

# *Ontology-based Enterprise Modeling for Human and Machine Interpretation and Model-driven Architecture*

*Prof. Knut Hinkelmann*

Latest version of slides and material:  
<https://knut.hinkelmann.ch/lectures/nemo2024>



# About Me



# OMILAB<sup>®</sup>



- Research Associate at University of Pretoria



- Visiting Professor at University of Camerino



# Models



not what we  
deal with

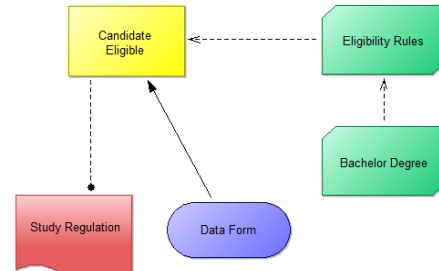
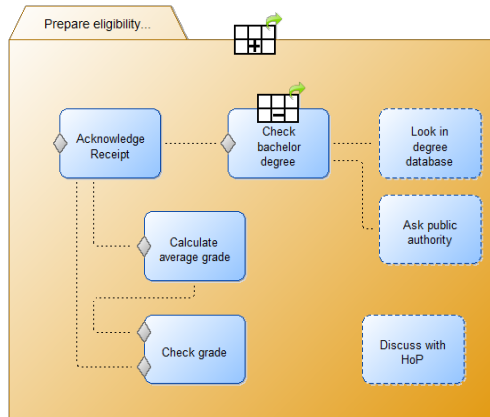
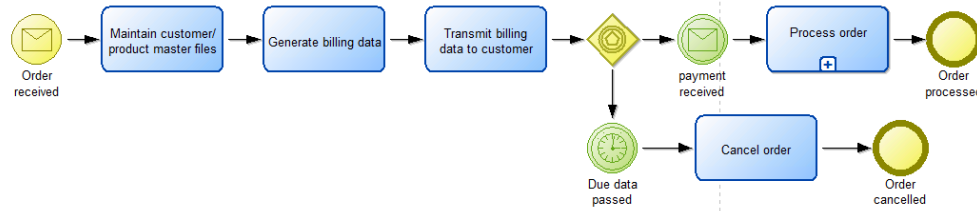


# Enterprise Models

Upps! Still  
not what we  
deal with

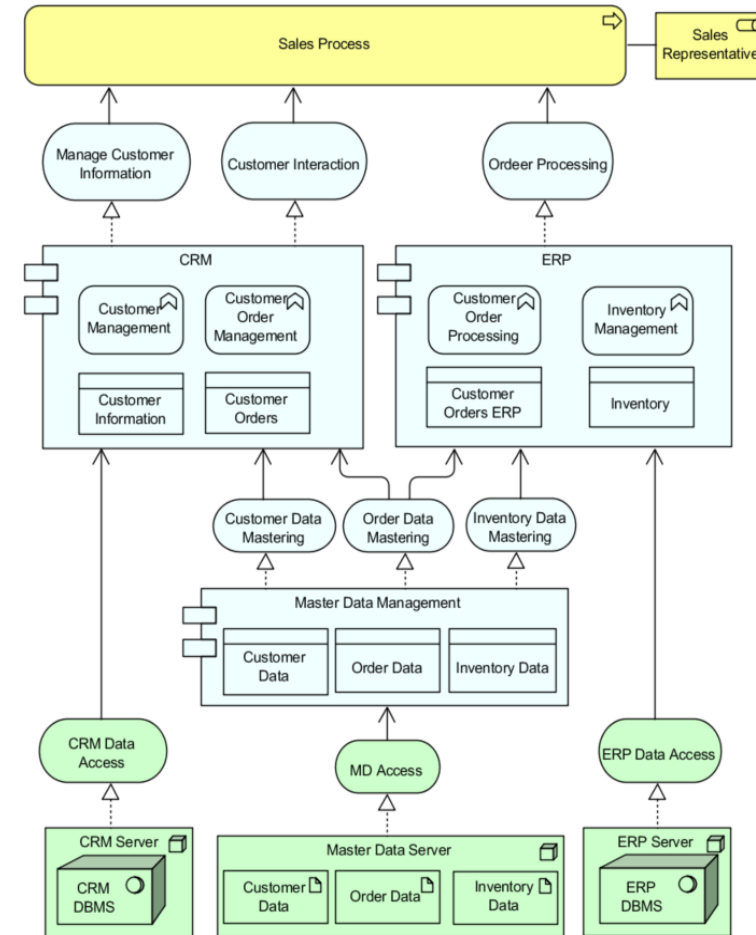
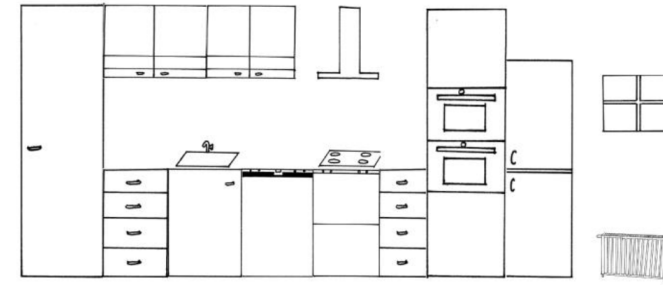
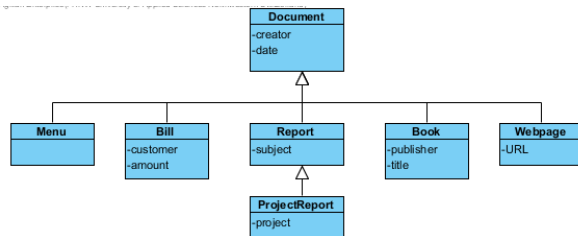


# Enterprise Modeling



Candidate Eligible			
FC	Bachelor Degree	University accredited	eligible
	yes, no	yes, no, unclear	yes, no
1	yes	yes	yes
2	no	no	no
3		no	no
4	yes	unclear	yes

Bachelor Degree			
FC	Bachelor Degree in	Bachelor Degree	
	Information Systems, Business Administration, Information Technology, other, none	yes, no	
1	Information Systems	yes	
2	Business Administration	yes	
3	Information Technology	yes	
4	other	yes	
5	none	no	

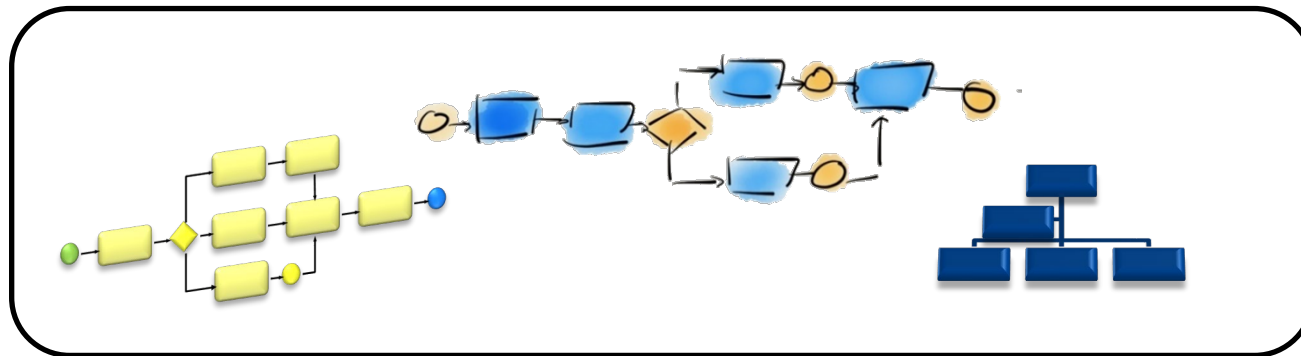


# Modelling: Reproduction of Relevant Part of Reality

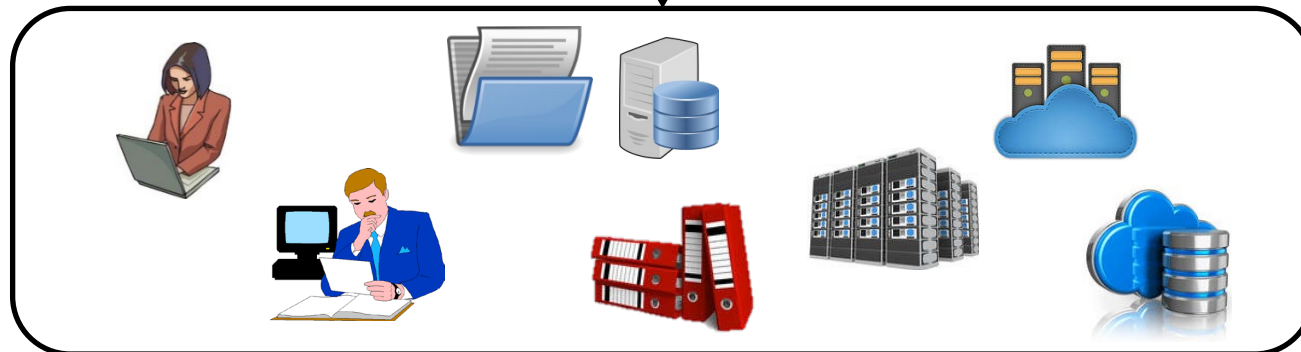
*Communication/  
Analysis/  
Decision Making*



*Models*



*Reality*



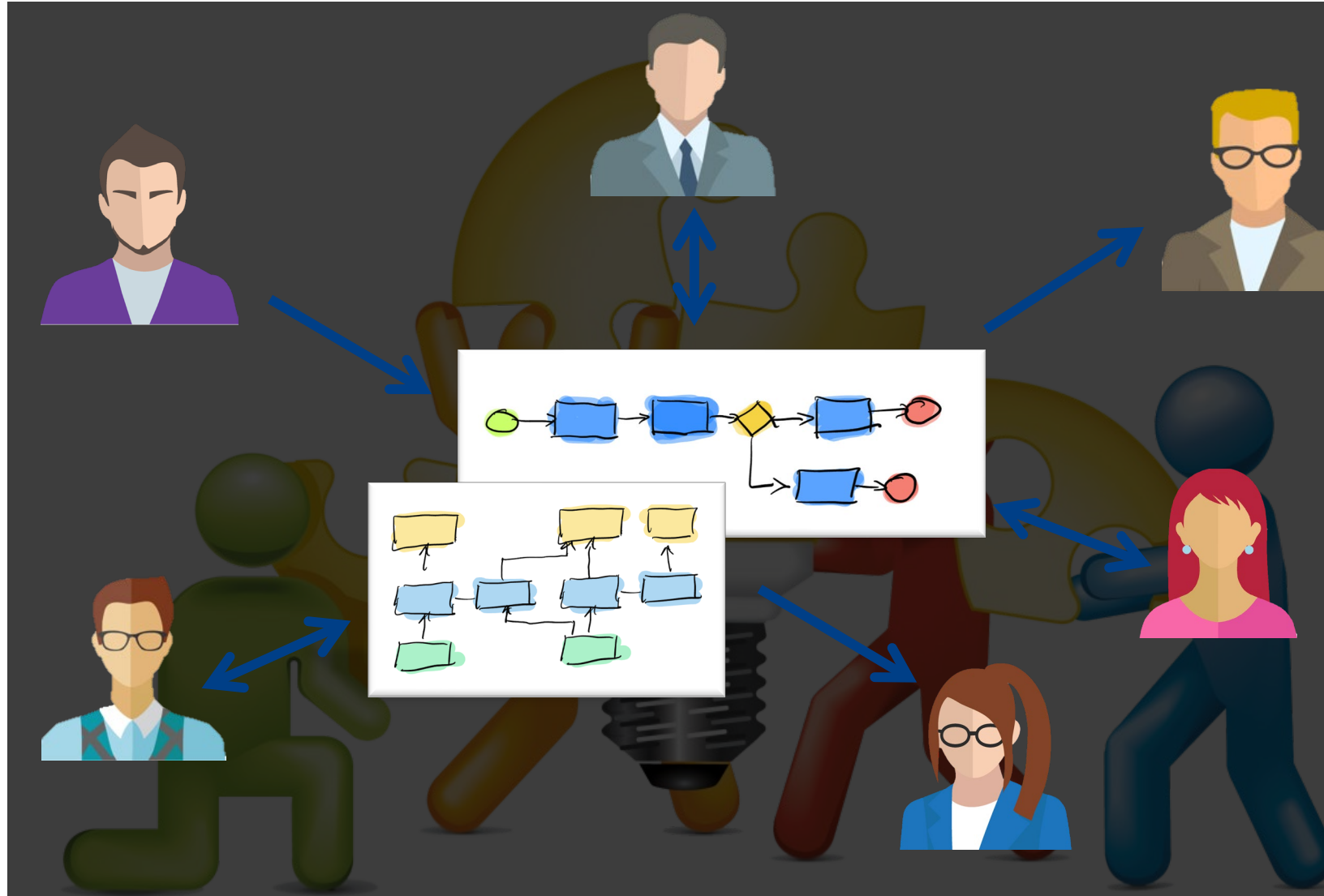
# Collaboration

*Shared understanding of a situation,  
agree on a common goal,  
determine a way to achieve a goal*





# Models in Collaboration: Make Information Explicit



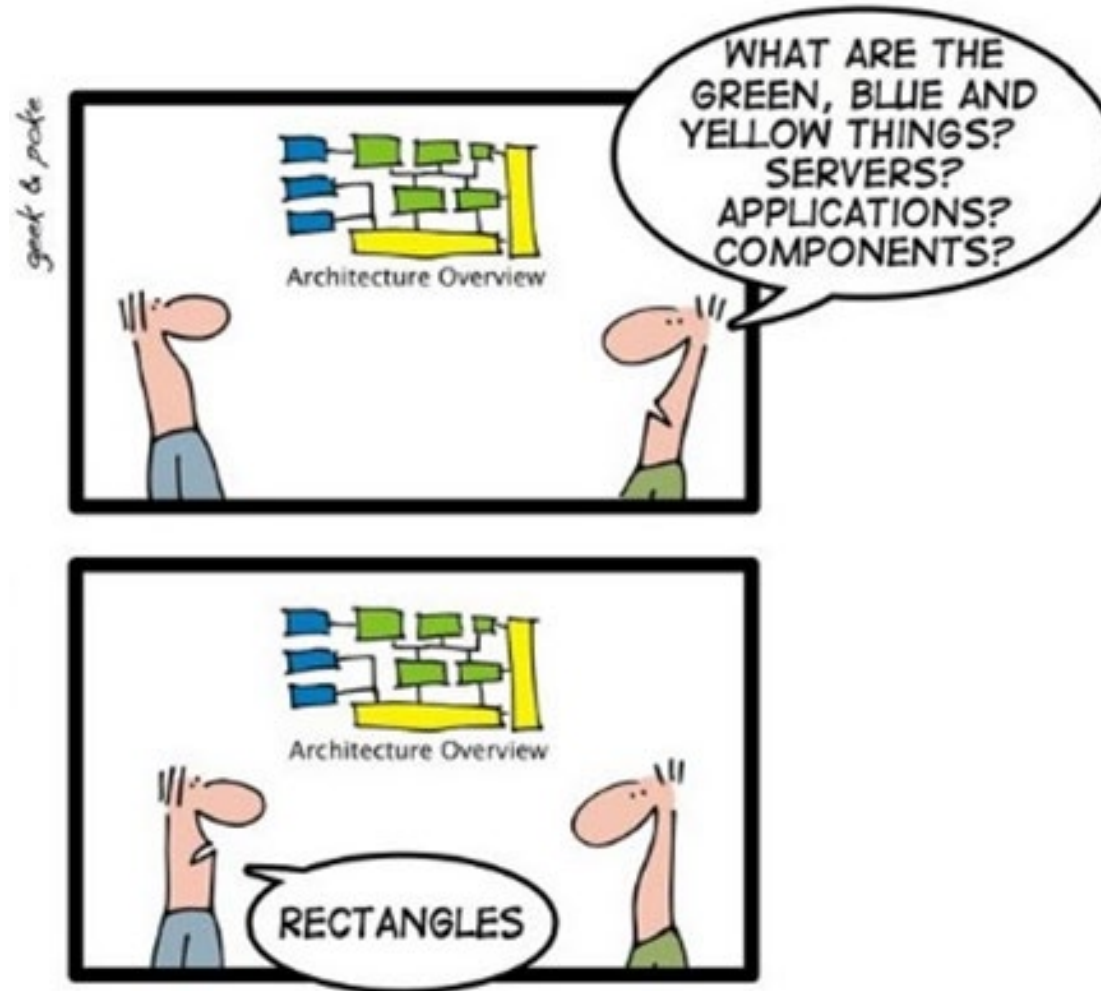
“What is not made explicit is **implicit**,  
which allows everyone to make  
**assumptions**.  
Incorrect assumptions are sources of  
miscommunication and **misunderstanding**”  
John Zachmann

## *Key Terms*

# Models

# Semantics

# *Meaning of Models*

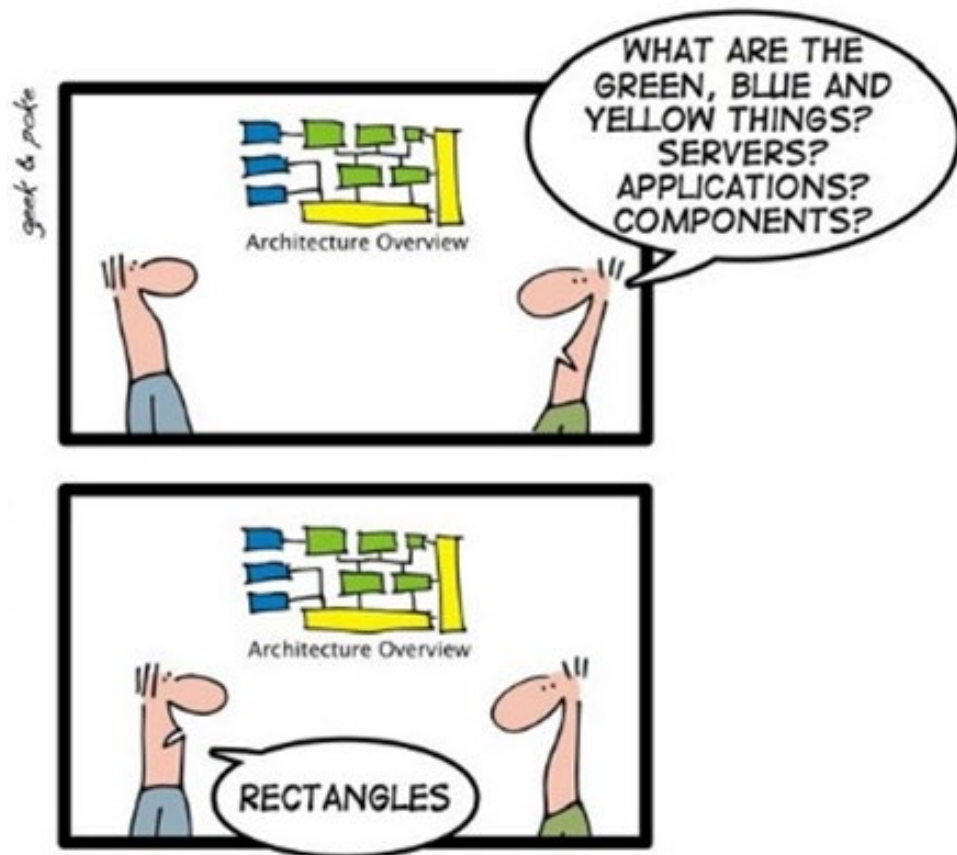


Cartoon adapted by K. Hinkelmann

## Models

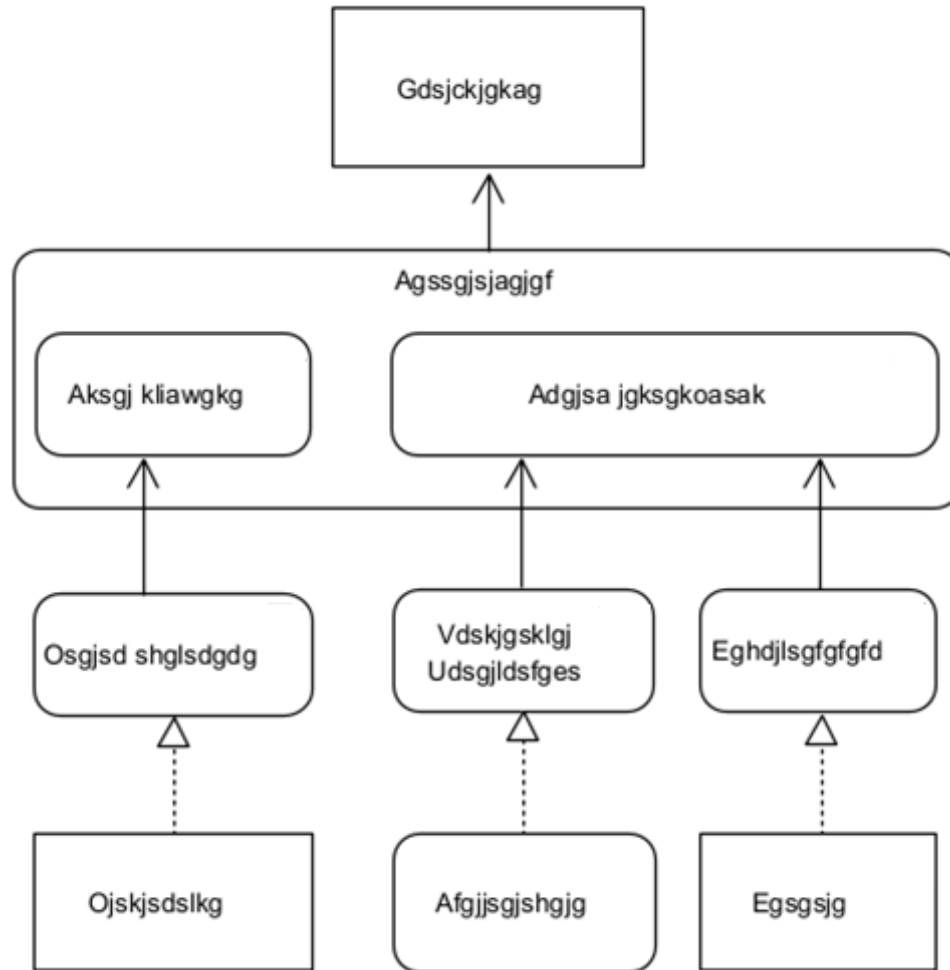
- Models are not mere pictures; rather, they
  - ◆ provide a precise, meaningful description that can be visualized in different ways for different stakeholders;
  - ◆ can be used to
    - develop applications,
    - analyze the impact of changes,
    - evaluate security, risks and compliance
    - calculate cost, risk and other relevant KPIs.

