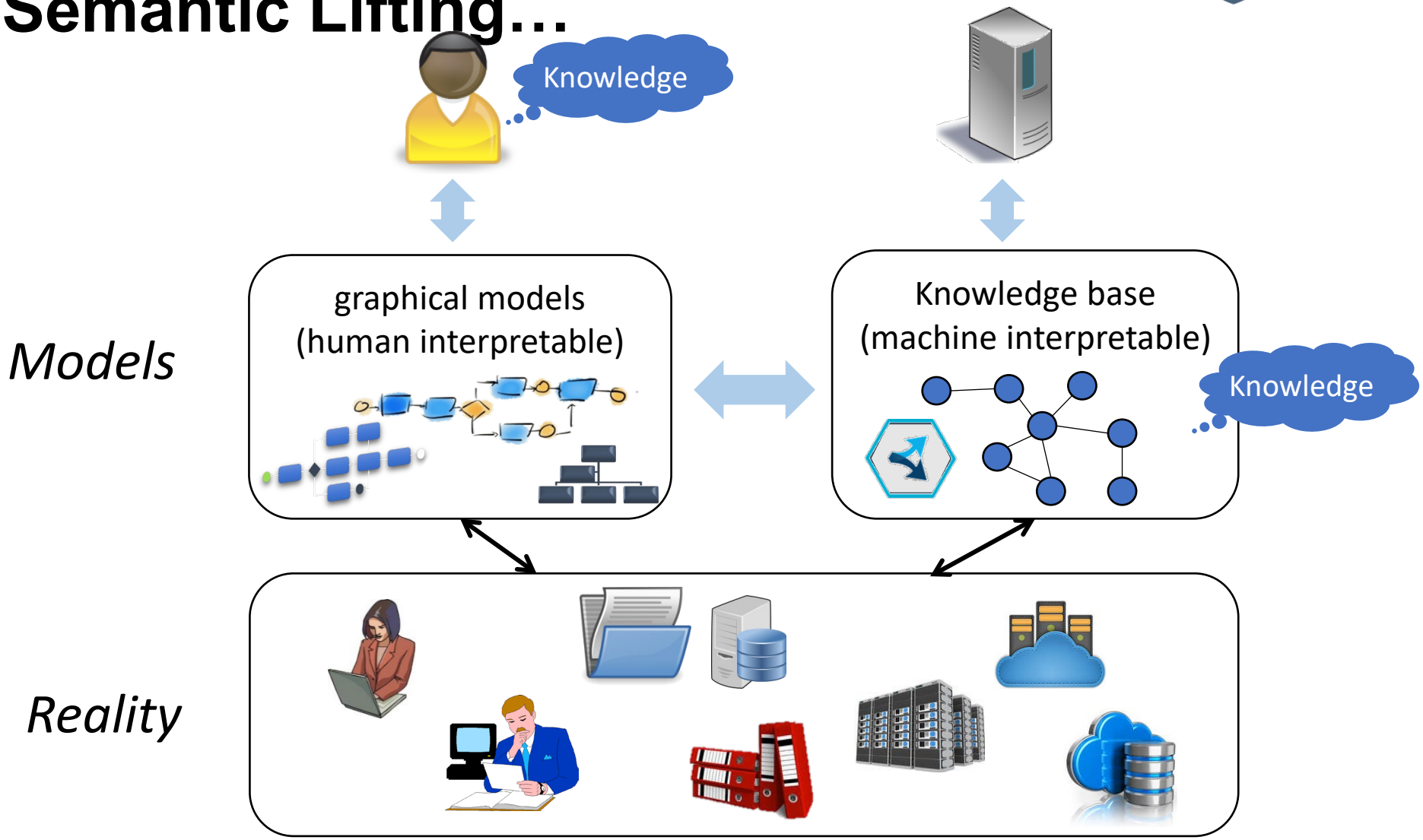
University of Applied Sciences and Arts Northwestern Switzerland
School of Business

