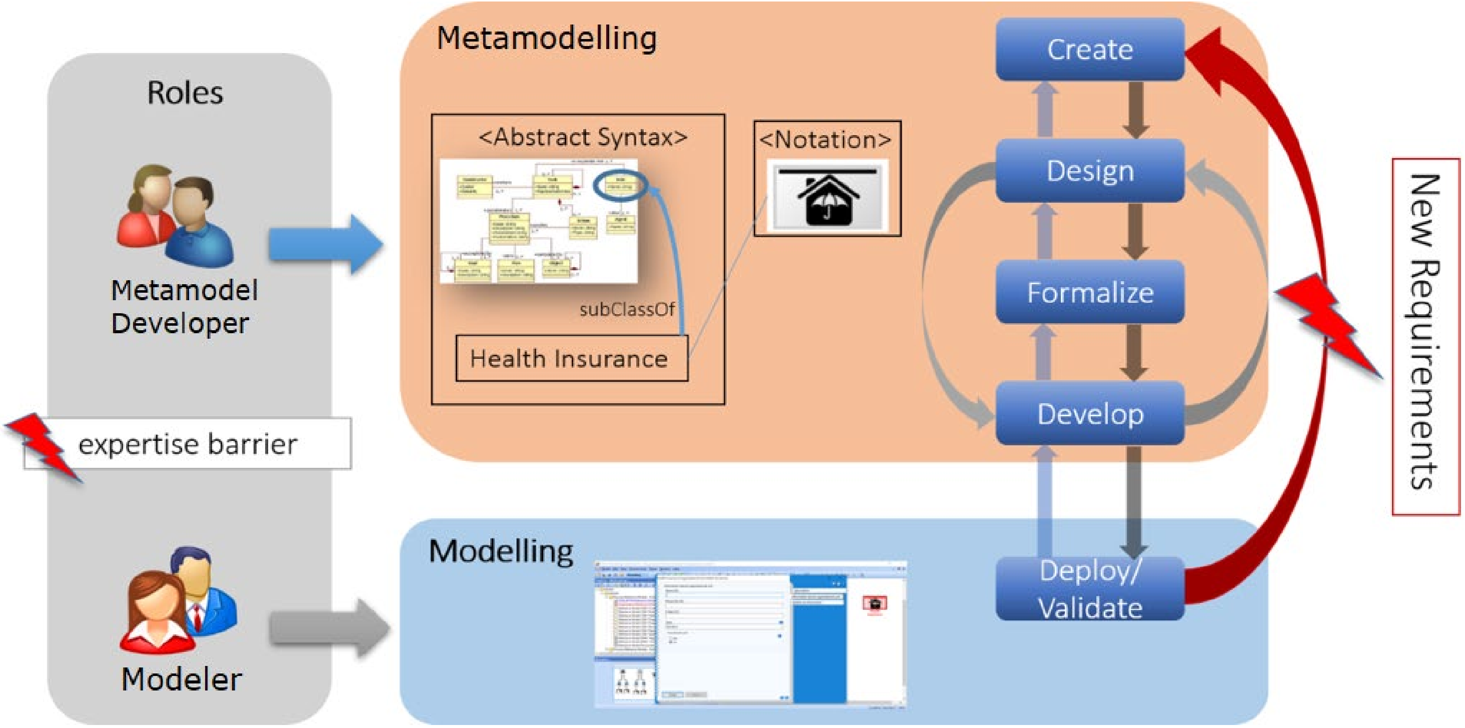
Filter the elements for BPMN Tasks and show the labels