# Interpretation of Models



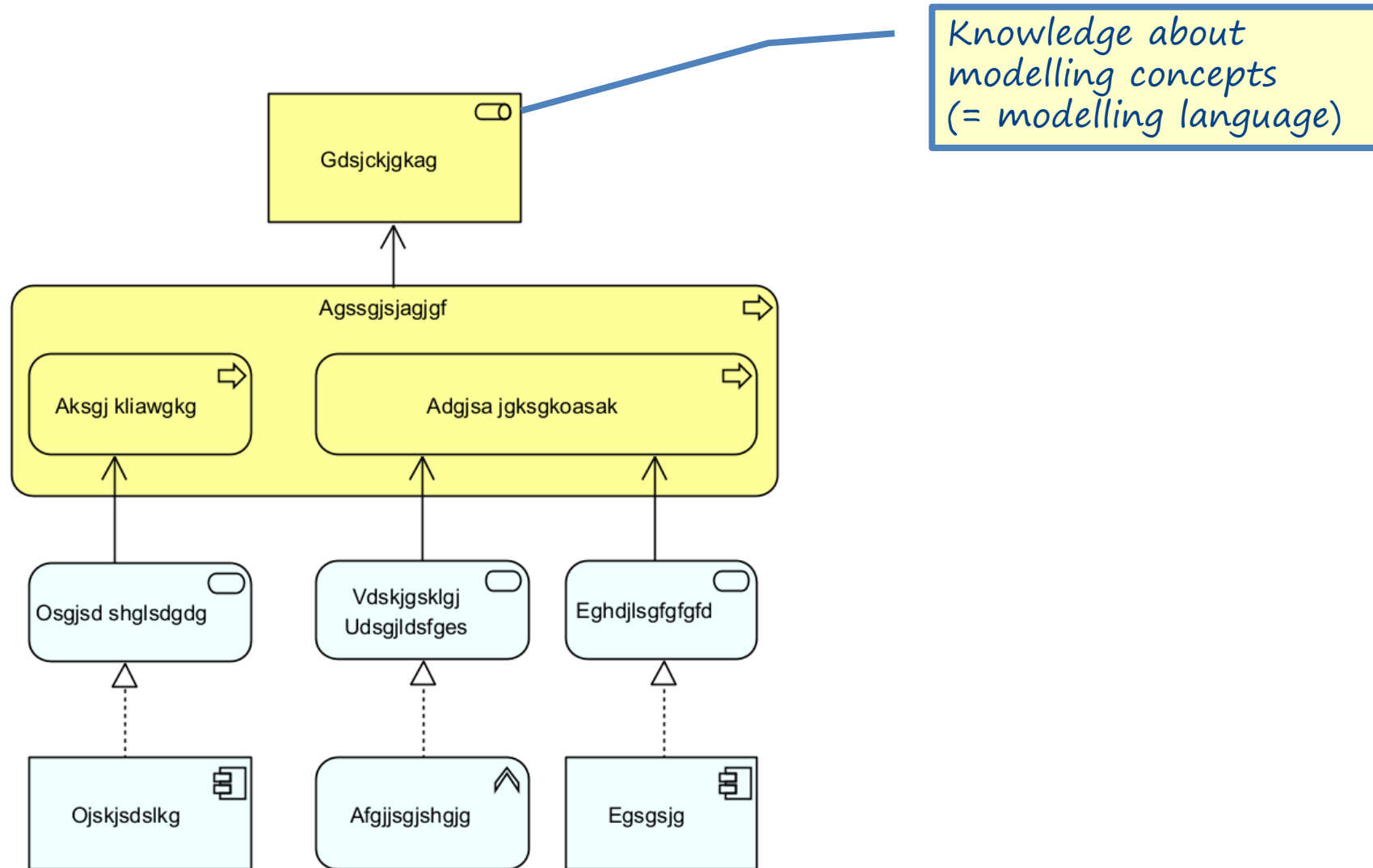
- Models are data
- Models become information by interpretation
- Interpretation requires knowledge

# What Knowledge is needed to interpret this Model?

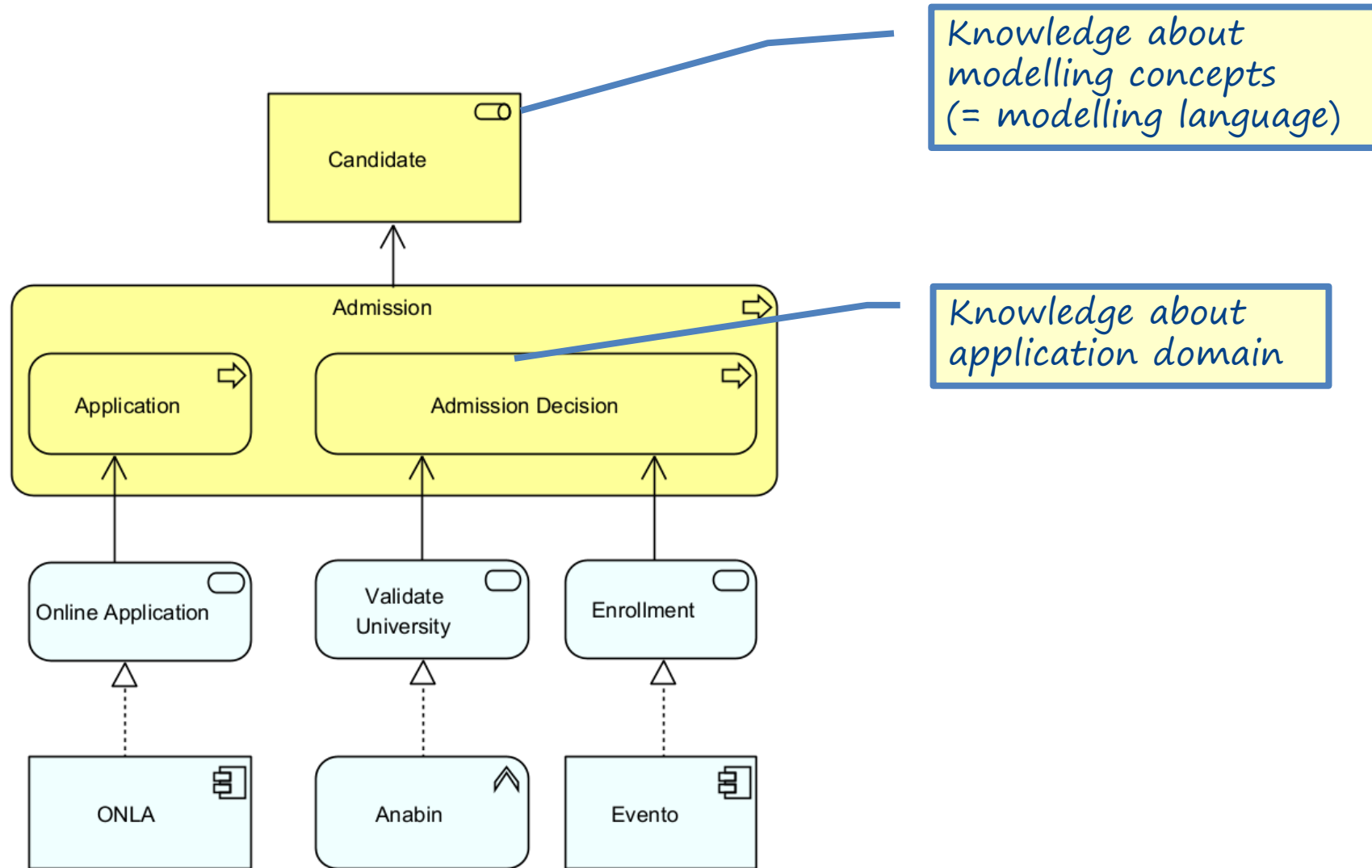




# What Knowledge is needed to interpret this Model?



# What Knowledge is needed to interpret this Model?

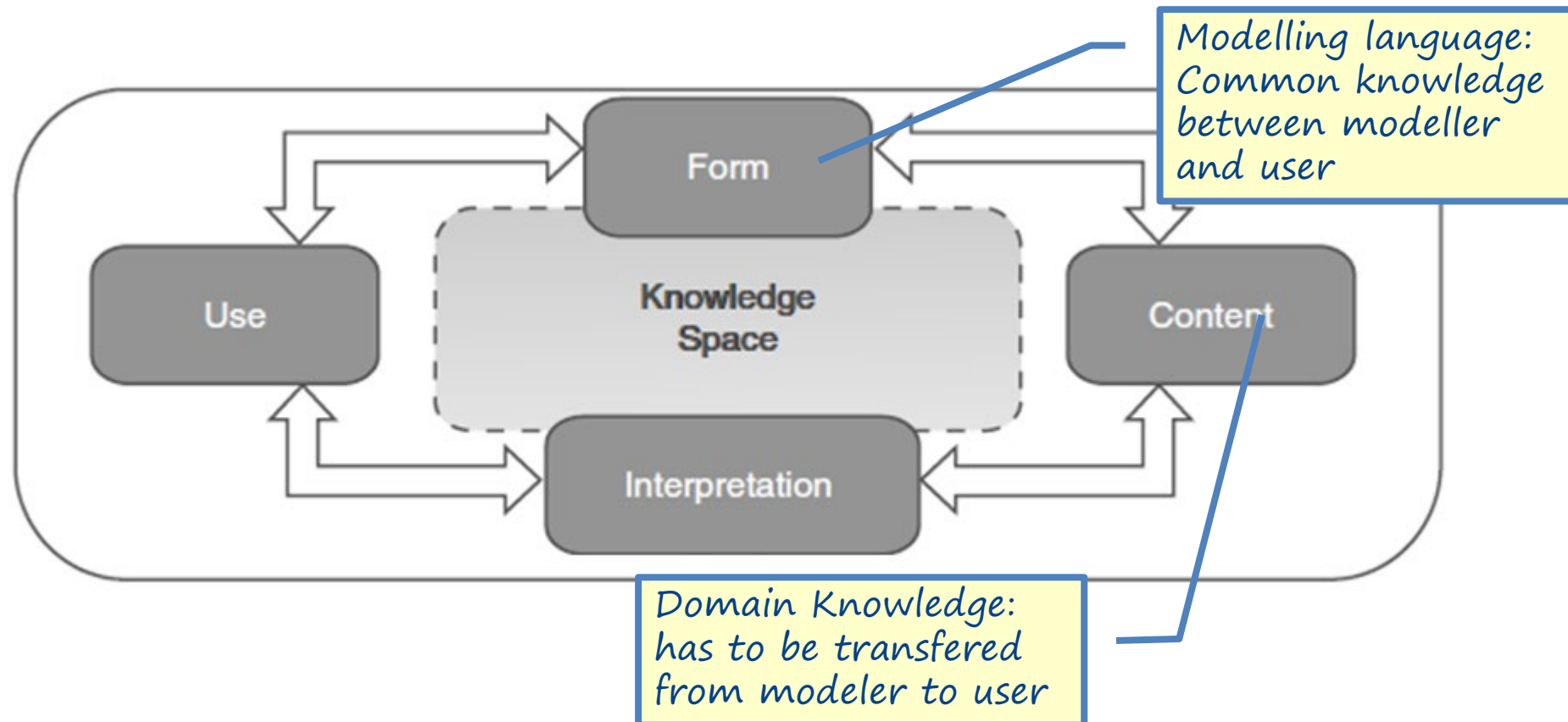


Knowledge about modelling concepts  
(= modelling language)

Knowledge about application domain

## Modelers and Users must have same Interpretation

- Interpretation requires knowledge
- Modelers and users must have the same knowledge



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

# Making Knowledge explicit

## ■ Remember:

*“What is not made explicit is implicit, which allows everyone to make assumptions. Incorrect assumptions are sources of miscommunication and misunderstanding”*

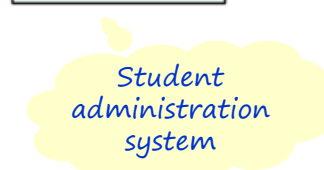
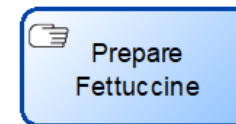
## ■ Make knowledge required for interpretation explicit

### ◆ Knowledge about modeling concepts

- modeling language

### ◆ Knowledge about the domain

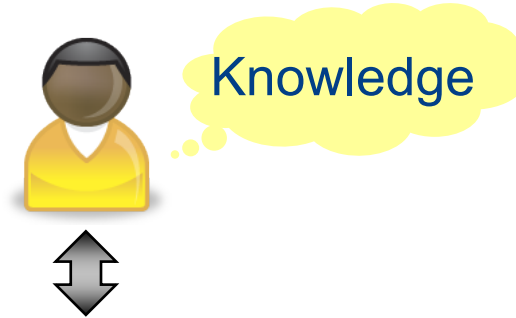
- ???