Università di Camerino

Ontology-based Meta-modelling

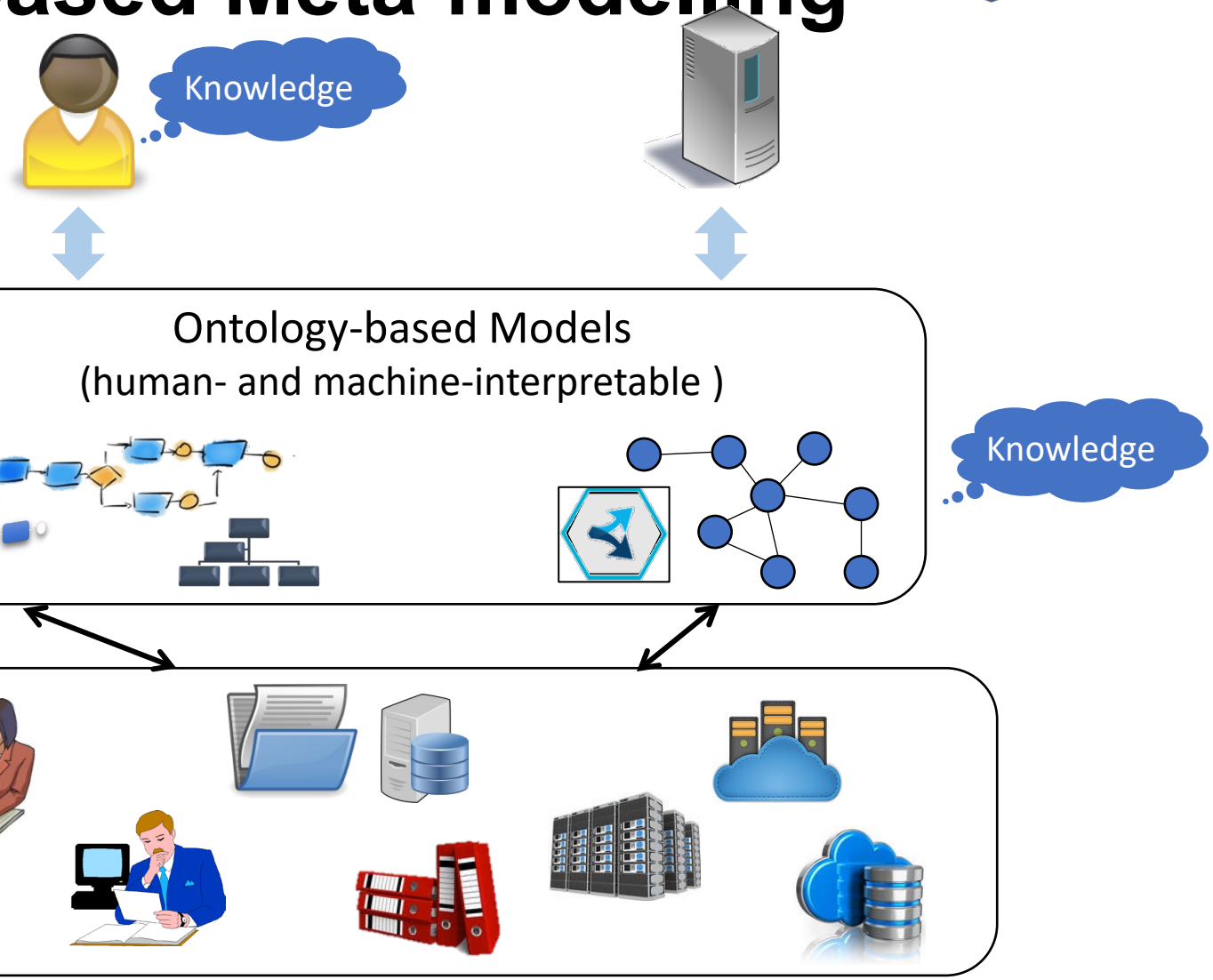
Knowledge Engineering SS25
MSc Computer Science
Camerino, 27/05/2025
Prof. Emanuele Laurenzi

From Semantic Lifting…



…to Ontology-based Meta-modelling

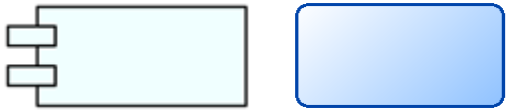
Models
+ Knowledge



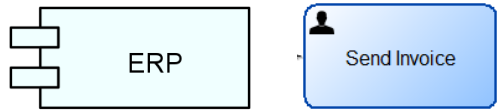
Reality

Objective:
Representing Complete Content as Ontology

- Meta-model Ontology:
  - Concepts of the meta model are classes in an ontology
  - Modelling = creating instances of classes