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

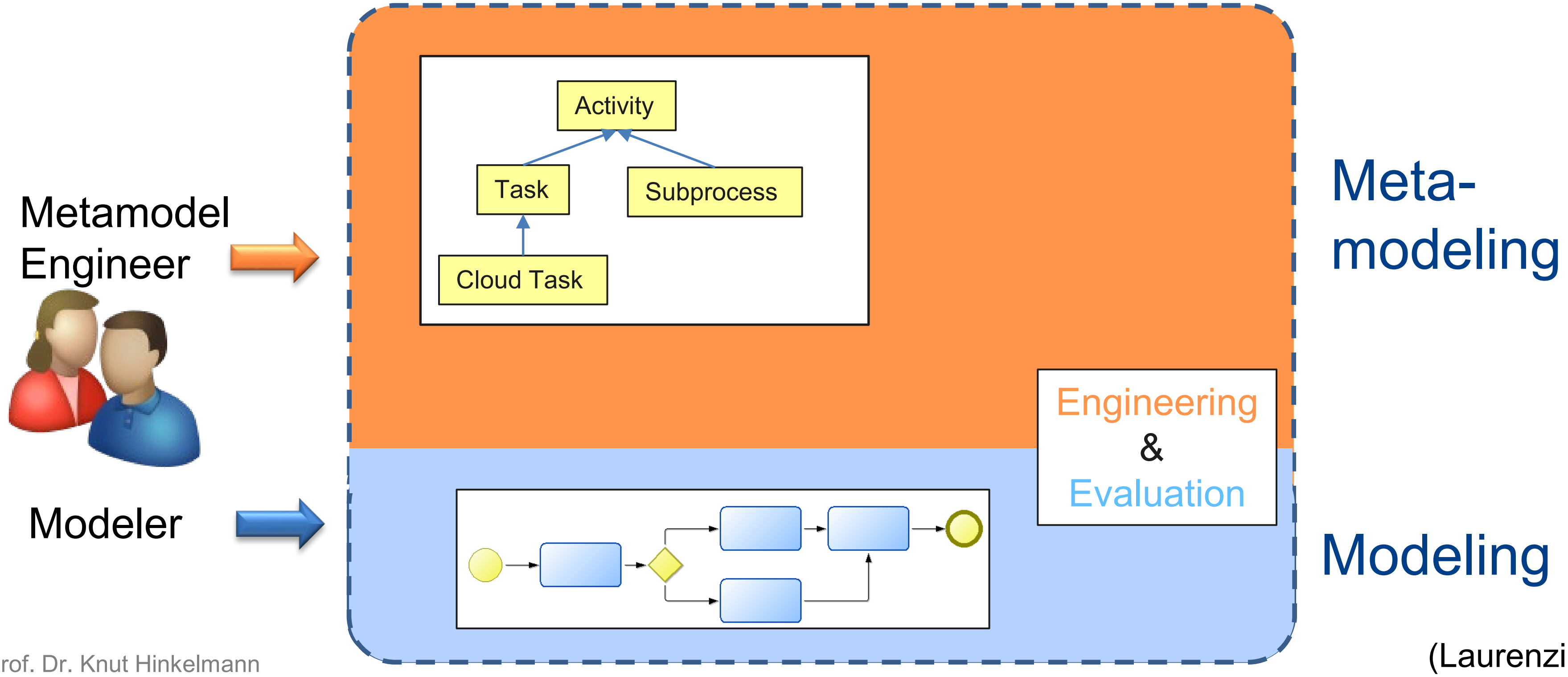


- Challenge 1: Metamodeling is a joint effort between metamodel experts and domain experts
- Challenge 2: Sequentialization of metamodeling and modeling is time consuming

Thanks to Emanuele Laurenzi

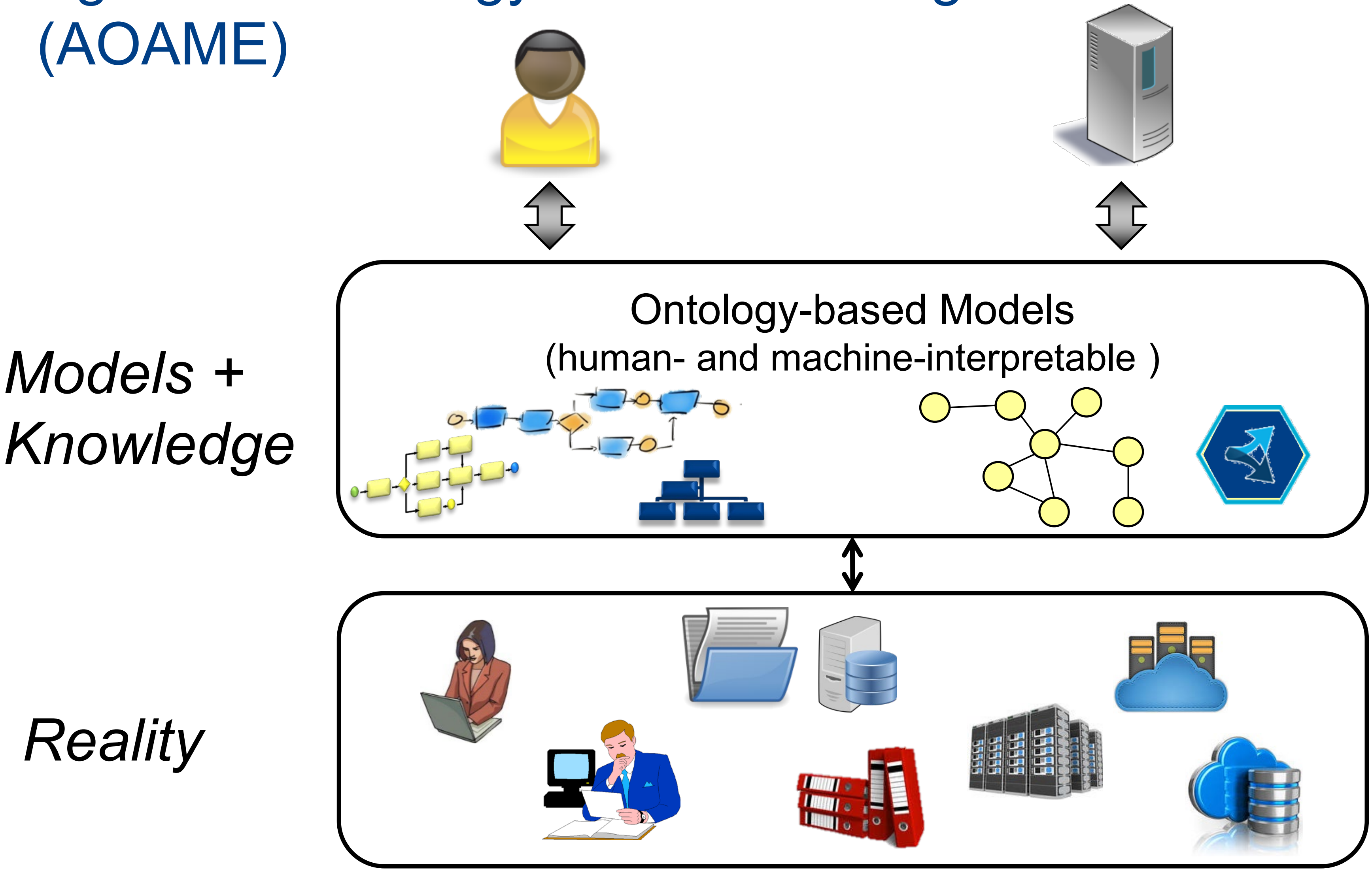
Integration Modeling and Metamodeling in a Single Environment