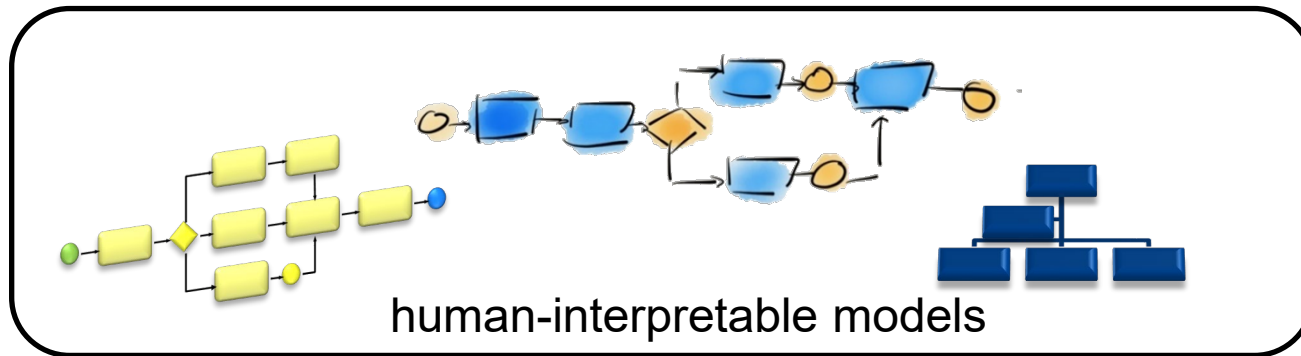
# *Human Interpretation of Models*

# Interpretation of Models requires Knowledge

*Communication/  
Analysis/  
Decision Making*



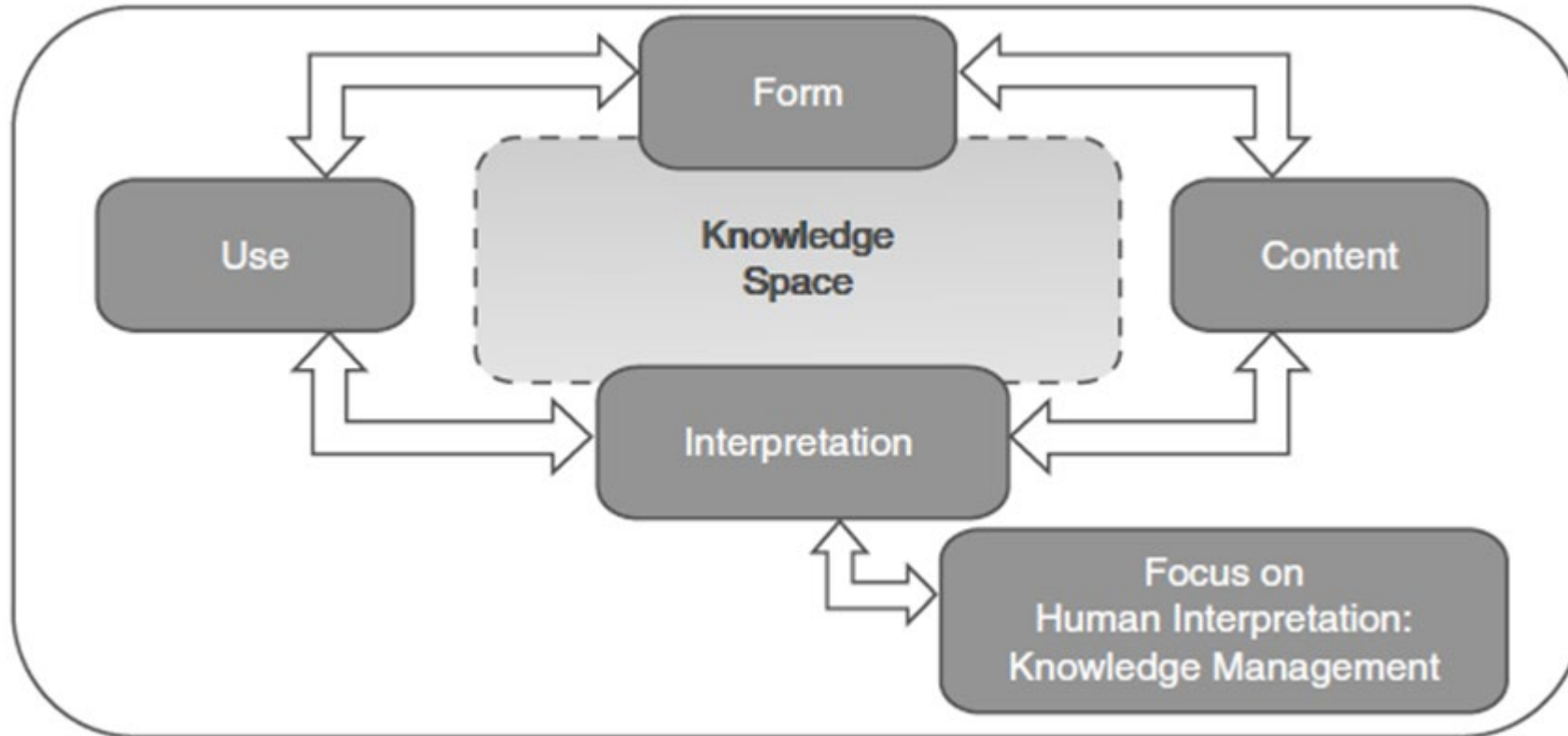
*Models*



*Reality*

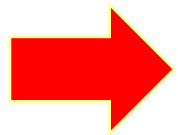


## Human Interpretation requires Knowledge Management



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

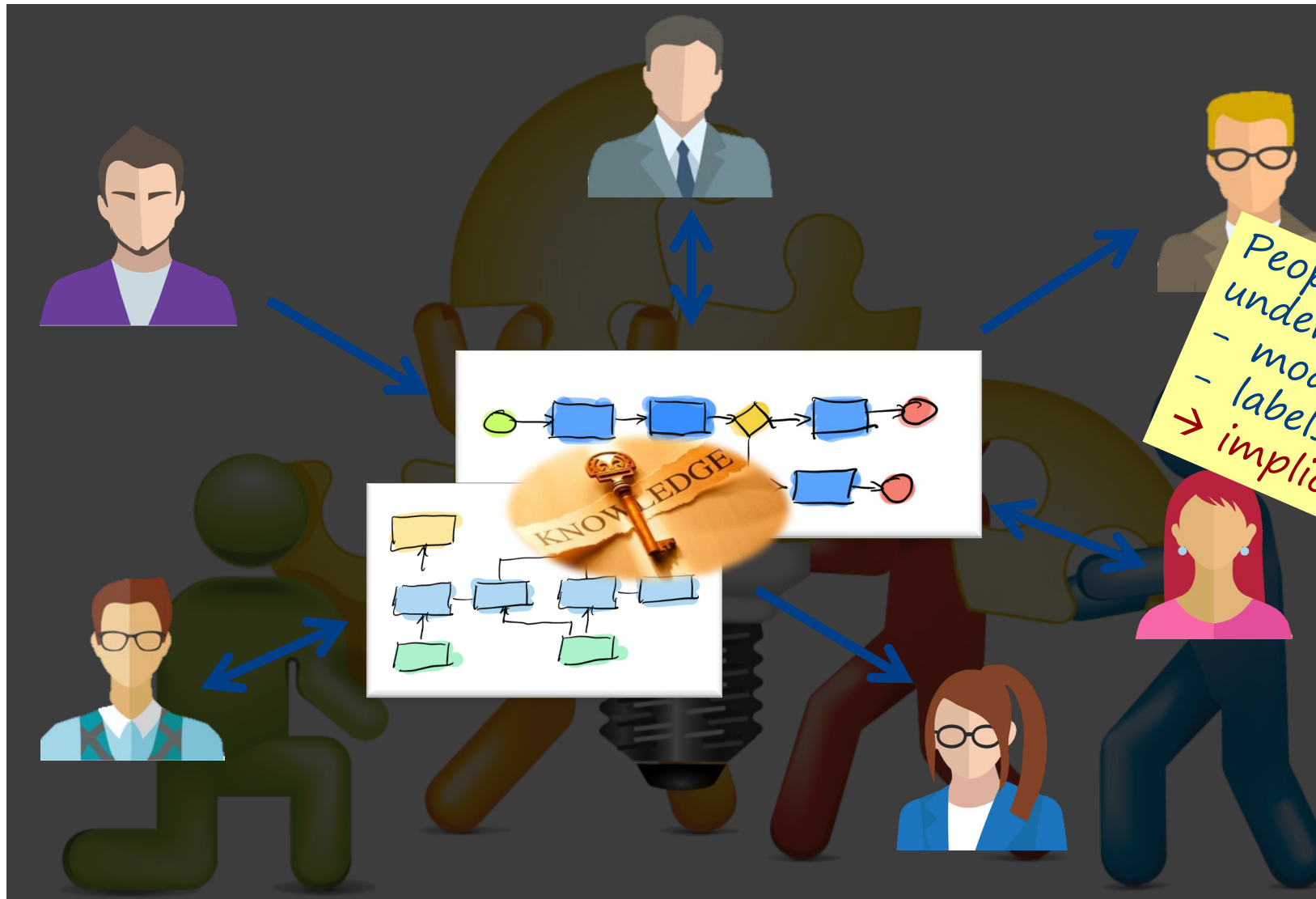
# Modelling



1. Acquiring information about application domain
2. Representing the information in models
3. Managing knowledge required for model interpretation



# Models in Collaboration: Share Information and Knowledge



People need a shared understanding of

- model elements (form)
- labels (content)

→ *implicit shared understanding*

# *Interpreting Models requires up-to-date Knowledge*

© 2000 Randy Glasbergen.  
[www.glasbergen.com](http://www.glasbergen.com)



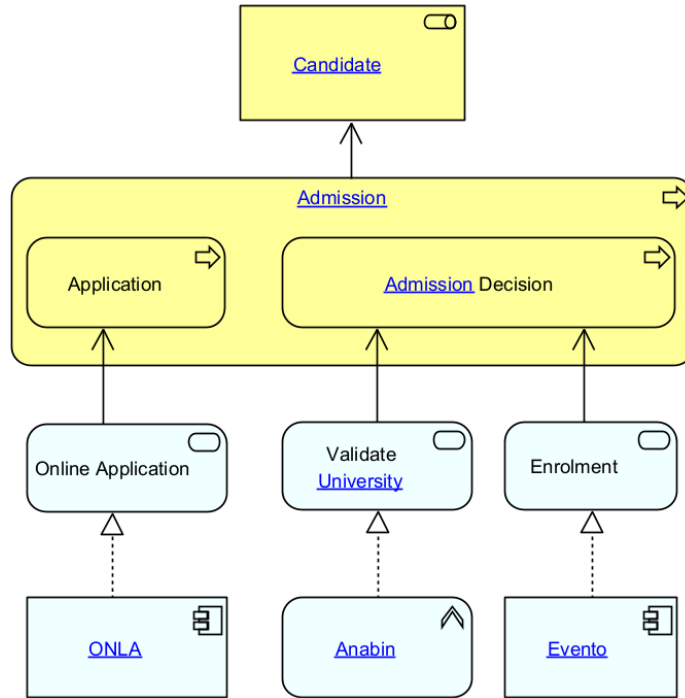
"THE COMPUTER SAYS I NEED TO UPGRADE MY BRAIN  
TO BE COMPATIBLE WITH ITS NEW SOFTWARE."

# Knowledge Management: Sharing Knowledge between Modelers and Model Users

- Make domain knowledge required for interpretation explicit
- Example: Glossary

Name	Aliases	Description
ONLA		The tool for online applications of students developed by the FHNW
Anabin		A web application provided the the German Kultusministerkonferenz, which contains information about recognition of universities
Candidate	Prospective student	A person who applied for a study program
Student		A person who is enrolled into a study program at a university
University		An education and research institution on tertiary level.
Admission		The process of deciding, wether a candidate is eligible for a study program.
Eligibility	acceptable	A candidate is eligible, if he/she satisfies all the eligibility criteria
Evento		The student administration system, which contains all information about student, staff and study programs.

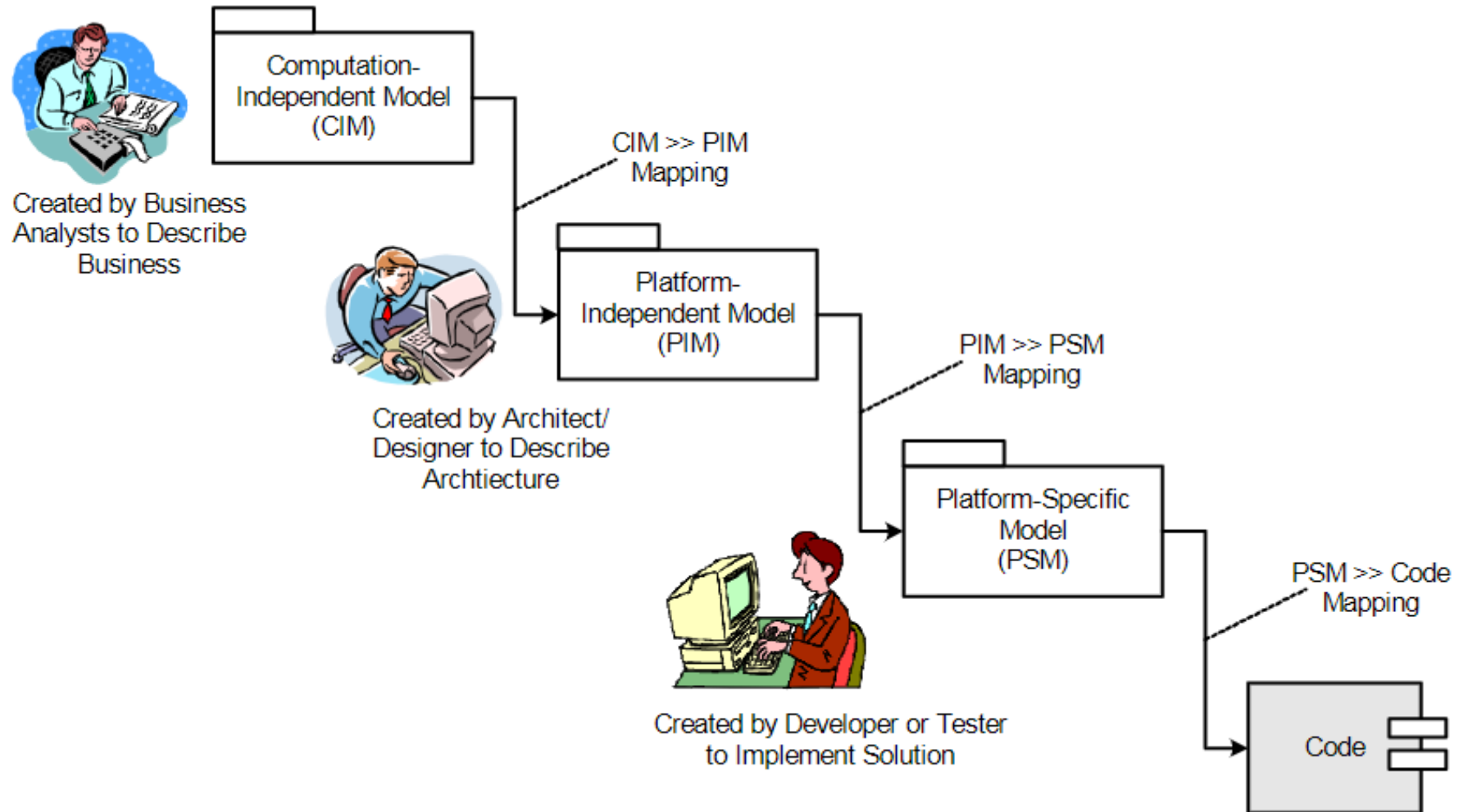
# Linking Models to Domain Knowledge



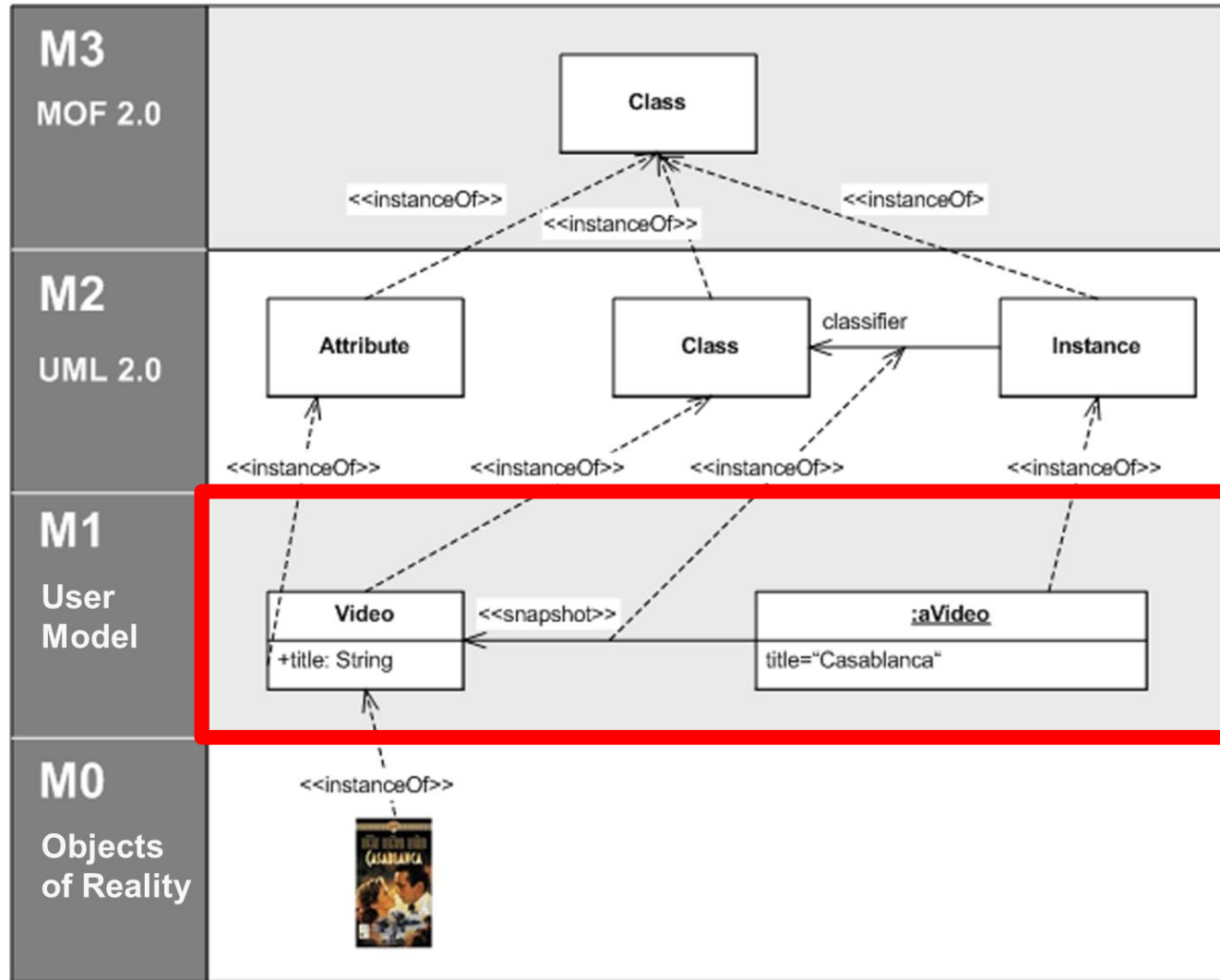
Name	Aliases	Description
ONLA		The tool for online applications of students developed by the FHNW
Anabin		A web application provided the the German Kultusministerkonferenz, which contains information about recognition of universities
Candidate	Prospective student	A person who applied for a study program
Student		A person who is enrolled into a study program at a university
University		An education and research institution on tertiary level.
Admission		The process of deciding, wether a candidate is eligible for a study program.
Eligibility	acceptable	A candidate is eligible, if he/she satisfies all the eligiblity criteria
Evento		The student administration system, which contains all information about student, staff and study programs.

# *Model-Driven Architecture*

# Model-Driven Architecture



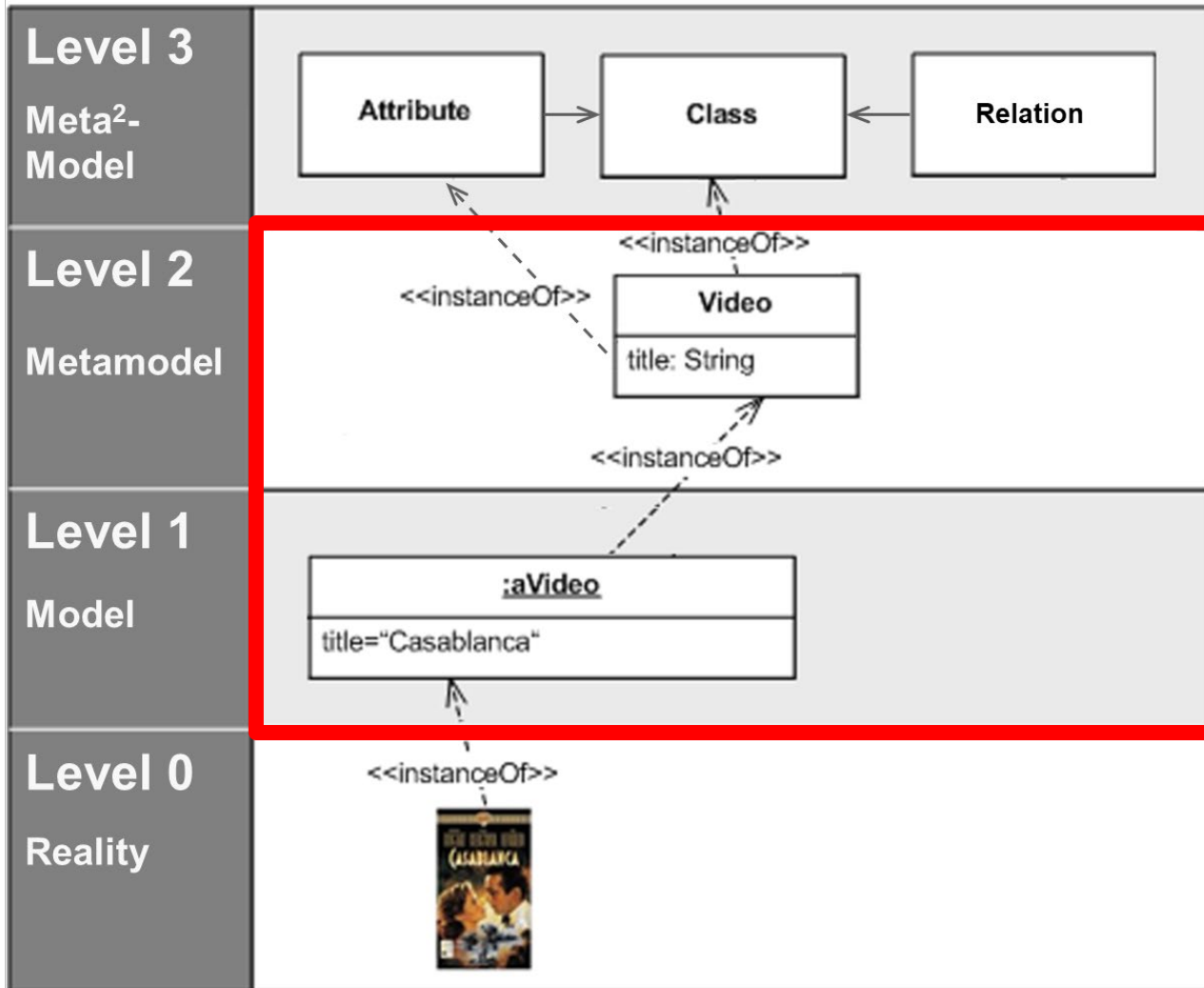
# Meta-Object Facility (MOF)



- M1 contains models *and* conceptualization
- Lack of shared understanding of the models
- Knowledge needed for model interpretation not explicit
  - ◆ high probability of misunderstanding (by humans)
  - ◆ limitations for automation (for transformation)

Original graphic from <https://commons.wikimedia.org/wiki/File:M0-m3.png>

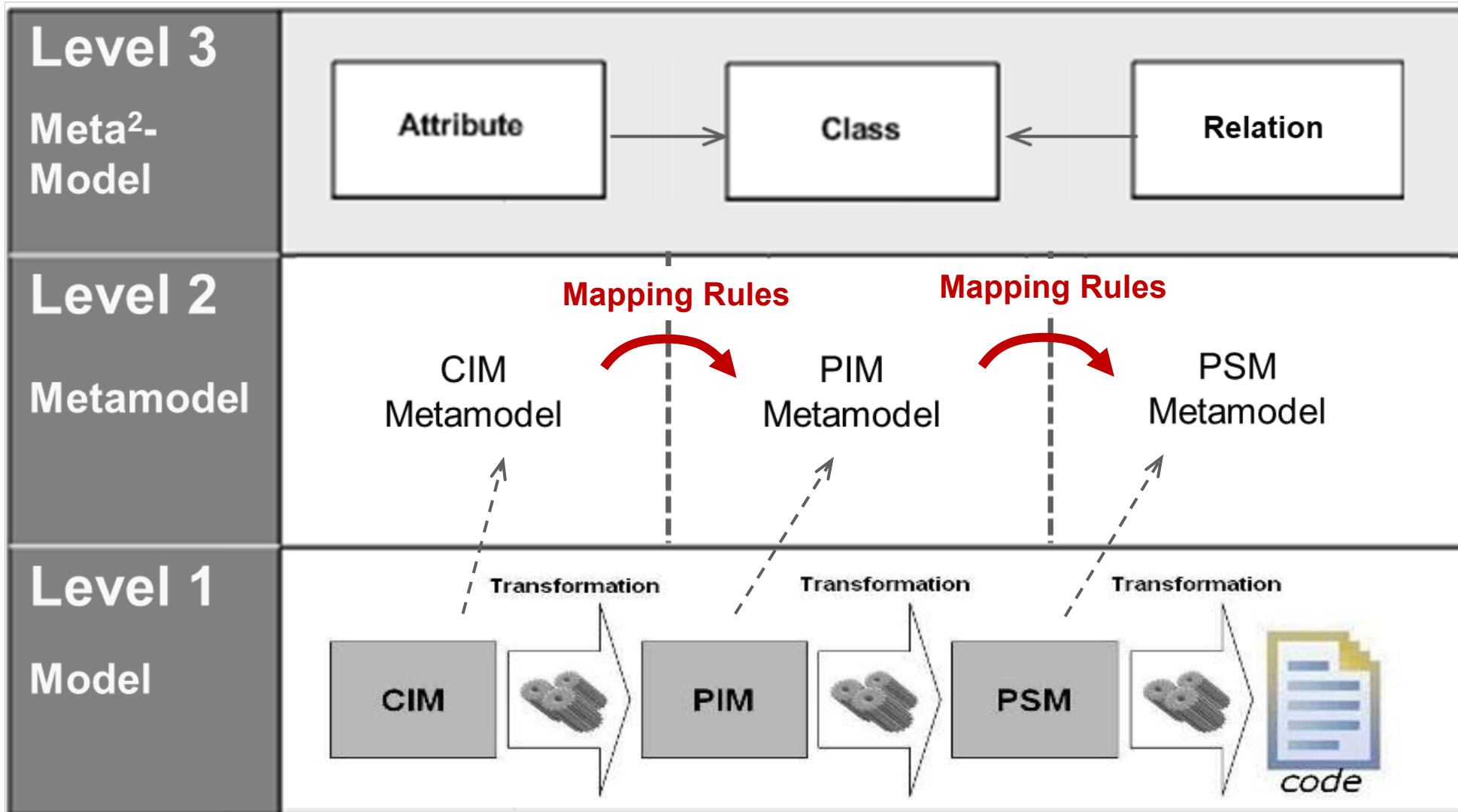
# Proposal: Apply Conceptual Modelling



- Separation of conceptualization and modelling
  - ◆ move metamodel one level higher
  - ◆ metamodel layers (Karagiannis & Kühn 2002)
- Metamodels can be agreed upon and reused, e.g.
  - ◆ process models
  - ◆ architecture
- Transformation rules can use conceptualization