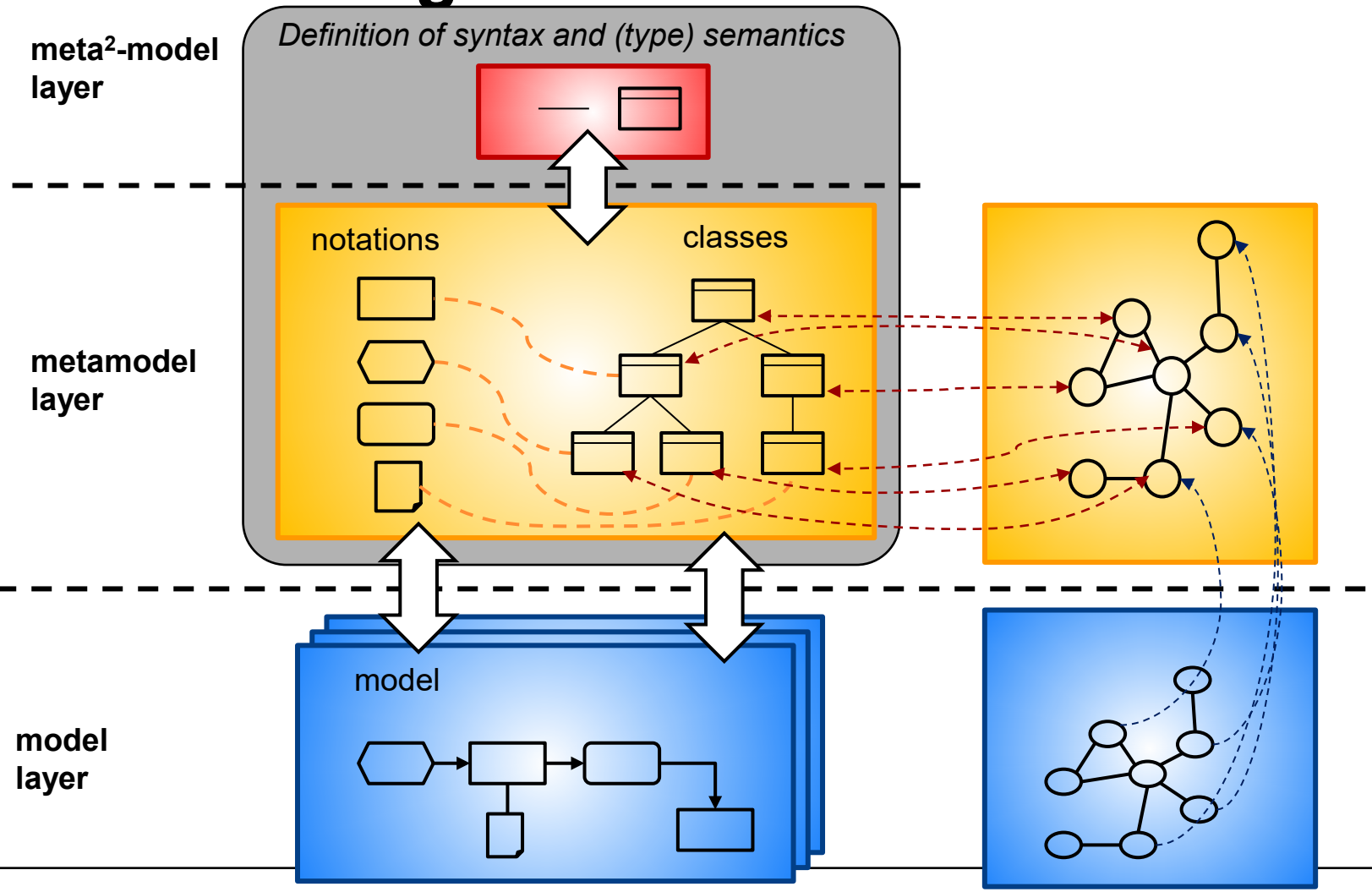


- Application Domain Ontology:
  - Model elements are annotated with domain knowledge from application domain ontology