- 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)

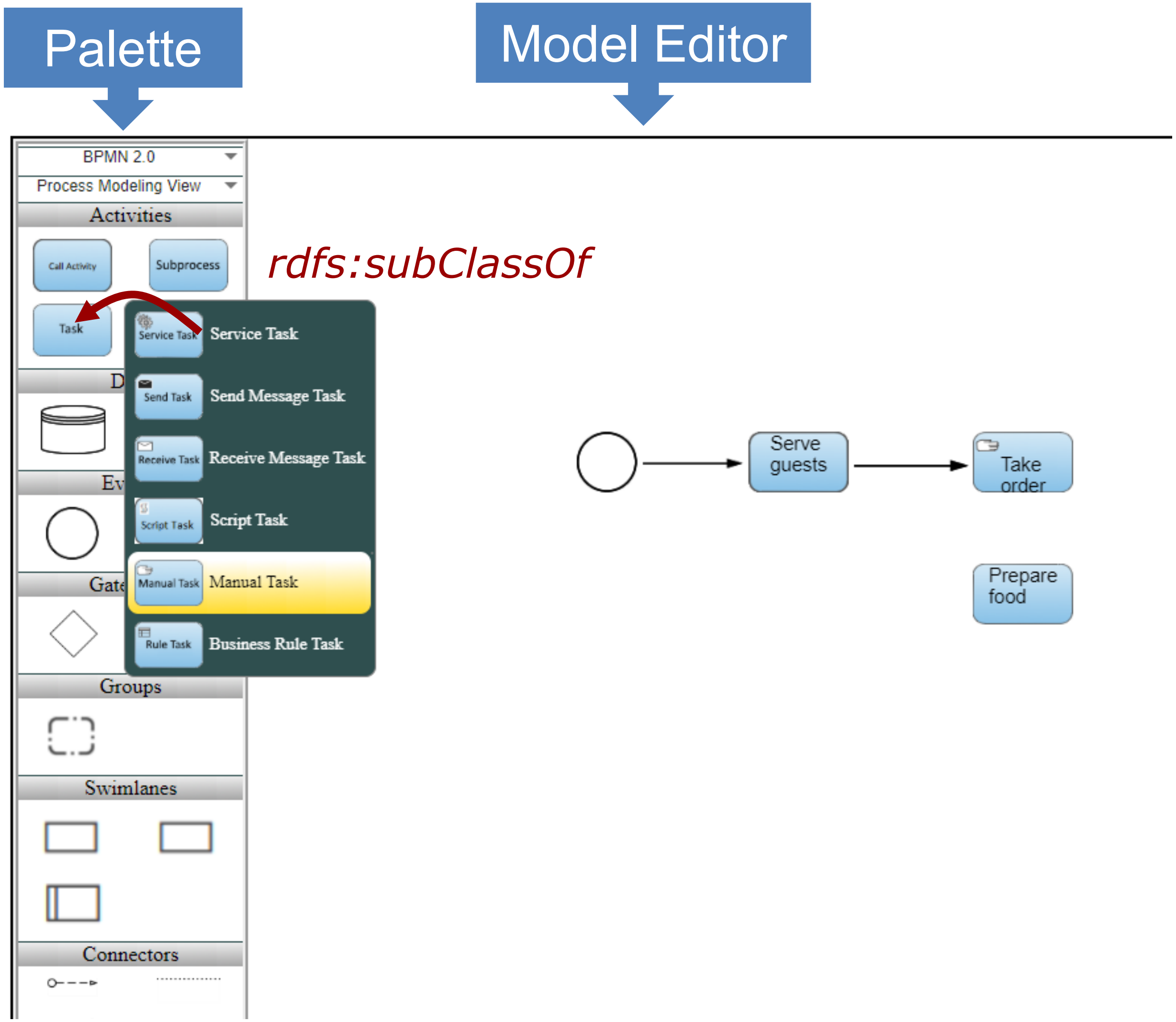


(Laurenzi et al. 2018)