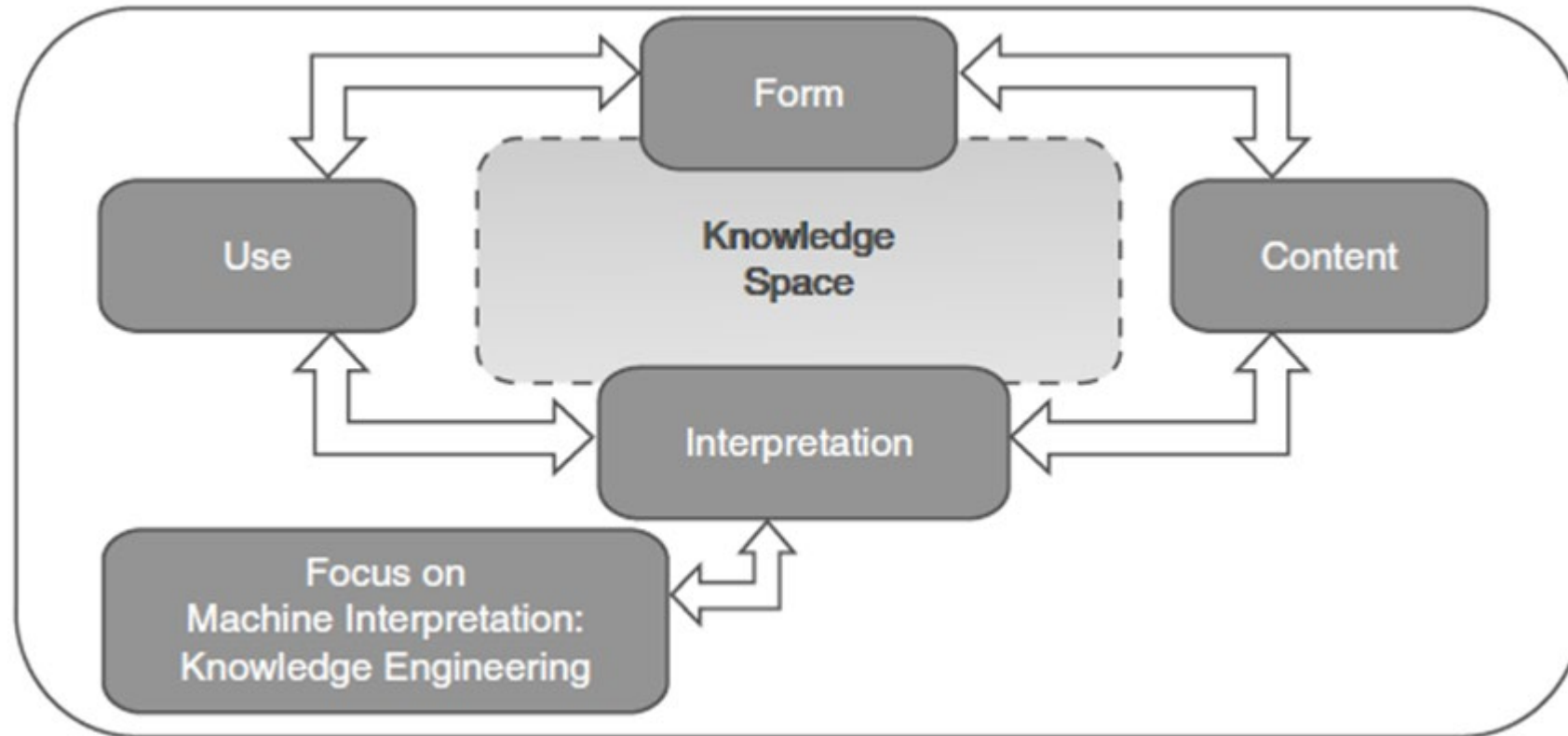


# Metamodels for Model-Driven Architecture



# *Machine-Interpretation of Knowledge*

# 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.

## *Objective*

Models should allow automated analysis, decision making and digitalization.

- Examples:
  - ◆ Translate from CIM to PIM to PSM
  - ◆ Validate architecture principles
  - ◆ Select appropriate cloud services
  - ◆ Plausibility checks

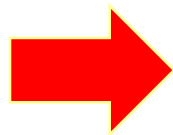
## *Objective*

Models should allow automated analysis, decision making and digitalization.

## *Approach*

Represent the models ***and knowledge*** so that it can be interpreted by a system for decision making and problem solving

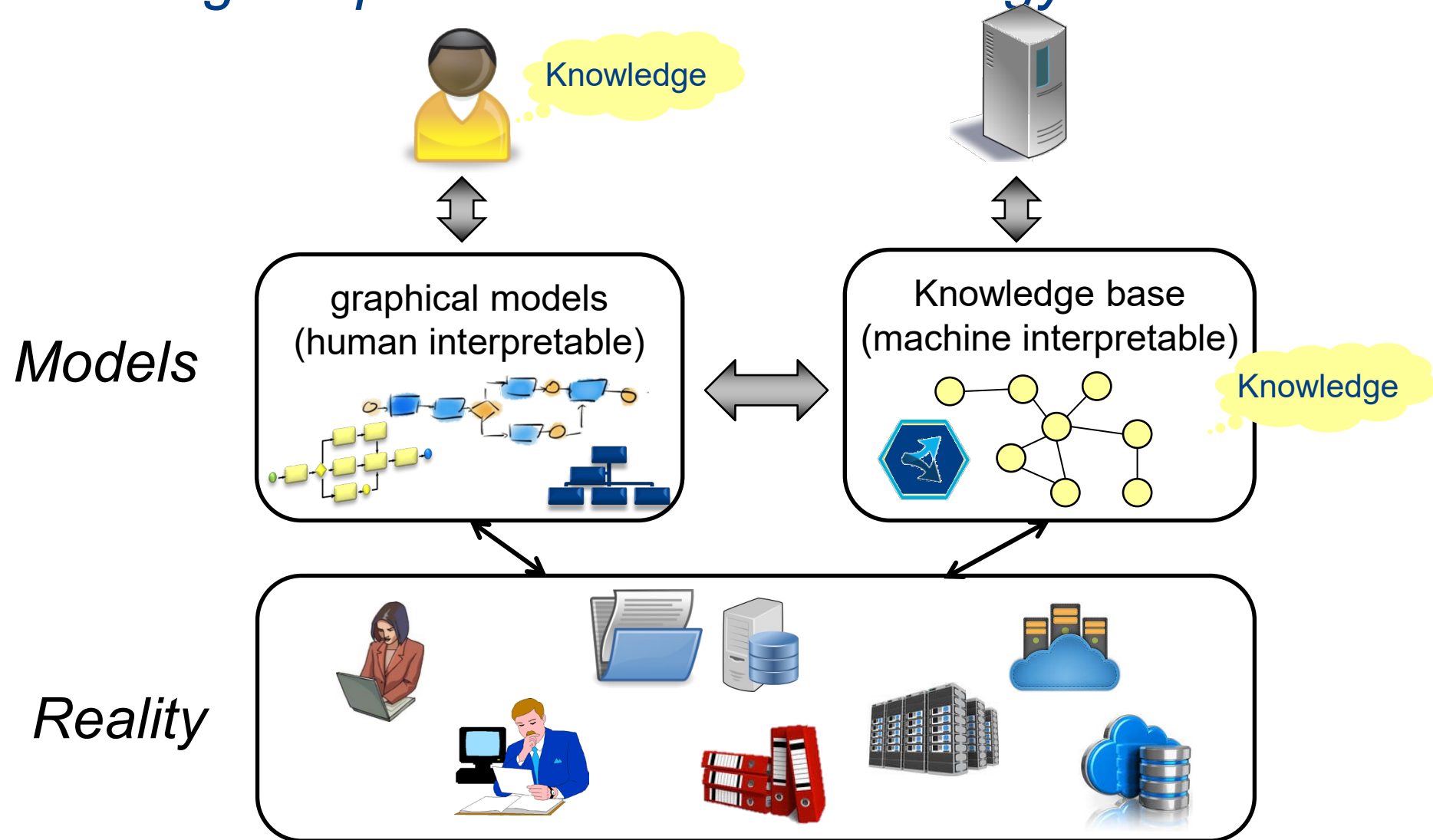
# Modelling



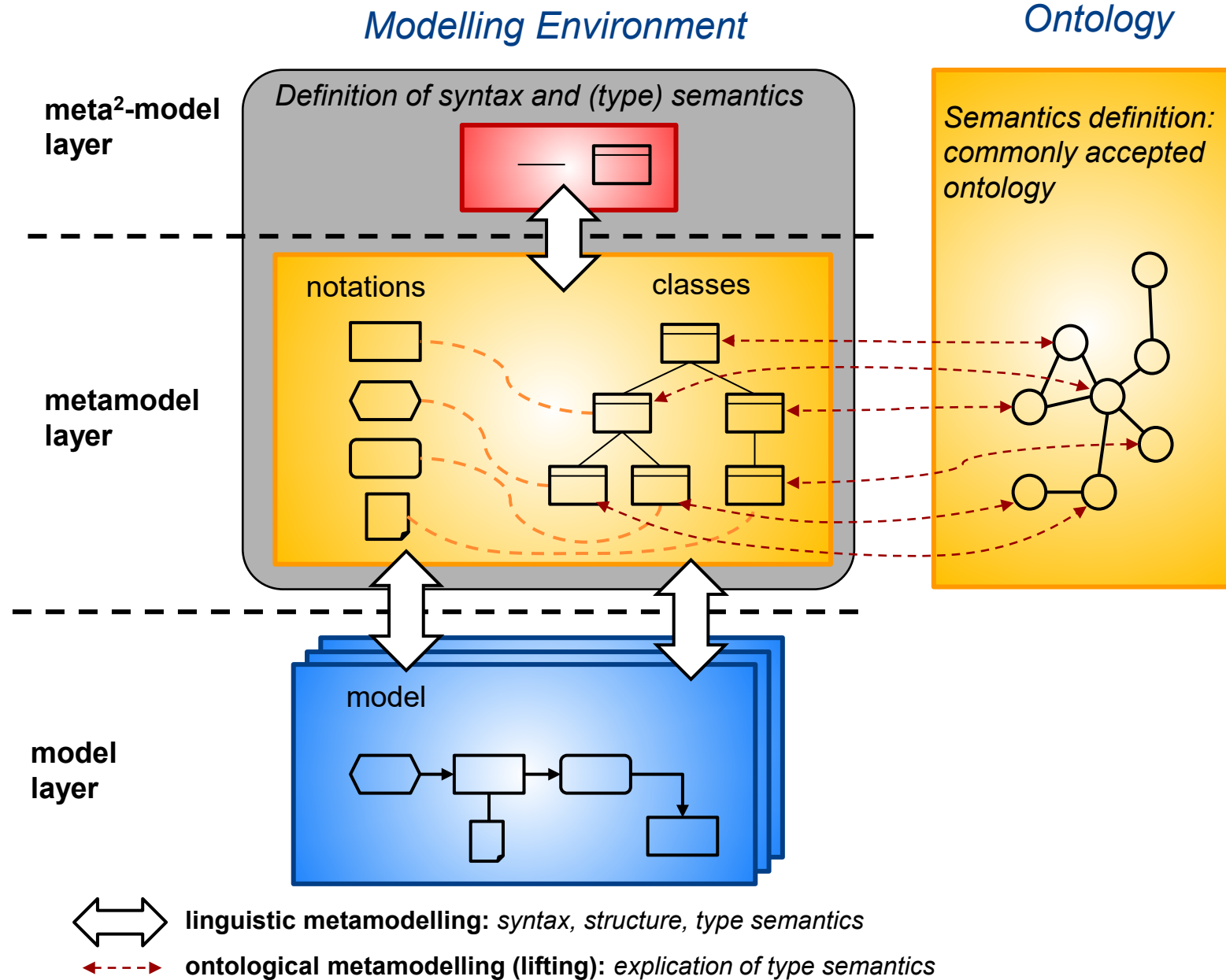
1. Acquiring information about application domain
2. Representing the information in models
3. Representing knowledge required for model interpretation

# *Semantic Lifting*

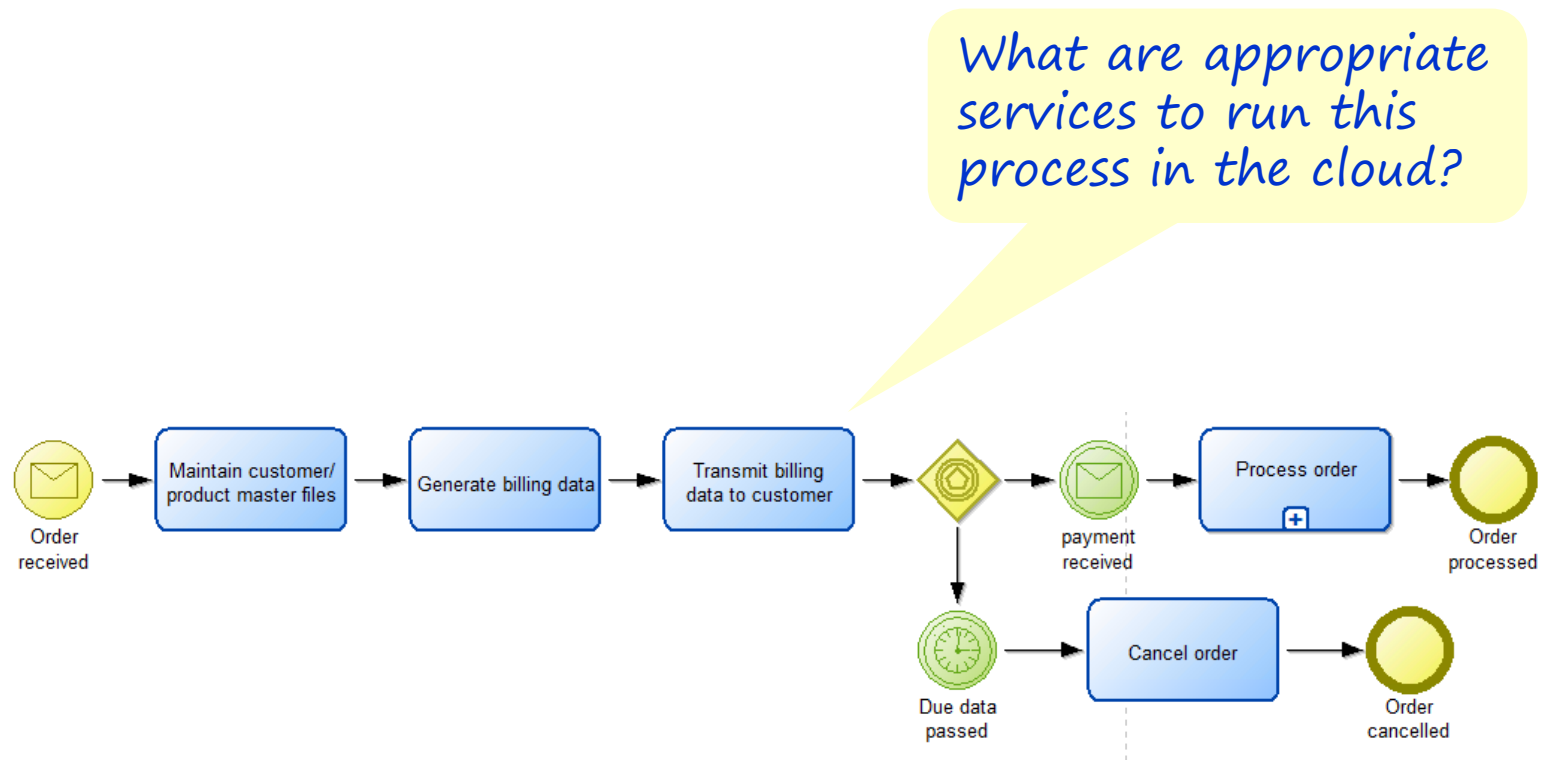
# Semantic Lifting: Map Models into an Ontology







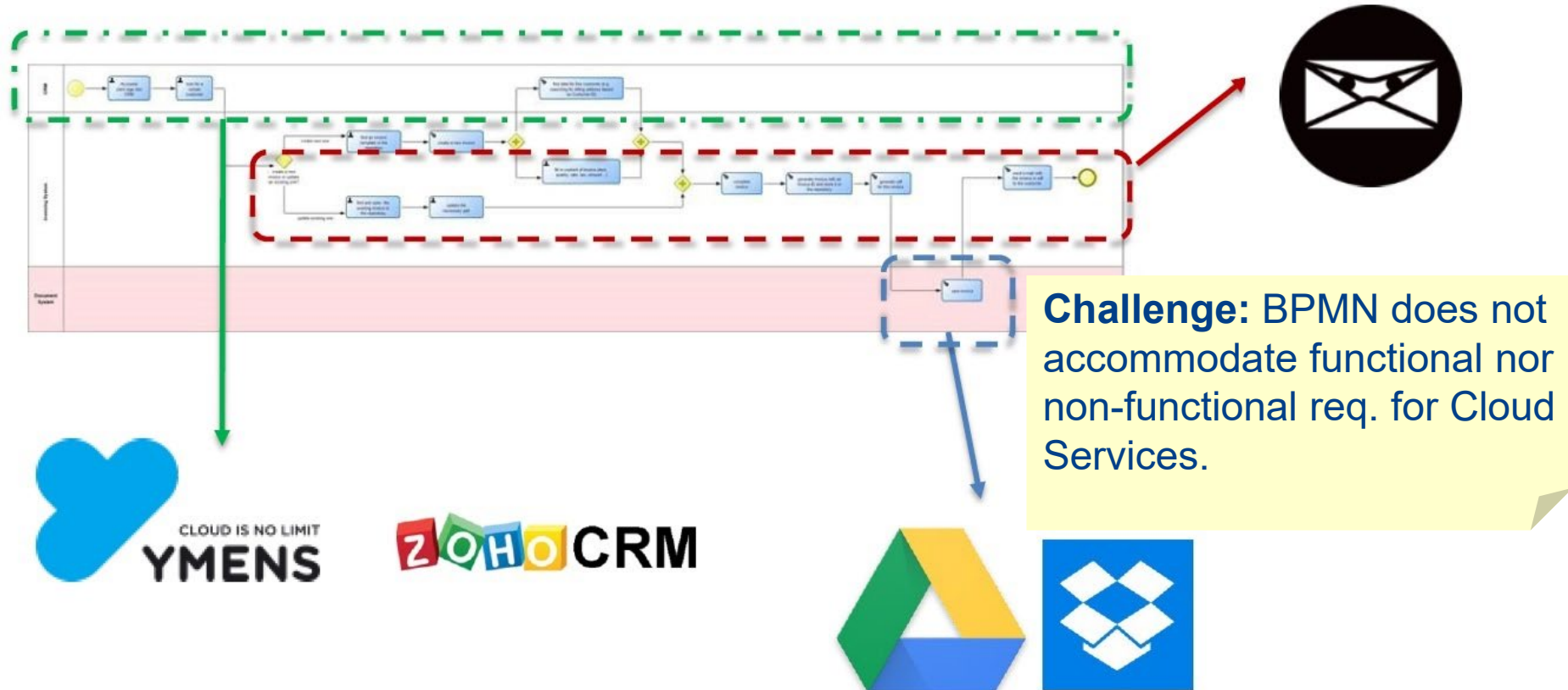
## Example: Business Process as a Service (BPaaS)



From: CoudSocket Project

# BPaaS: From Platform-independent to Platform-specific model

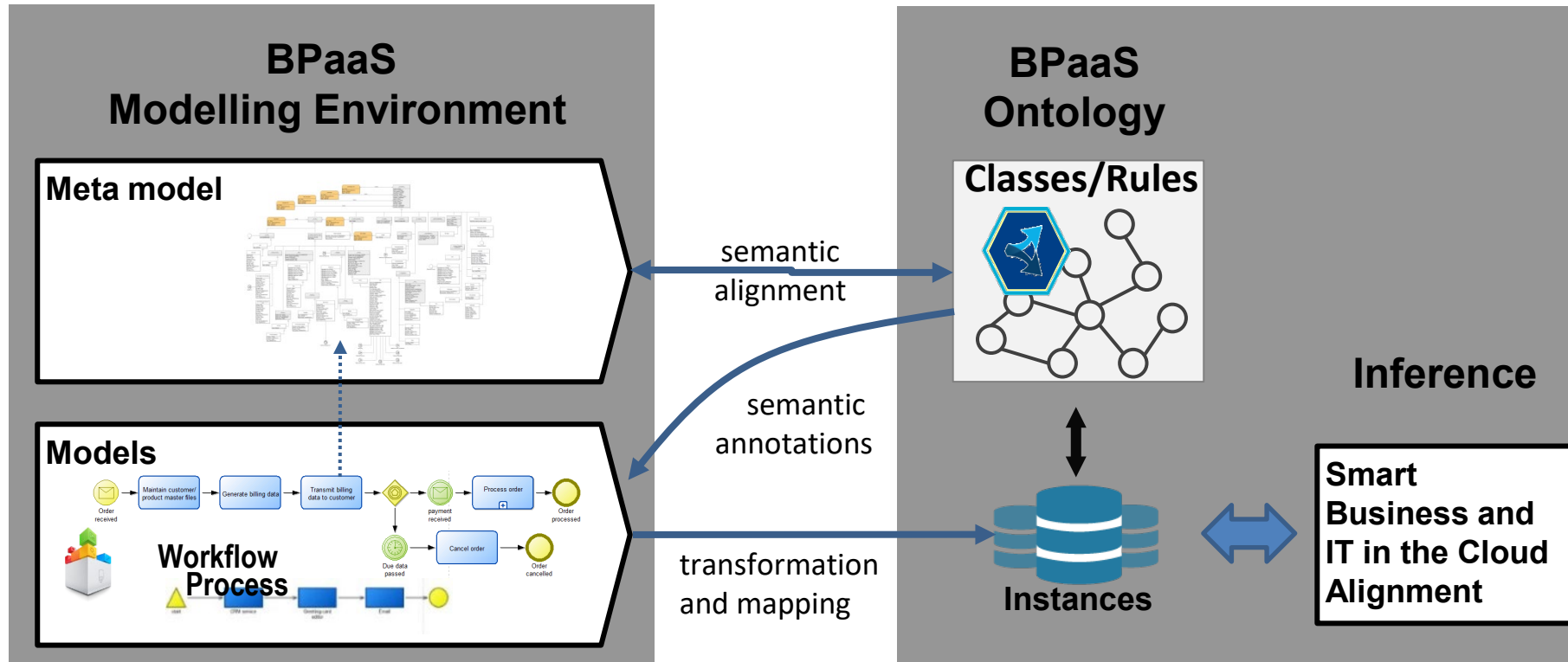
Given a BPMN business process, retrieve all the Cloud Services that satisfy the functional and non-functional requirements.



# Semantic Lifting for BPaaS

**human interpretation**  
informal and semi-formal

**machine interpretation**  
formal



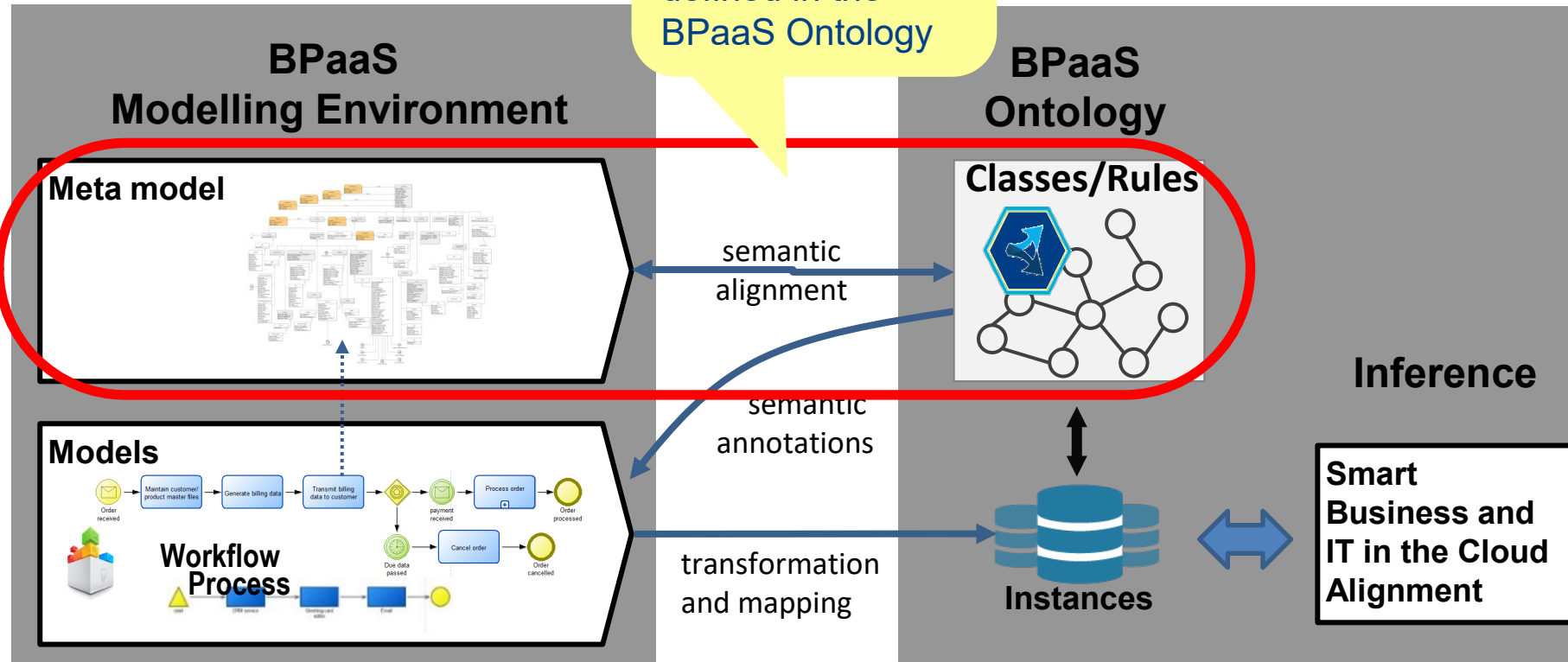
From: CoudSocket Project

# Semantic Alignment

**human interpretation**  
informal and semi-formal

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

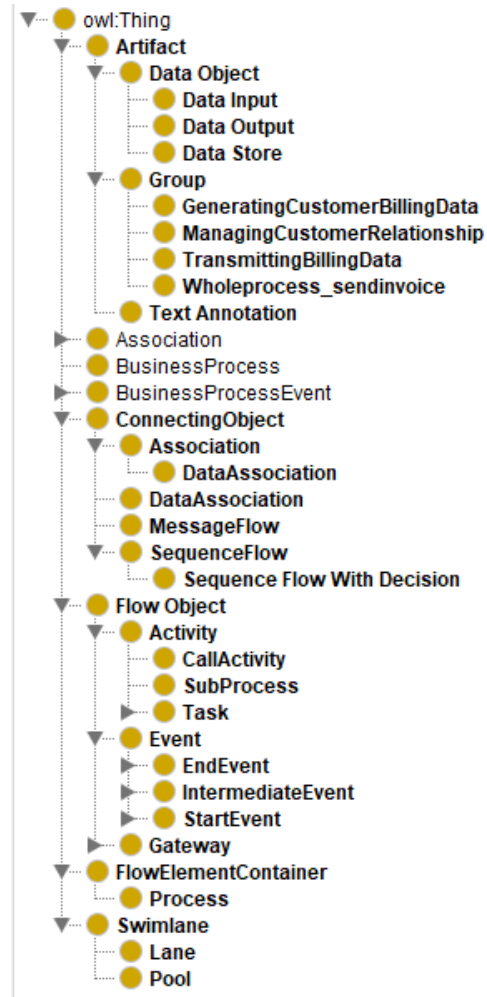
**machine interpretation**  
formal



From: CoudSocket Project

# Ontology with BPMN Modeling Concept

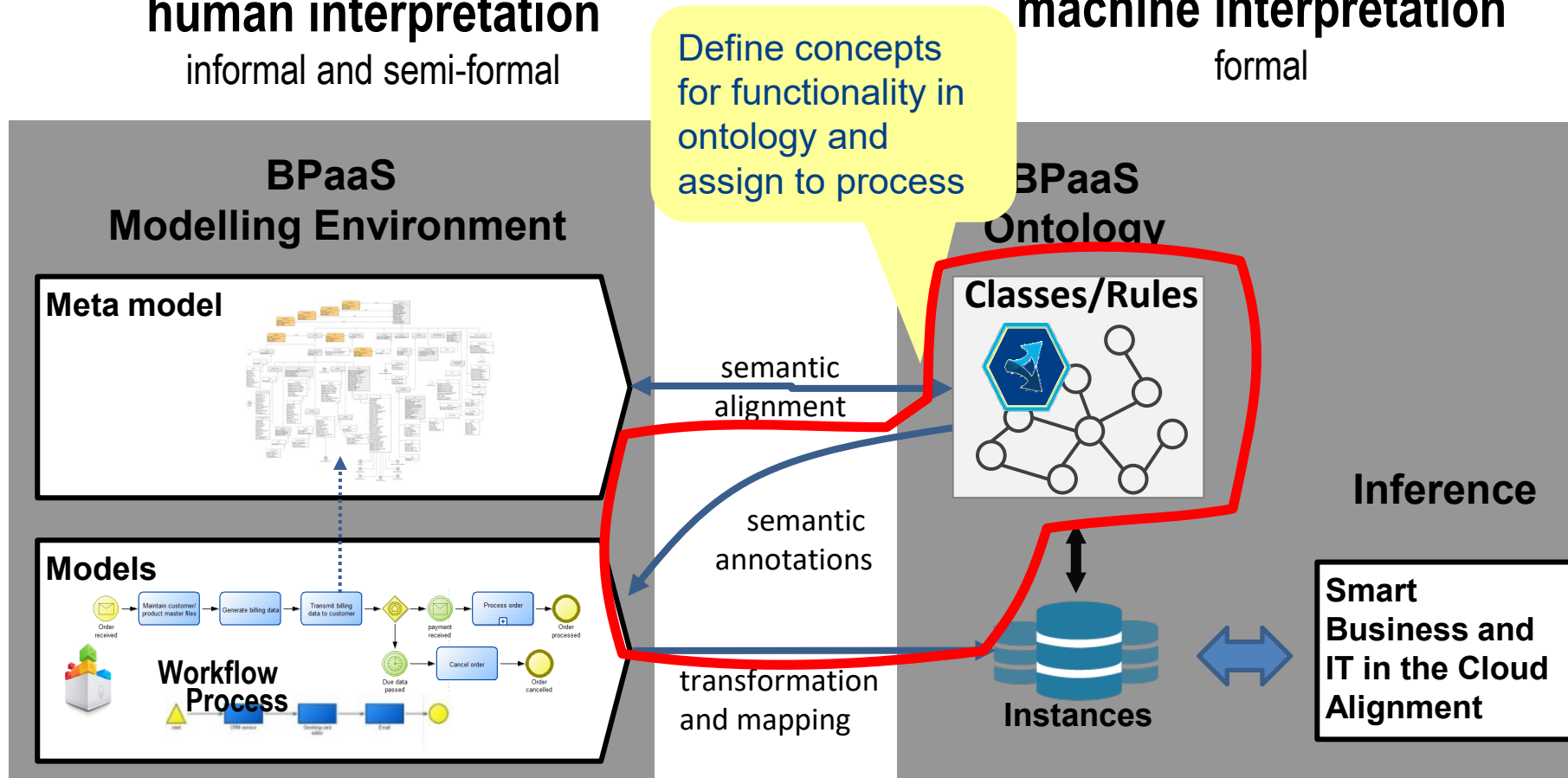
## BPMN Ontology:



# Semantic Annotation

**human interpretation**  
informal and semi-formal

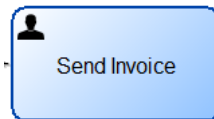
**machine interpretation**  
formal



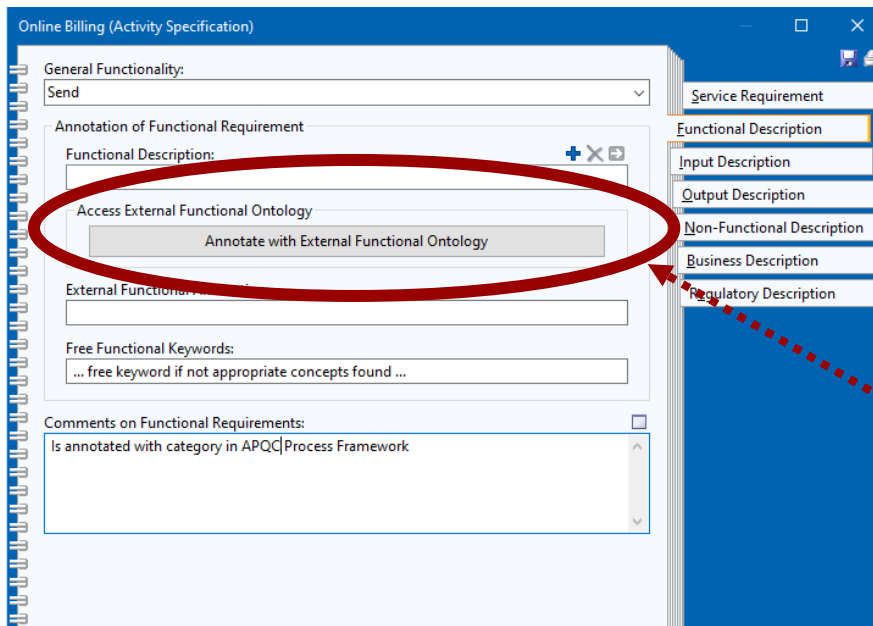
From: CoudSocket Project

# Application Knowledge: Semantic Annotations

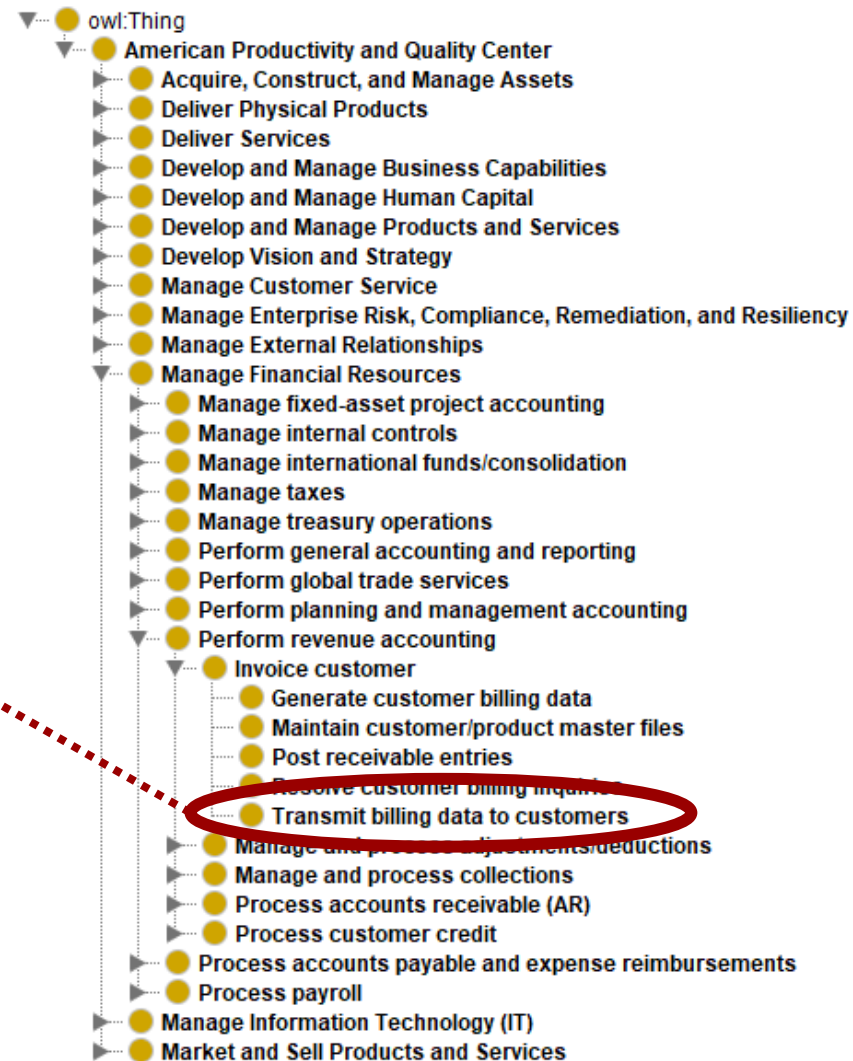
Annotate modeling elements with classes from the domain ontology



Example: Functionality of a Service

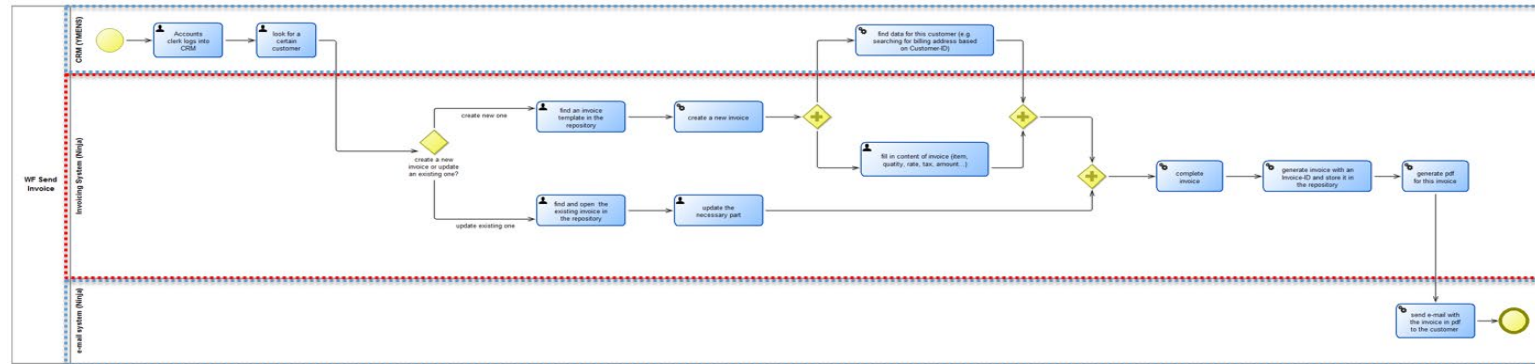


Domain Ontology:  
APQC Process Classification Framework





# All Application Knowledge Concepts are defined in the Ontology



## Functionality

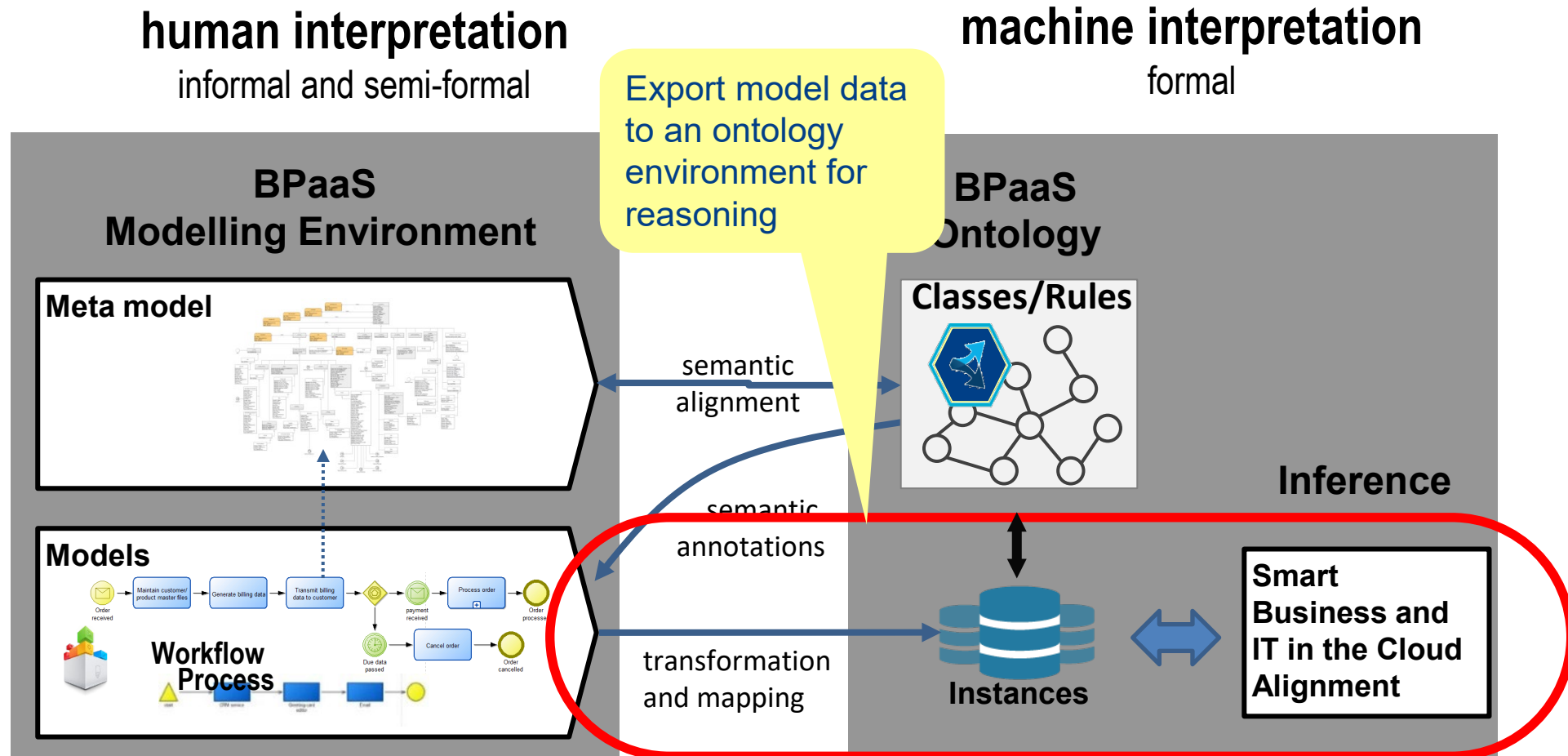
<p>APQC</p> <p>APQC Annotation: apqc:9_2_2_3_Transmit_billing_data_to_customers_10796</p> <p>Set APQC</p> <p>Set APQC</p>	<p>Description</p> <p>Functional</p> <p>Data Security Infrastructure</p> <p>Performance</p> <p>Support Service</p> <p>Payment</p>
<p>Action</p> <p>Action Annotation: fbpdo:Send</p> <p>Set Action</p> <p>Set Action</p>	
<p>Object</p> <p>Object Annotation: fbpdo:Invoice</p> <p>Set Object</p> <p>Set Object</p>	