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

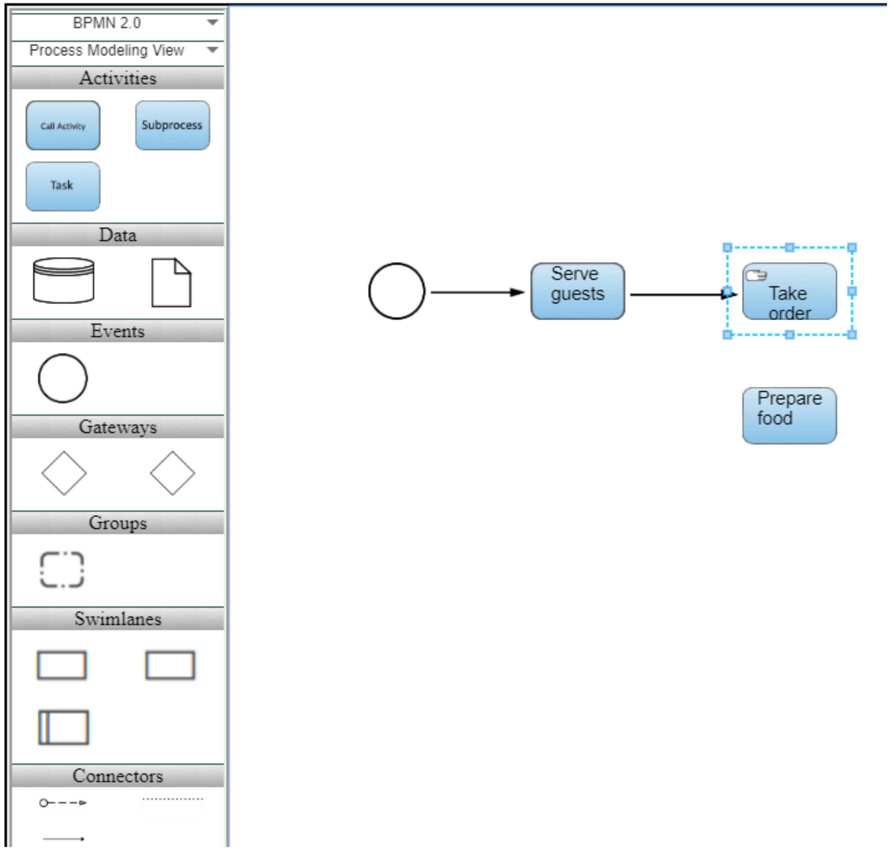
Ontology-Based Modeling in AOAME

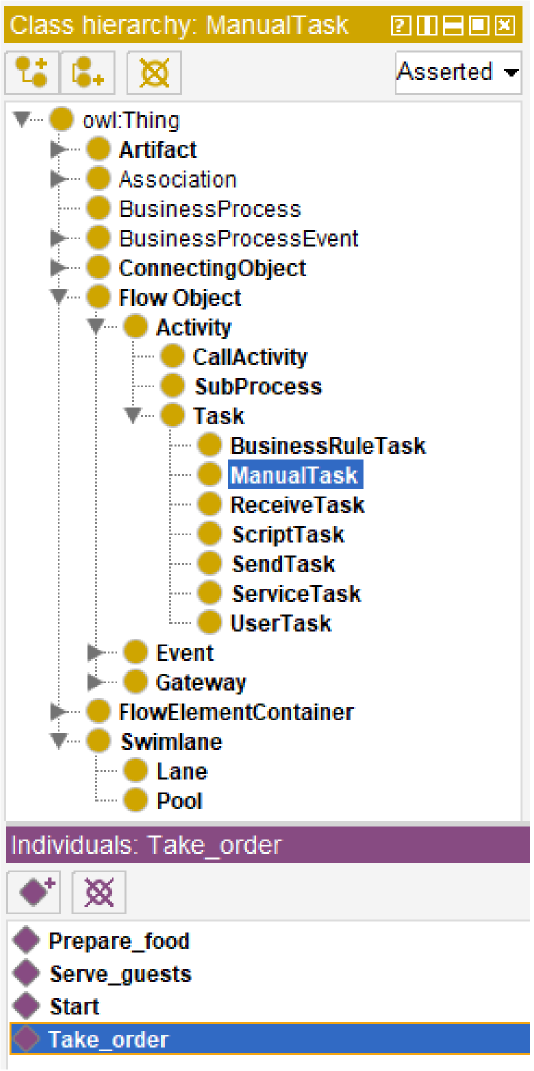


Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Ontology-Based Modelling

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




Class hierarchy: ManualTask

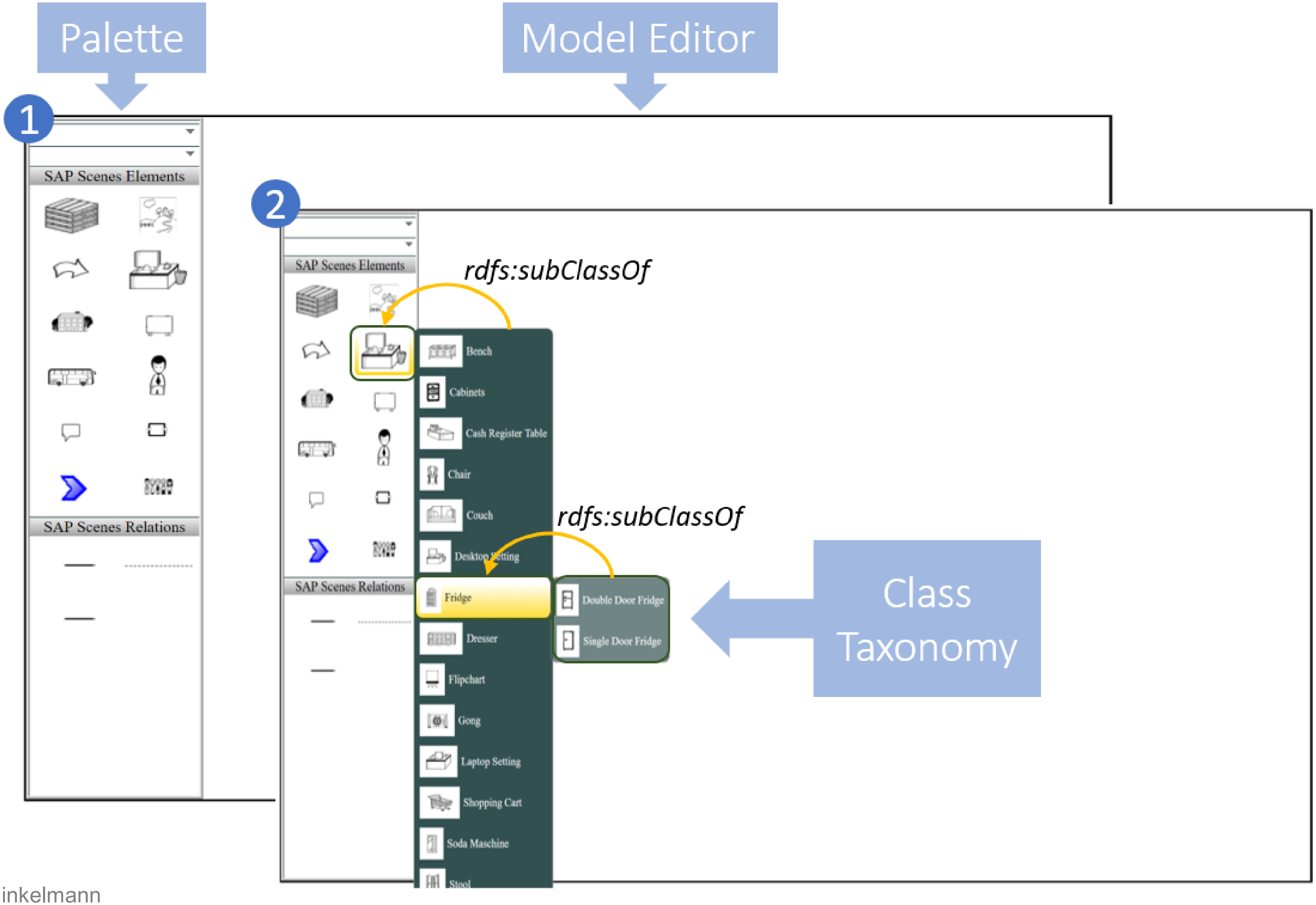
Asserted

- owl:Thing
  - Artifact
  - Association
  - BusinessProcess
  - BusinessProcessEvent
  - ConnectingObject
  - Flow Object
    - Activity
      - CallActivity
      - SubProcess
      - Task
        - BusinessRuleTask
        - ManualTask
        - ReceiveTask
        - ScriptTask
        - SendTask
        - ServiceTask
        - UserTask
    - Event
    - Gateway
  - FlowElementContainer
  - Swimlane
    - Lane
    - Pool