## Non-functional requirements

<p>Availability</p> <p>Availability in %: 99.999</p>	<p>Description</p> <p>Functional</p> <p>Data Security Infrastructure</p>
<p>Capacity</p> <p>Max Available Data Storage in GB per Month: 5.000000</p> <p>Maximum Simultaneous Connections: 500</p> <p>Maximum Simultaneous Service Users: 500</p>	<p>Performance</p> <p>Support Service</p> <p>Payment</p>
<p>Response Time</p> <p>Max Average Response Time: 00:00:00:00:01</p>	



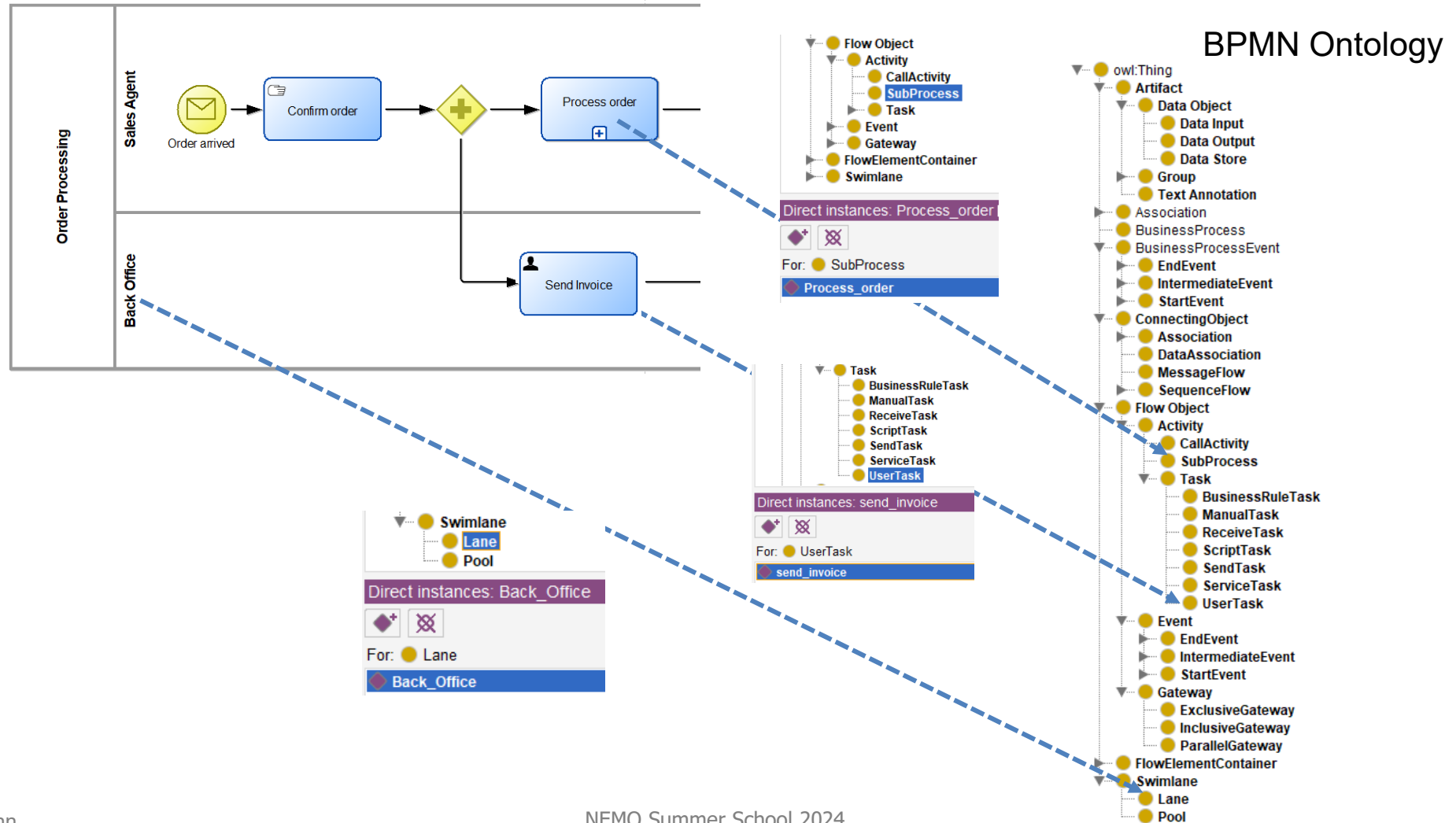
# Transformation and Mapping for Inferencing



From: CoudSocket Project

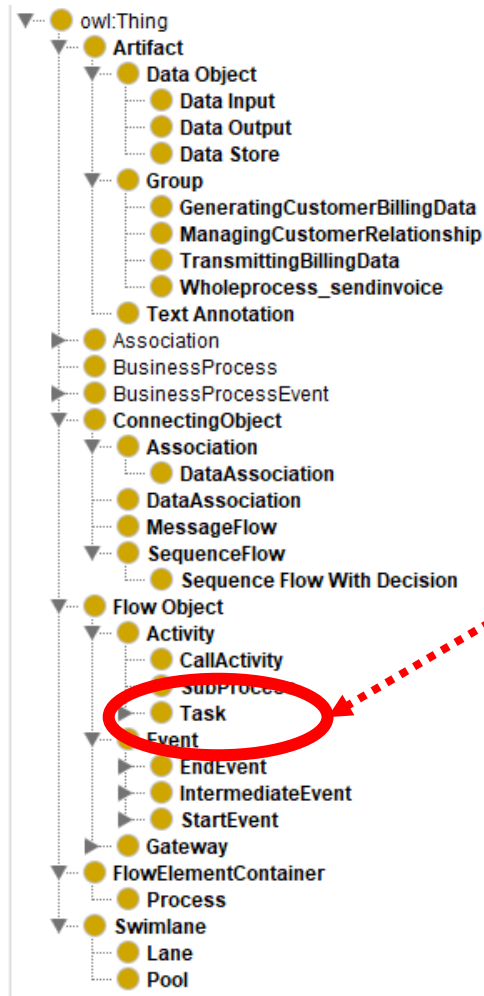
# Transformation and Mapping

The model elements are exported as instances of ontology classes



# Modelling Language and Application Domain Ontologies

BPMN Ontology:



Direct instances: send\_invoice

For: UserTask

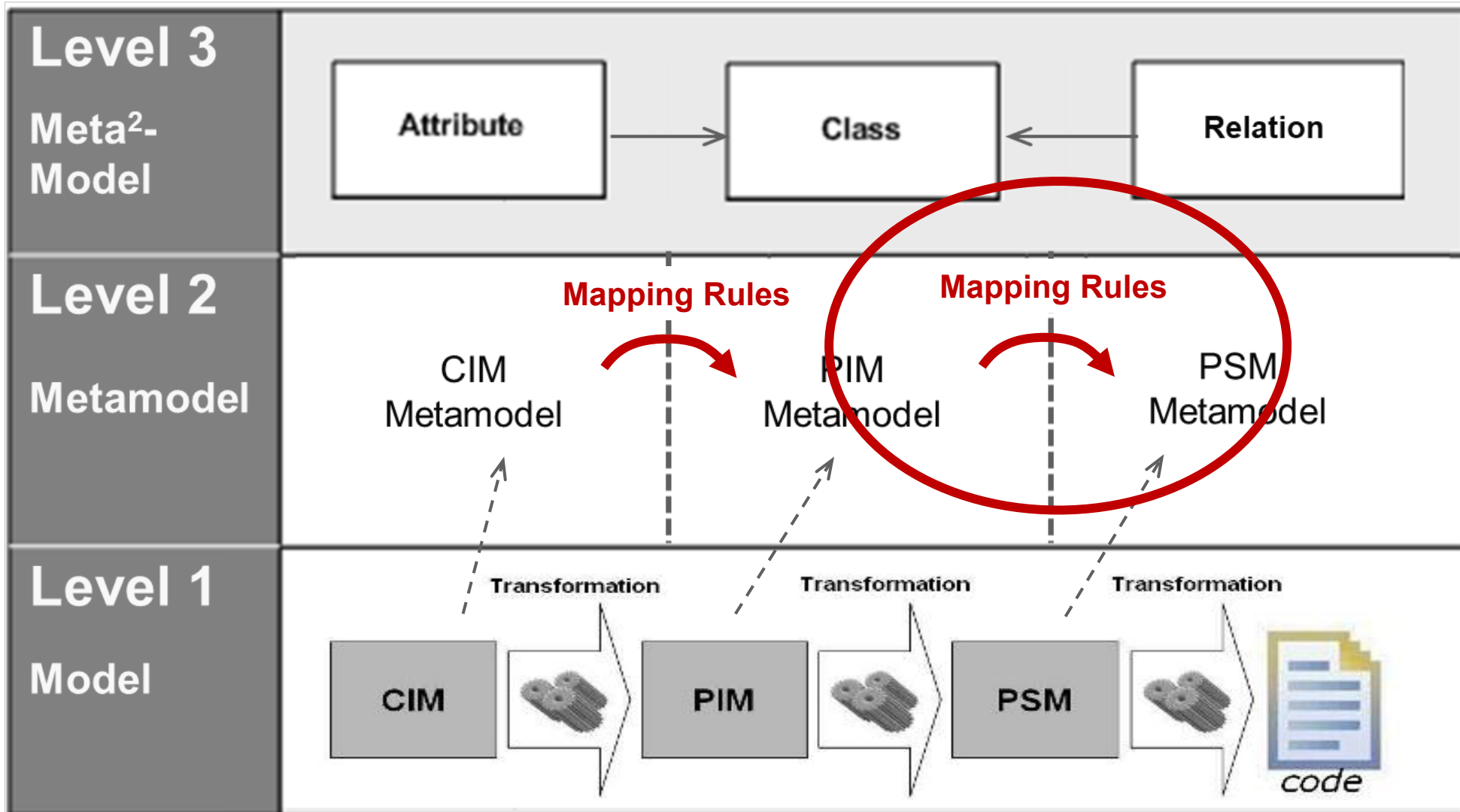
send\_invoice

functionality

Domain Ontology:  
APQC Process Classification Framework



# Supporting PIM to PSM Mapping



# Cloud Services

- 355 Cloud Services from 4 Marketplaces – with 13.098 specification values
- Imported as Ontology Instances

The image displays five vertical screenshots of data tables, each representing a different marketplace. The tables are filled with a dense grid of small text, likely representing the 13,098 specification values for 355 cloud services. The first four tables have a yellow header bar, while the fifth table has a red header bar. The tables are arranged horizontally from left to right, with the fifth table being significantly smaller than the others.

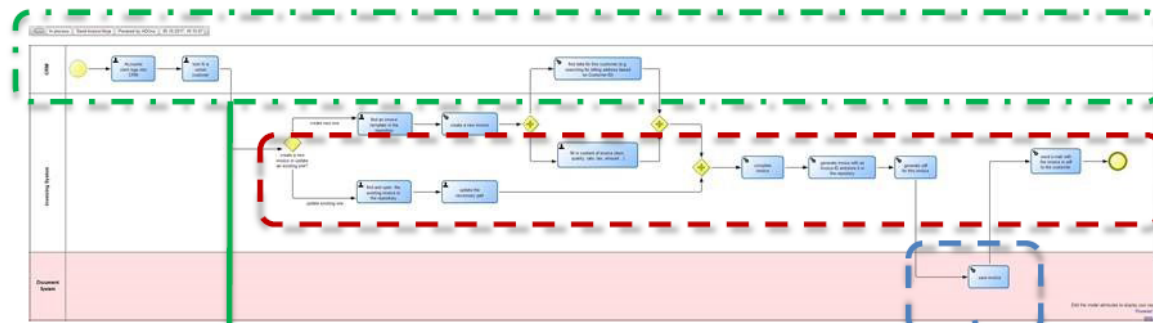
## *Inference: Retrieve suitable Cloud Services*

- Semantic rules (in SPARQL),
  - ◆ For Business-IT mapping
  - ◆ To turn implicit knowledge into explicit knowledge
- Example: If a service offers a backup retention time up to one year, implies also

- ◆ Up to six months
- ◆ Up to one month
- ◆ Up to one week
- ◆ Up to one day

```
[CONSTRUCT {  
  ?cservice bpaas:cloudServiceHasBackupRetentionTime bpaas:up_to_6_months .  
  ?cservice bpaas:cloudServiceHasBackupRetentionTime bpaas:up_to_1_month .  
  ?cservice bpaas:cloudServiceHasBackupRetentionTime bpaas:up_to_1_week .  
  ?cservice bpaas:cloudServiceHasBackupRetentionTime bpaas:up_to_1_day .  
}  
WHERE{  
  ?cservice rdf:type bpaas:CloudService .  
  ?cservice bpaas:cloudServiceHasBackupRetentionTime bpaas:up_to_1_year .  
}]
```

# Cloud Services for the given Business Process Model



**Invoice Management System**

**Specifications**

**Action:** Generate

**Object:** Invoice

**APQC:** 9.2.2.2 Generate Customer Billing Data

**Storage Location:** Europe

**Availability:** 99.999%

**N. simultaneous users:** 100



CLOUD IS NO LIMIT  
**YMENSE**



**Customer Management System**  
**Specifications**

**Action:** Manage

**Object:** Customer

**APQC:** 3.5.2.4 Manage Customer Relationship

**Storage Location:** Europe

**Availability:** 99.999%



**Action:** Manage

**Object:** Document

**APQC:** 9.2.2.2 Generate Customer Billing Data

**Storage Type:** Storage Service

**Storage Location:** Europa

**Availability:** 99.999%



## *Drawbacks of Semantic Lifting*

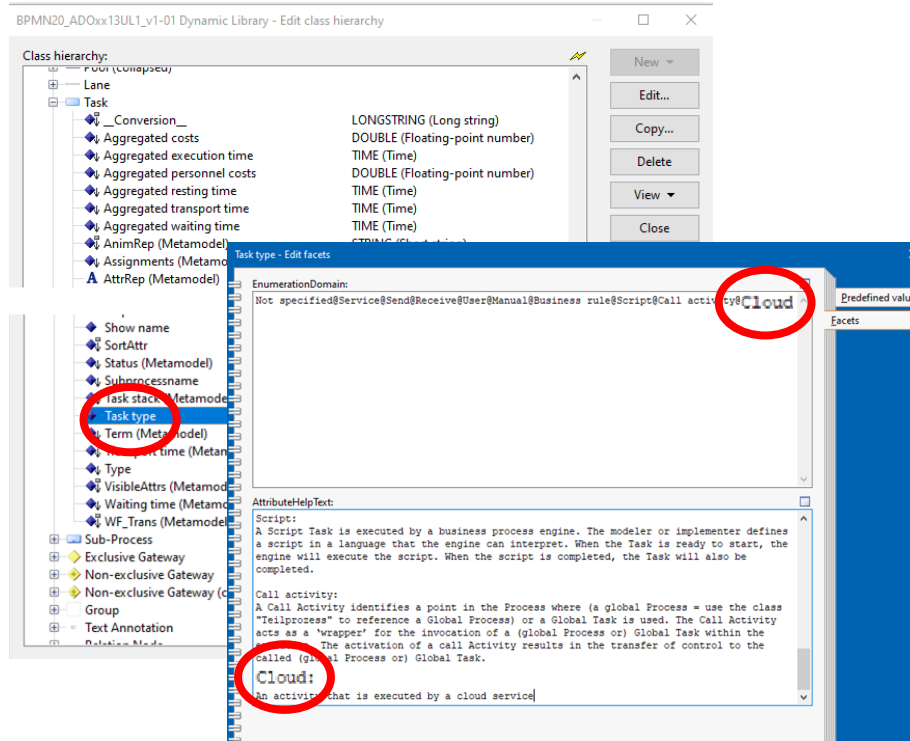
- Separate Environments
  - ◆ Modelling and Metamodelling
  - ◆ Ontology
- Inconsistency
  - ◆ Metamodel and ontology must represent the same semantics but are maintained independently
  - ◆ Each change in metamodel must be reproduced in the ontology and vice versa
- Effort
  - ◆ After each change the models must be translated again into the ontology instances

# Example: New Model Element

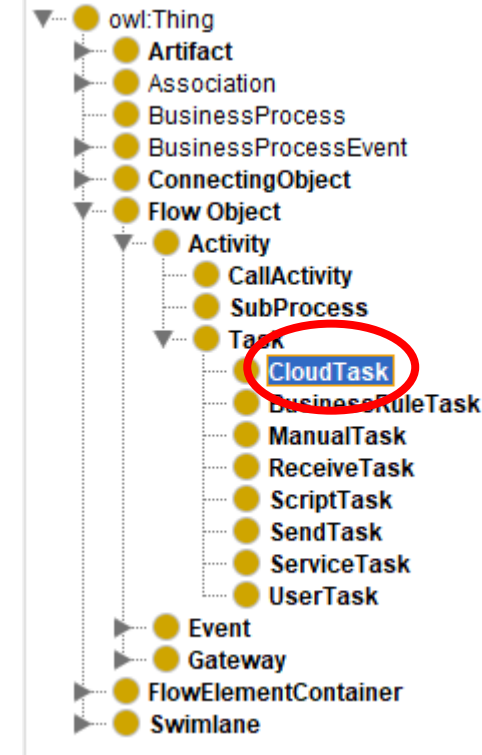
## ■ New task type: Cloud Task



Change in the meta model:



Change in the ontology:

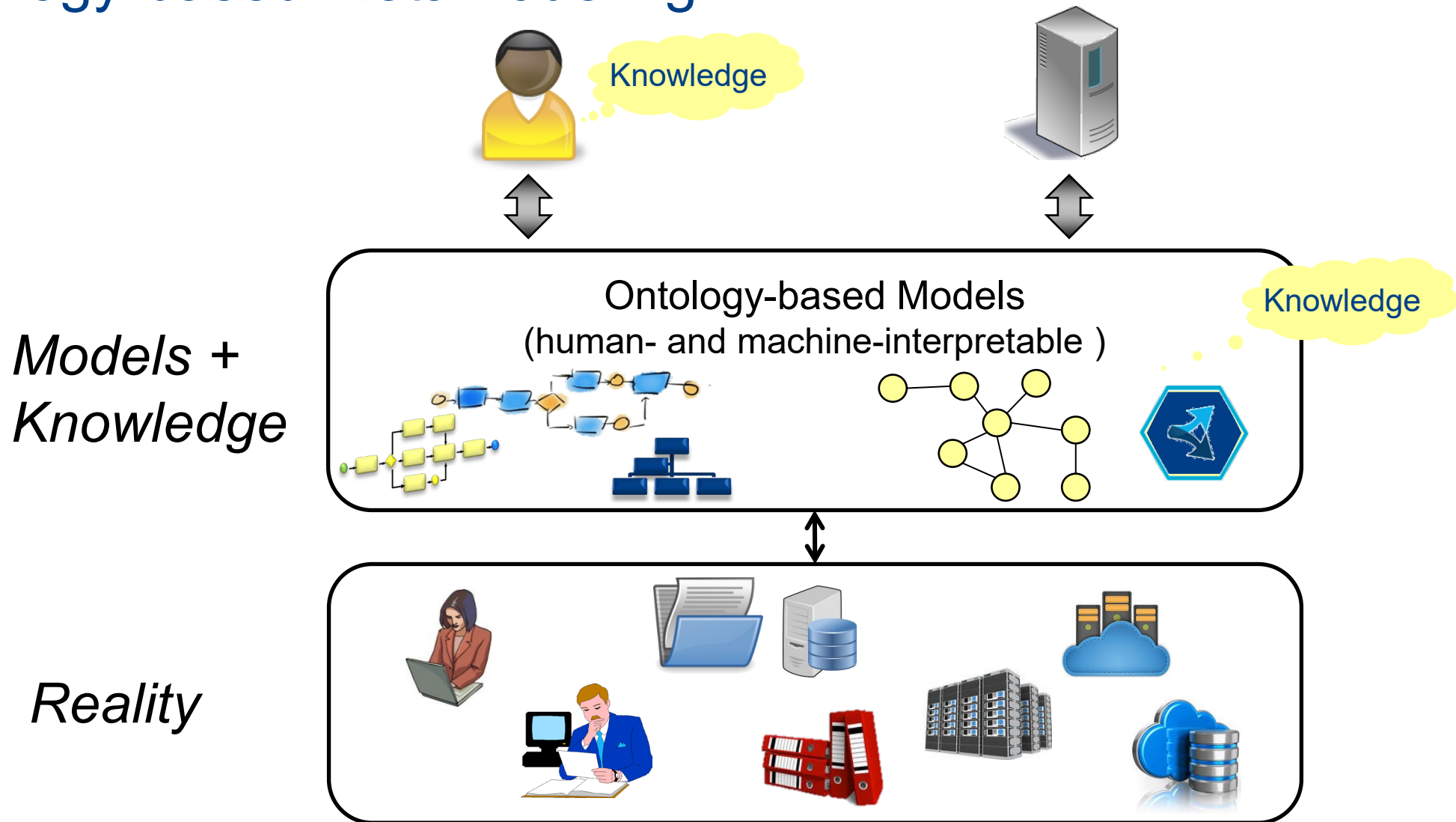


# *Ontology-based Metamodeling*

## *Objective*

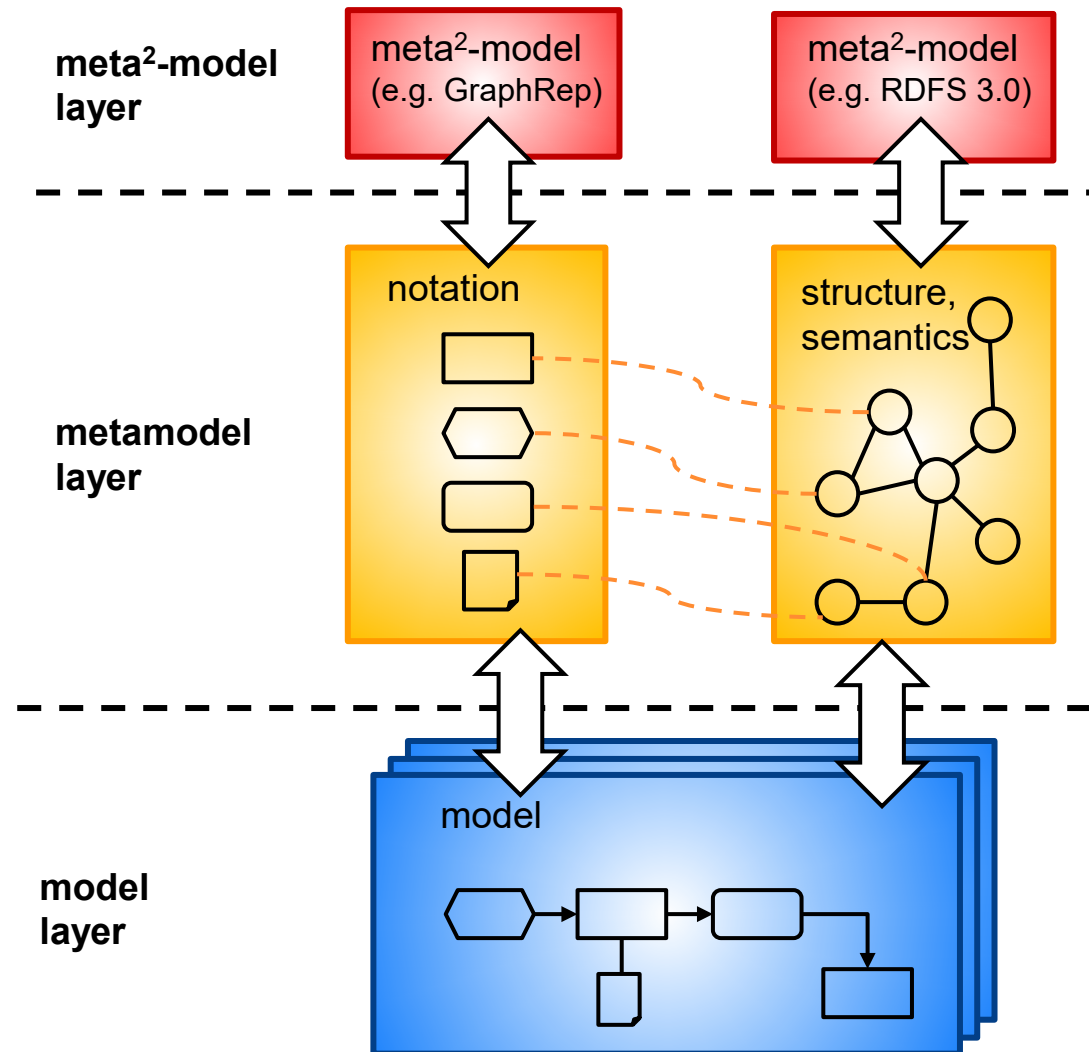
Ensure a precise shared interpretation  
of new modeling constructs to both  
**humans and machines**

# Ontology-based Metamodeling



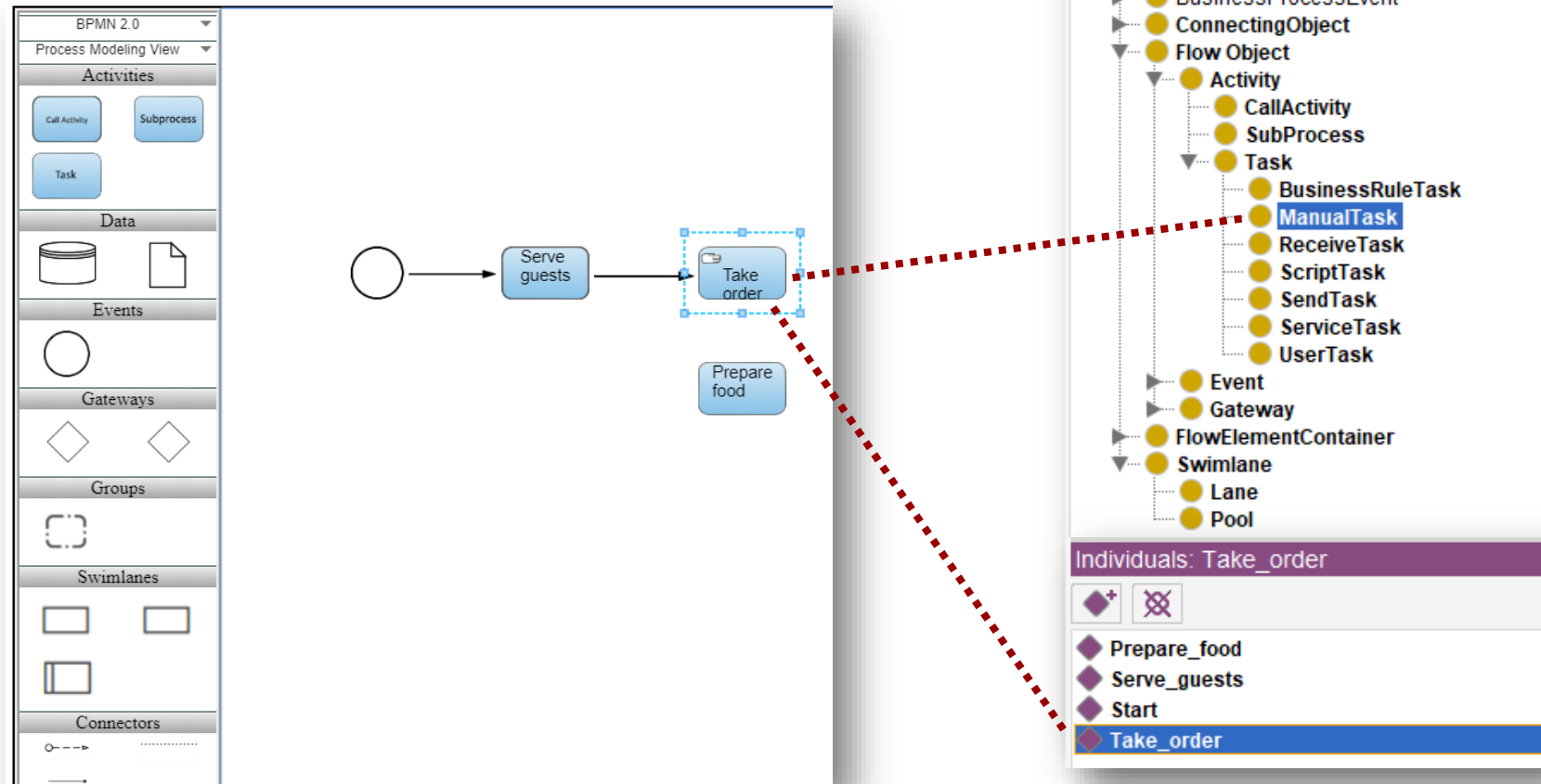
(Laurenzi et al. 2018)

# Ontology-based Metamodeling

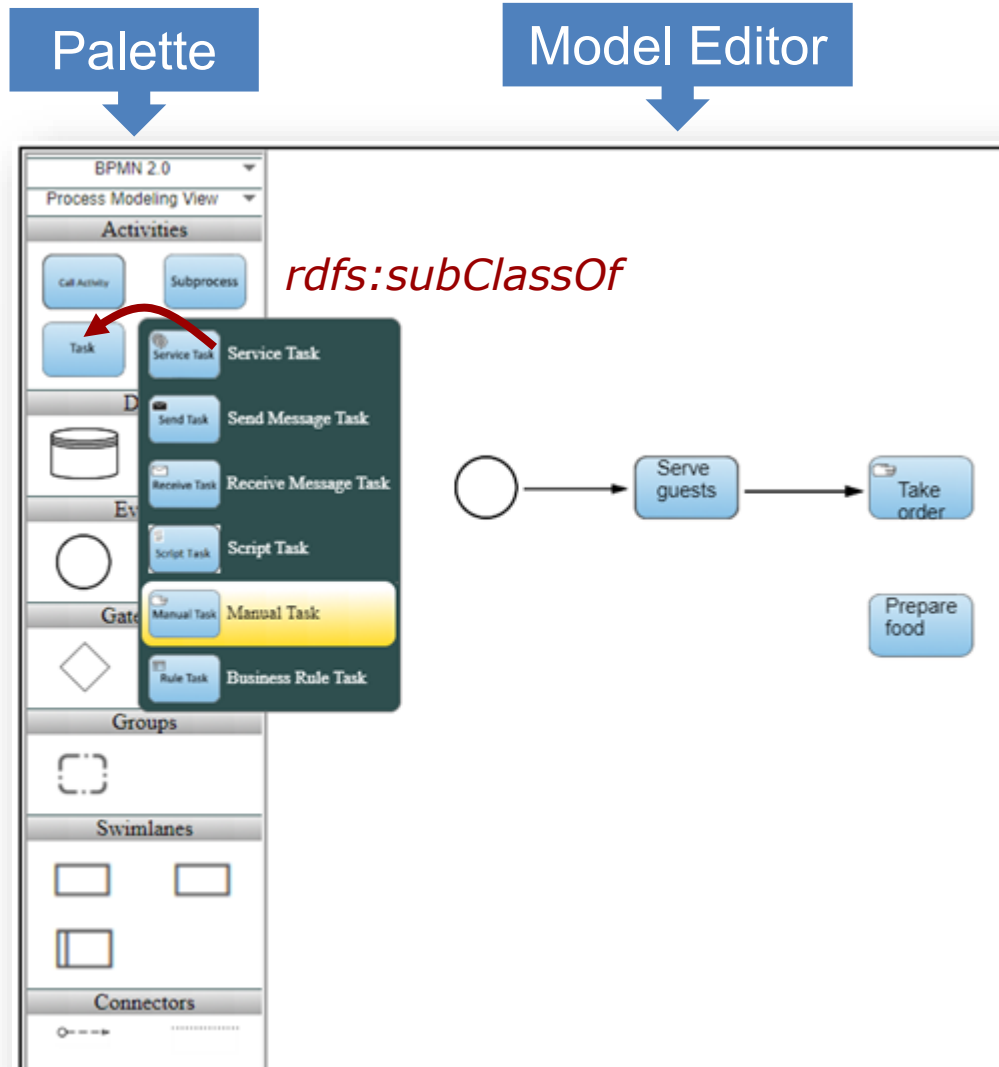


# AOAME – A Tool for Ontology-Based Modelling

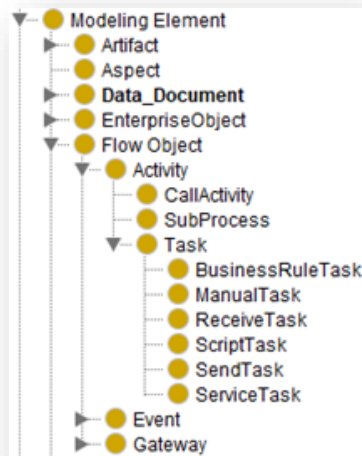
- Single environment for modelling and ontology
- Model elements are directly created as instances in the ontology



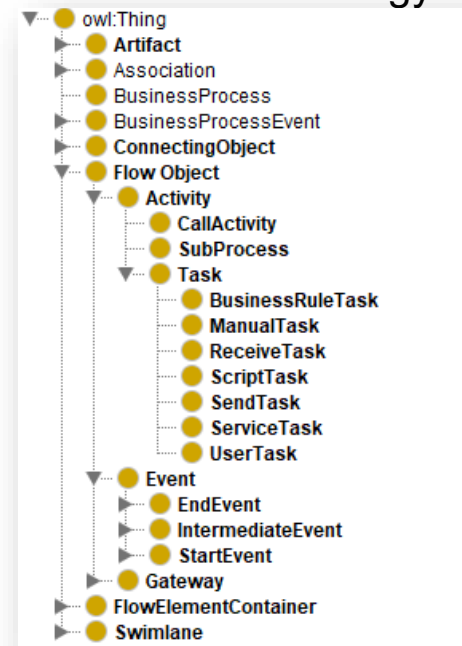
# Ontology-Based Modeling in AOAME



Palette Ontology (excerpt)