Individuals: Take_order

- Prepare_food
- Serve_guests
- Start
- Take_order

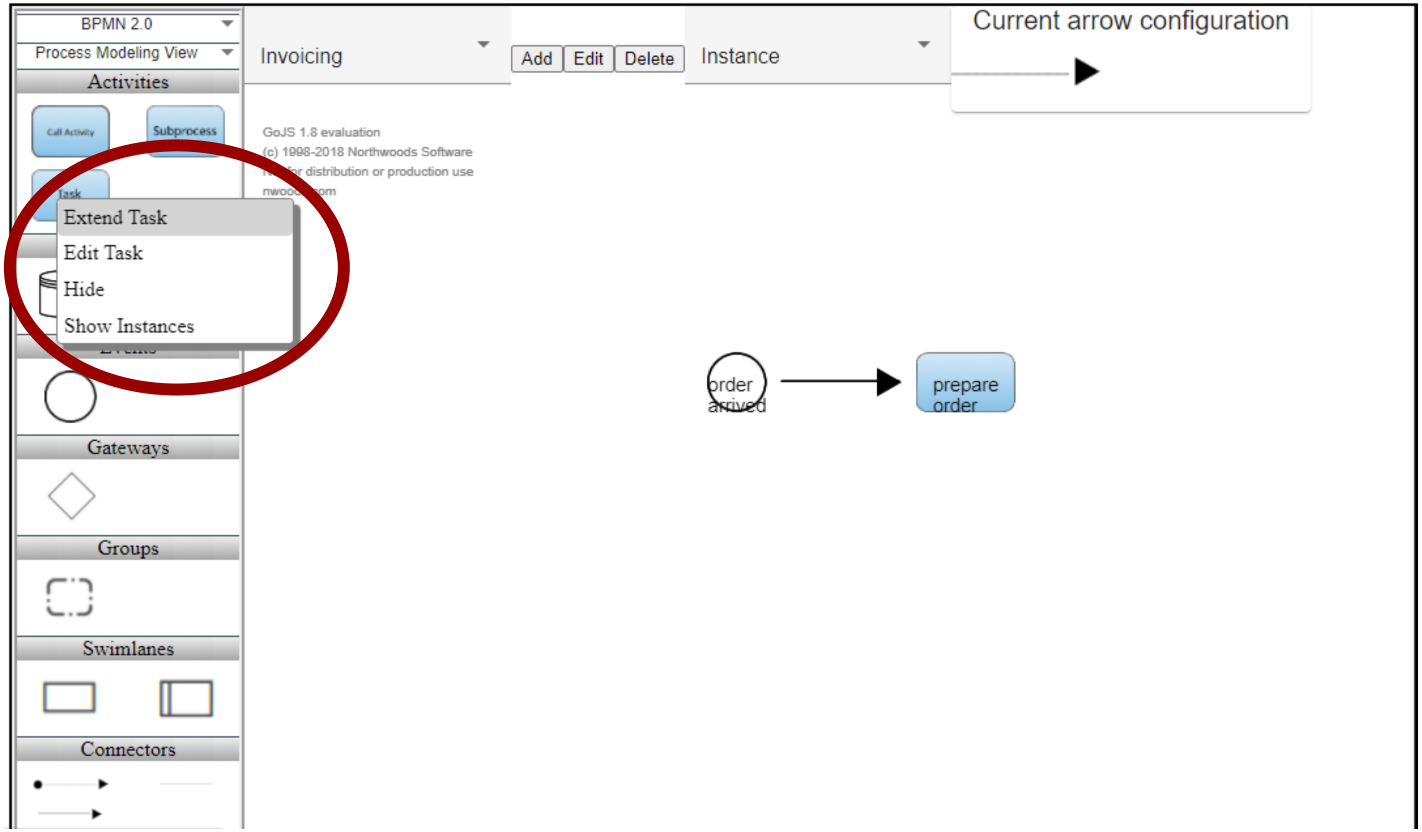
Modeling Elements are represented in a Class Hierarchy