BPMN Meta model Ontology





## 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#>

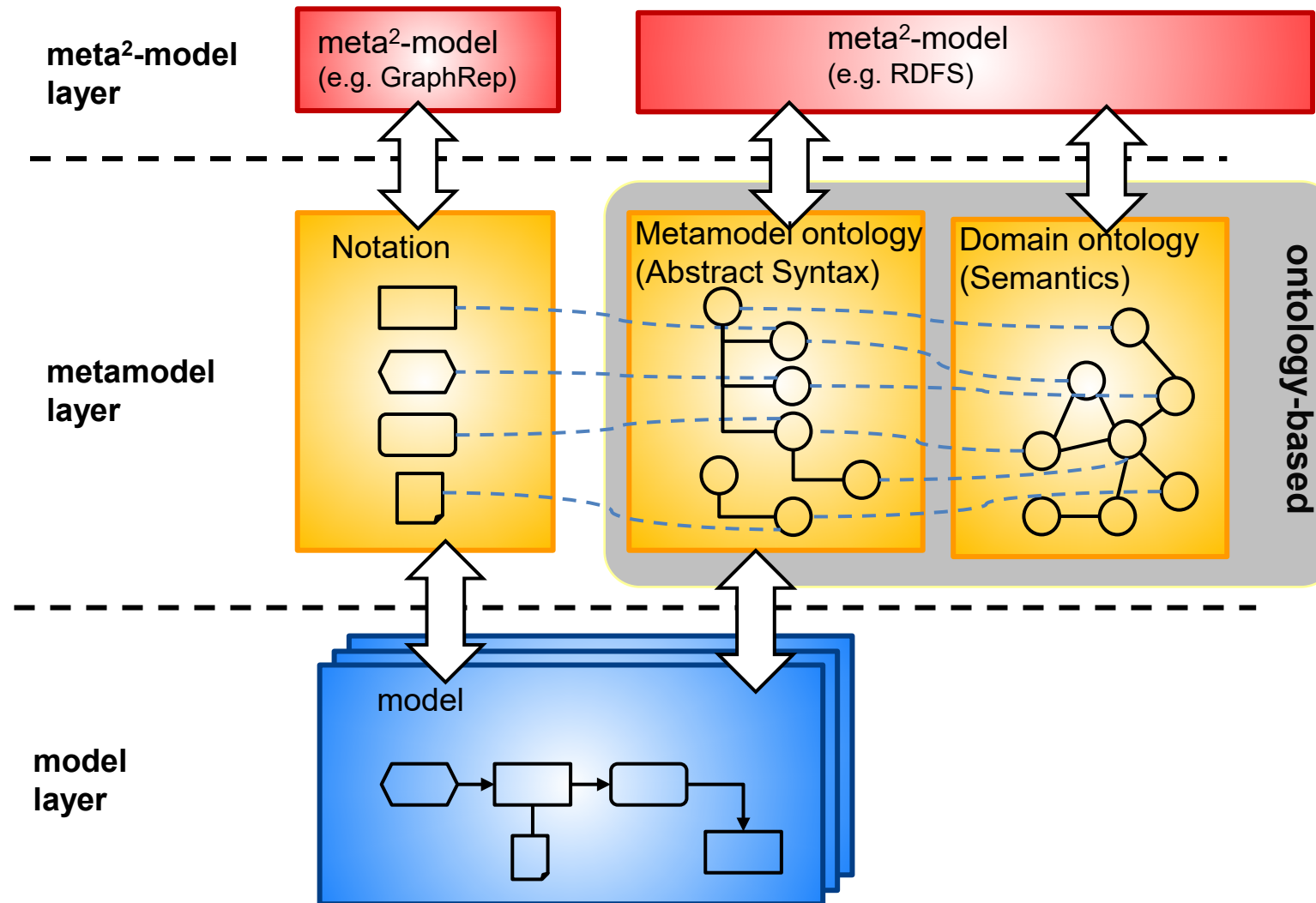
SELECT ?model ?shape ?task ?l
WHERE {
  ?model rdfs:label "Serve Guests".
  ?model mod:modelHasShape ?shape.
  ?shape mod:shapeVisualisesConceptualElement ?task.
  ?task rdf:type bpmn:Task .
  ?shape rdfs:label ?l.
}
```

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

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



# Representing Complete Content as Ontology

## ■ Meta model Ontology:

- ◆ Concepts of the meta model are classes in an ontology
- ◆ Modelling = creating instances of classes



## ■ Application Domain Ontology:

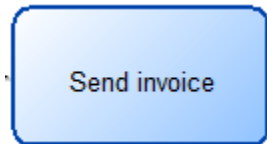
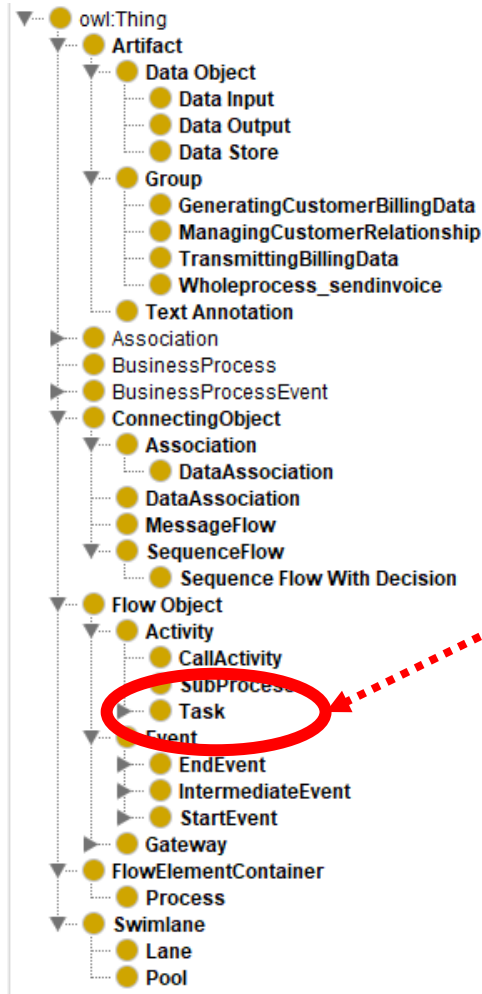
- ◆ Model elements are annotated with domain knowledge from application domain ontology



- Ontology reasoning can be applied to the complete content knowledge in the models

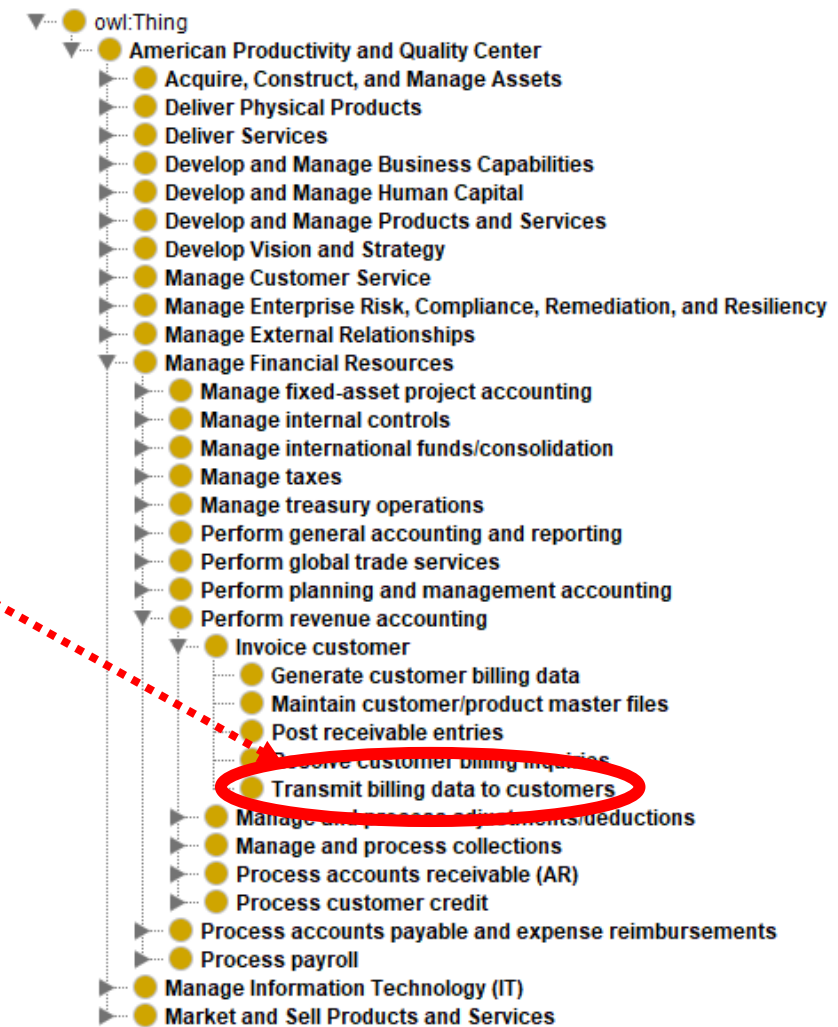
# Language and Domain Ontologies

BPMN Ontology:



functionality

Domain Ontology:  
APQC Process Classification Framework



# Computer needs up-to-date Knowledge, too

© 2000 Randy Glasbergen.  
www.glasbergen.com

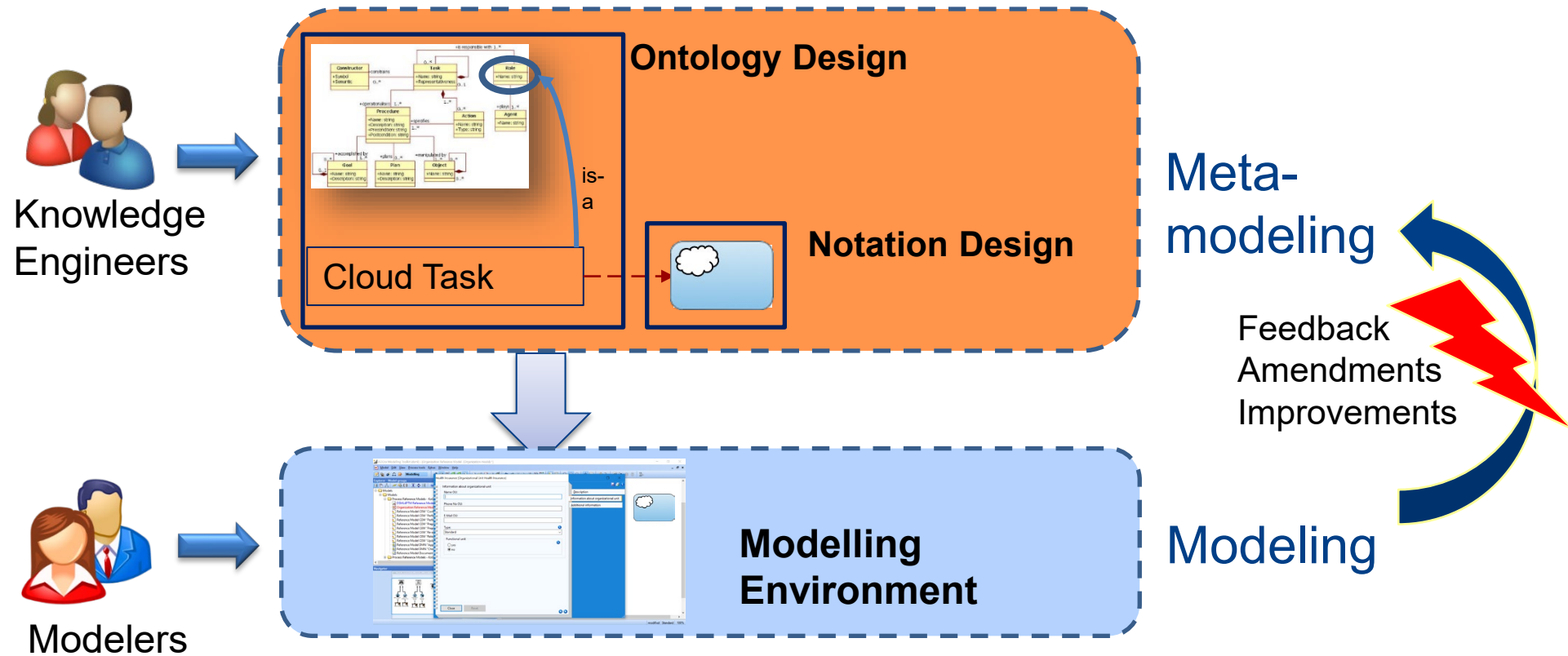


~~THE COMPUTER SAYS I NEED TO UPGRADE MY BRAIN  
TO BE COMPATIBLE WITH ITS NEW SOFTWARE."~~

**THE COMPUTER NEEDS TO UPGRADE ITS ONTOLOGY  
TO BE COMPATIBLE WITH THE NEW MODEL**

# *Agile Modeling and Meta-Modeling*

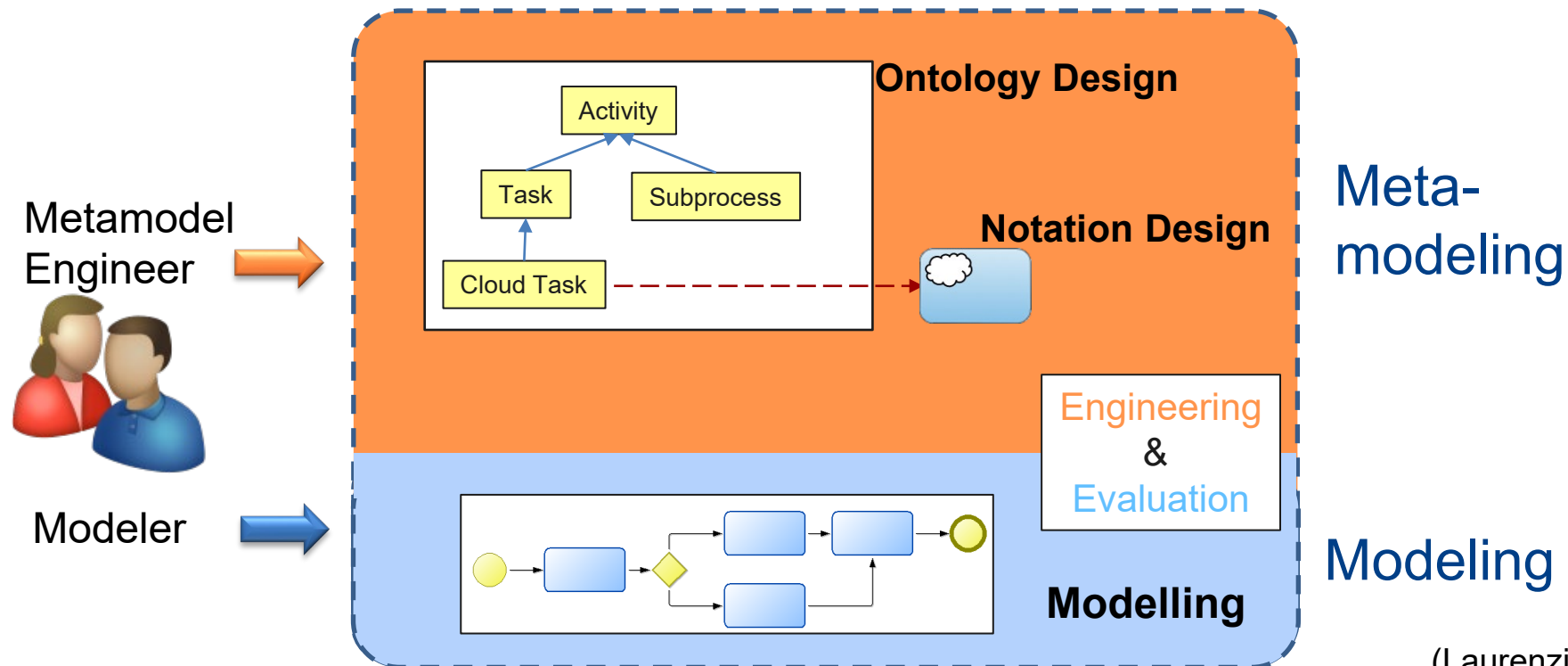
# Change of Metamodel



*Time-consuming engineering effort!*

# Integration Modeling and Metamodeling in a Single Environment

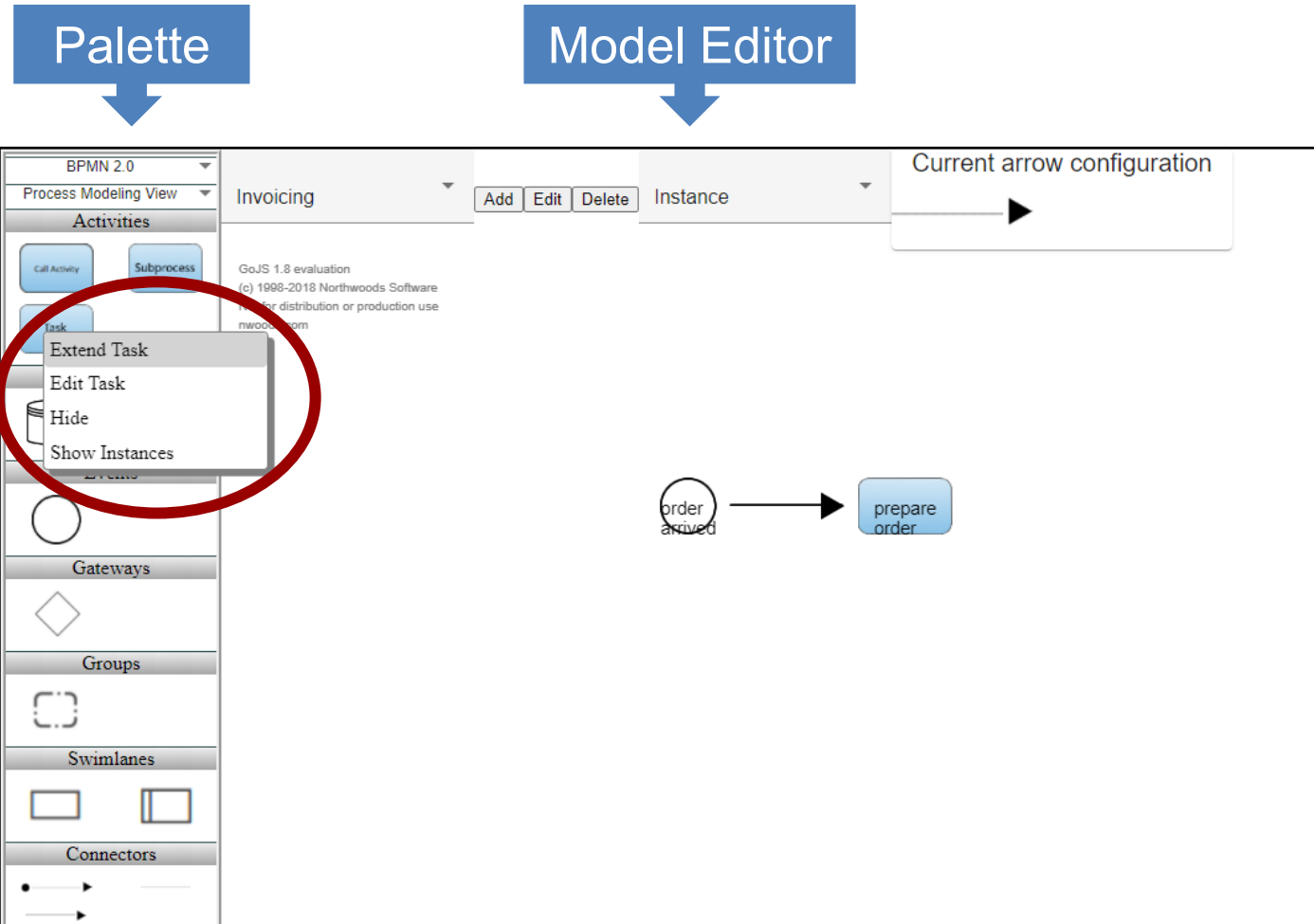
- Tight collaboration between metamodel developer and modeler
- Modeler can also take the role of metamodel developer



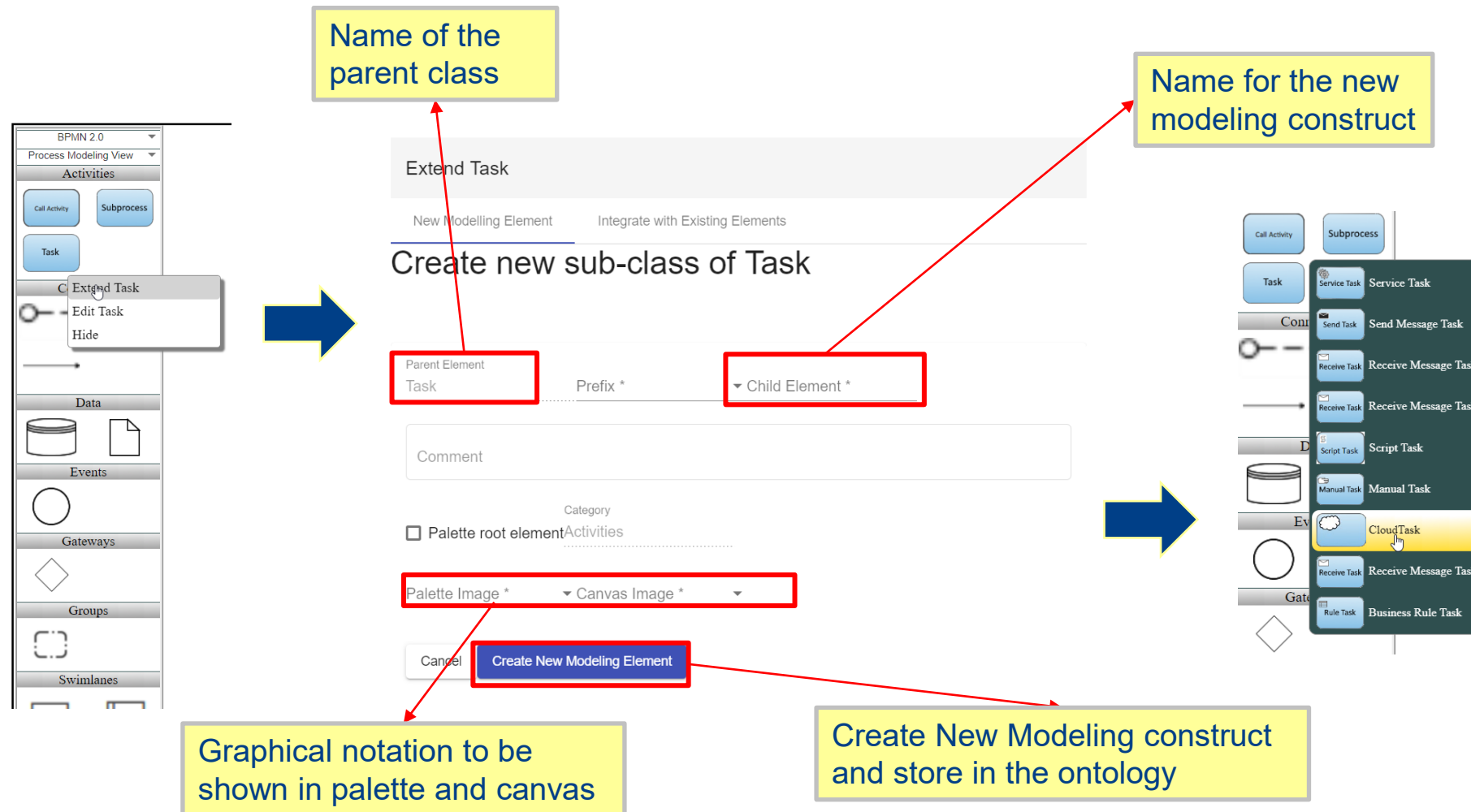
(Laurenzi et al. 2018)



# Extending AOAME Modeling Languages – on the fly



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



# Semantic Alignment in AOAME

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

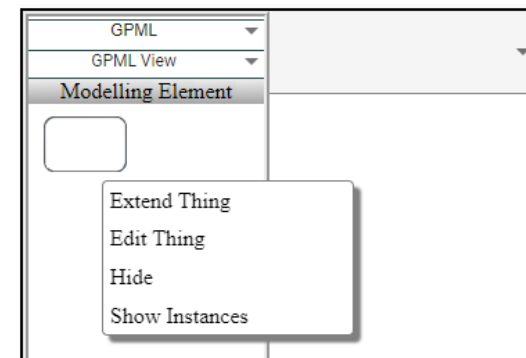
The screenshot displays the AOAME interface. The top navigation bar includes tabs for 'CloudTask', 'Datatype', 'Bridging Connector', and 'Semantic Mapping', with the latter highlighted by a red circle. Below the tabs, the 'Edit CloudTask' panel is visible, showing fields for 'Prefix' (bpmn), 'New Label \*' (CloudTask), 'Comment', 'Palette Image', 'Canvas Image \*', and 'Arrow Stroke'. A 'Create New Relation' dialog is overlaid on the right, titled 'Relations for CloudTask'. It contains a 'Create new ObjectProperty' section with a 'Label \*' field containing 'paymentplan' and a text input field with 'bpaas:PaymentPlan'. A 'Create New Domain Element' button is present below the input field. At the bottom right of the dialog is a 'Create Relation' button. The dialog also has 'Cancel' and 'Ok' buttons at the bottom.

## The Cool Thing

AOAME is a graphical ontology development tool



- Start with an empty ontology: Just a root node
- Step by step expand the ontology
- Result
  - ◆ Domain ontology
  - ◆ Graphical representation of classes
- ***Can be done by business people***



# 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
  - ◆ Marco Di Ianni
  - ◆ Jan Eich

# The End