Extending AOAME Modeling Languages – on the fly

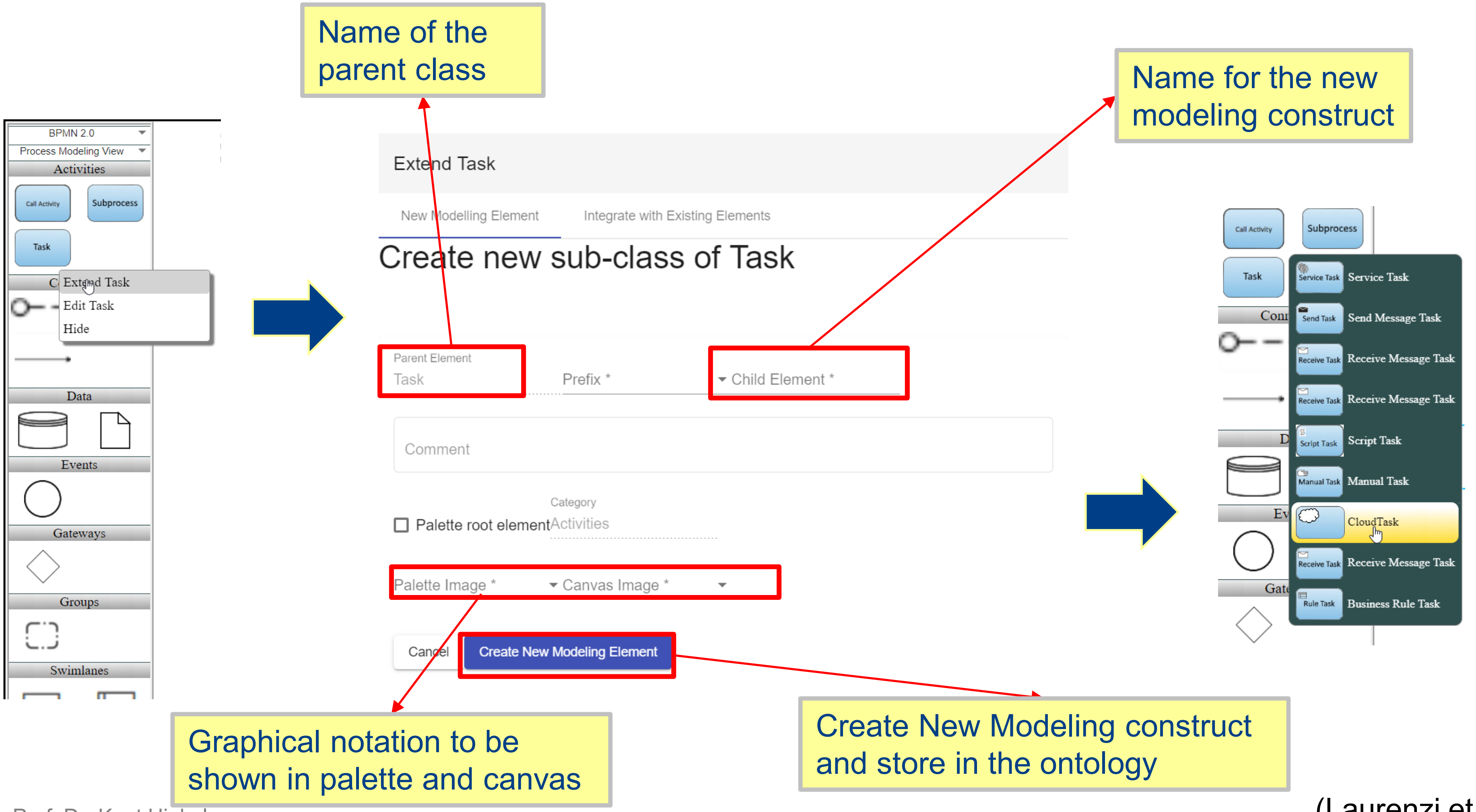
Palette

Model Editor



GoJS 1.8 evaluation
(c) 1998-2018 Northwoods Software
N... for distribution or production use
nwoo...om

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



Name of the parent class

Name for the new modeling construct

Extend Task

New Modelling Element | Integrate with Existing Elements

Create new sub-class of Task

Parent Element: Task

Prefix *

Child Element *

Comment

☐ Palette root element

Category: Activities

Palette Image *

Canvas Image *

Cancel

Create New Modeling Element

Graphical notation to be shown in palette and canvas

Create New Modeling construct and store in the ontology

(Laurenzi et al. 2018)

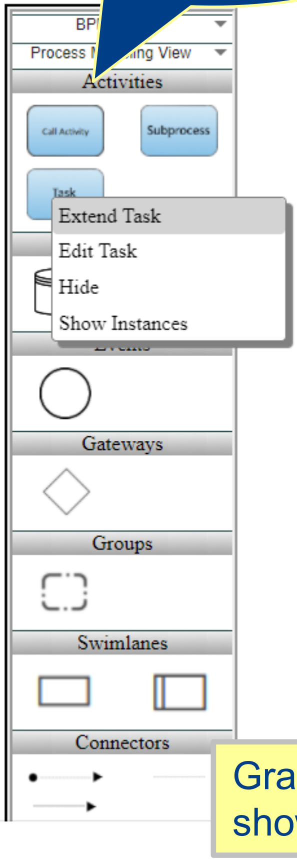
Integration of Meta-modeling and Modeling: On-the-fly Modeling Language Adaptation

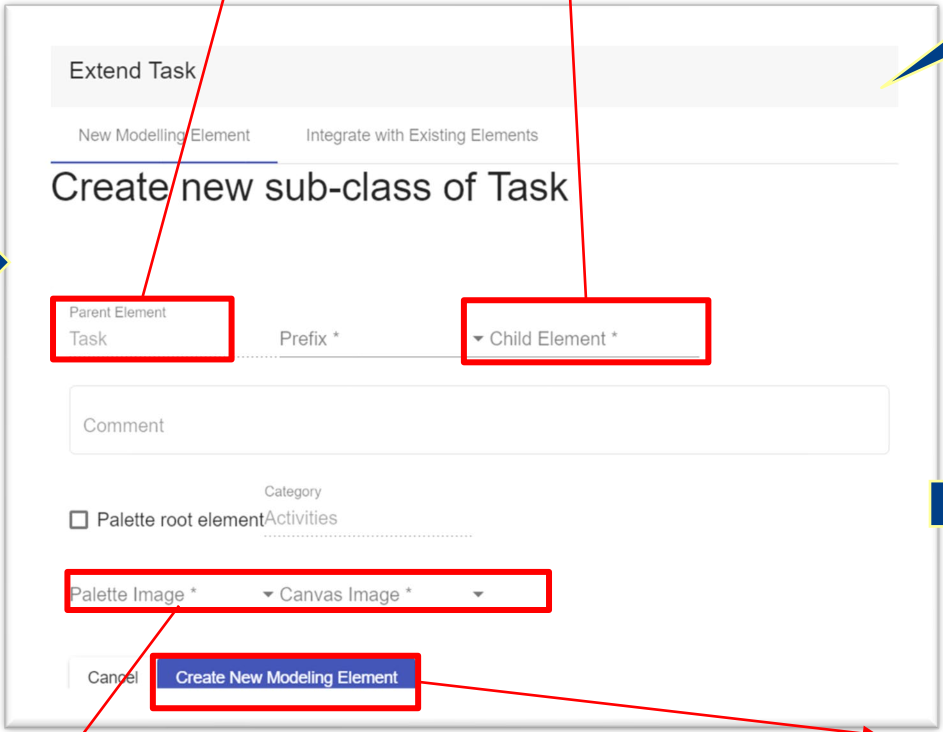
Ontology-based palette

Name of the parent class

Name for the new modeling construct

Ontology-based metamodel





Extend Task

New Modelling Element | Integrate with Existing Elements

Create new sub-class of Task

Parent Element: Task

Prefix *

Child Element *

Comment

☐ Palette root element

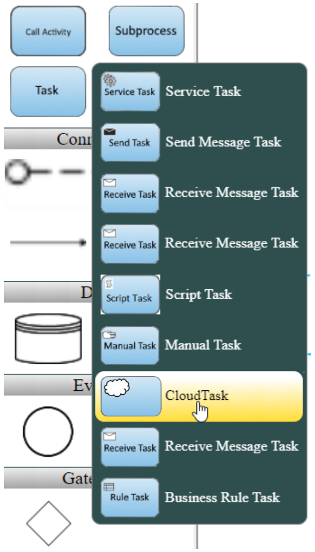
Category: Activities

Palette Image *

Canvas Image *

Cancel

Create New Modeling Element



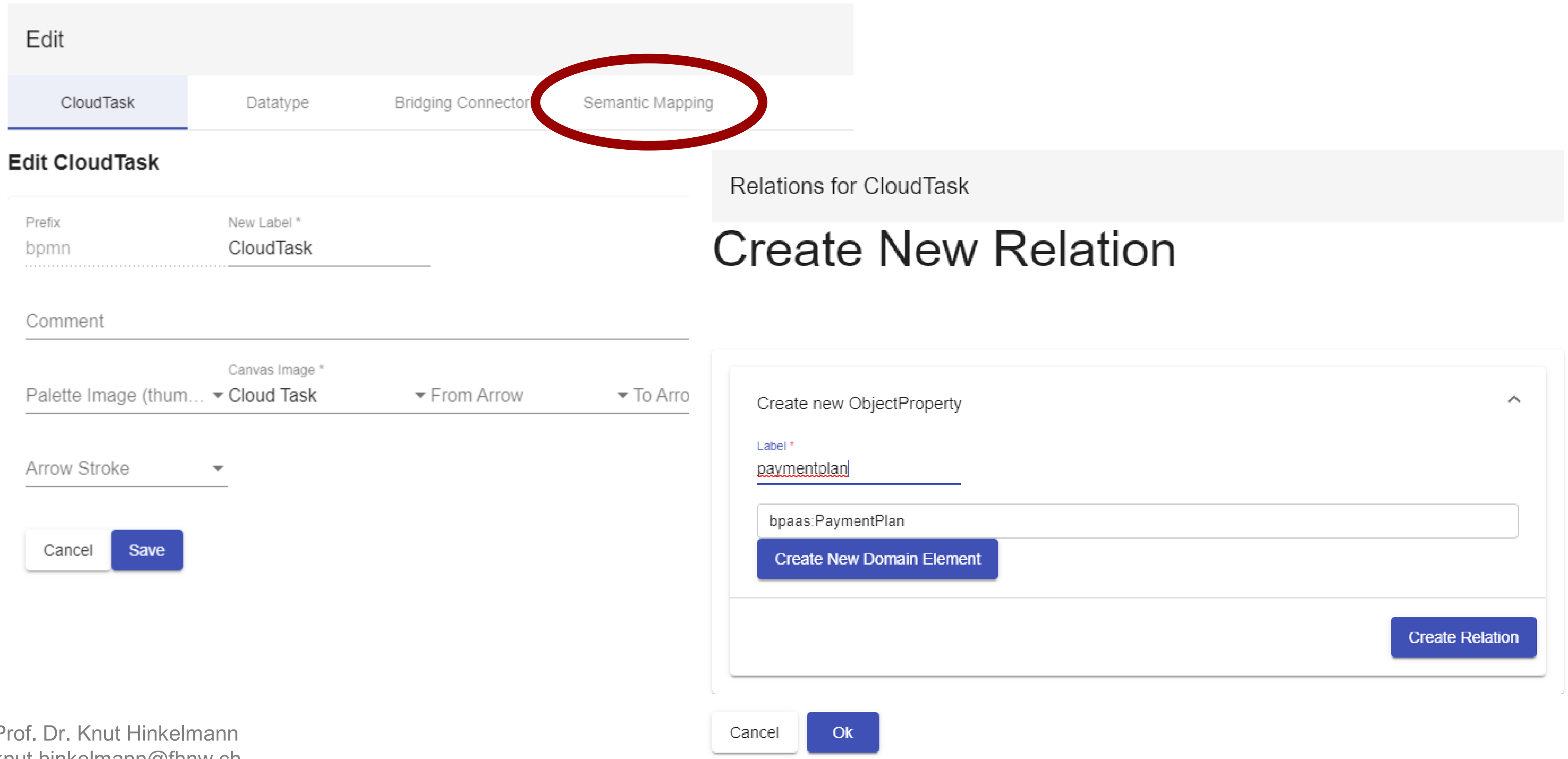
Graphical notation to be shown in palette and canvas

Create New Modeling construct and store in the ontology

Thanks to Emanuele Laurenzi

Semantic Alignment in AOAME

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



Edit

CloudTask | Datatype | Bridging Connector | Semantic Mapping

Edit CloudTask

Prefix: bpmn

New Label *: CloudTask

Comment

Palette Image (thum...

Canvas Image *: Cloud Task

From Arrow

To Arro

Arrow Stroke

Cancel | Save

Relations for CloudTask

Create New Relation

Create new ObjectProperty

Label *: paymentplan

bpaas:PaymentPlan

Create New Domain Element

Create Relation

Cancel | Ok