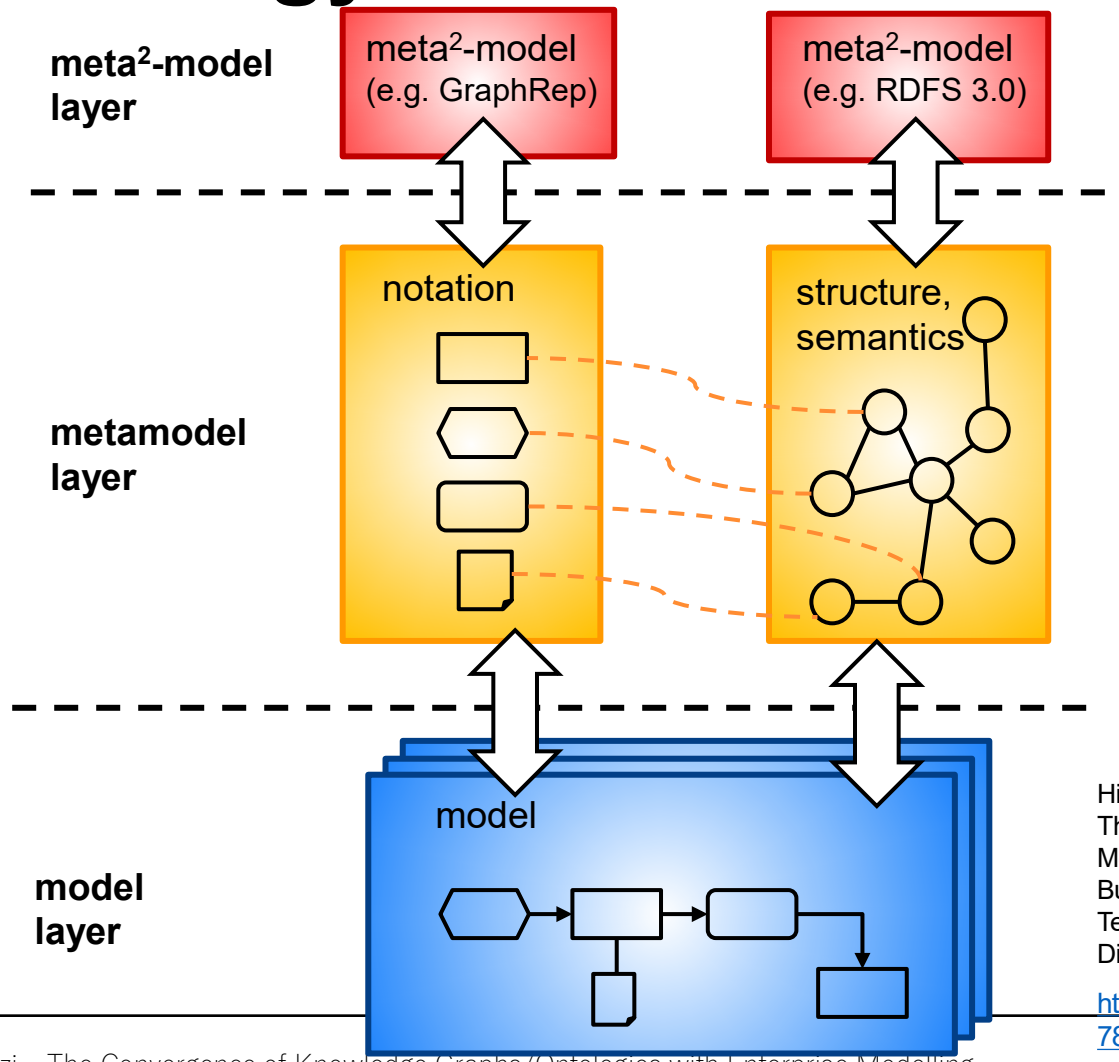


- Ontology reasoning can be applied to the full content knowledge in the models.

From Semantic Lifting…



…to Ontology-based Meta-modelling

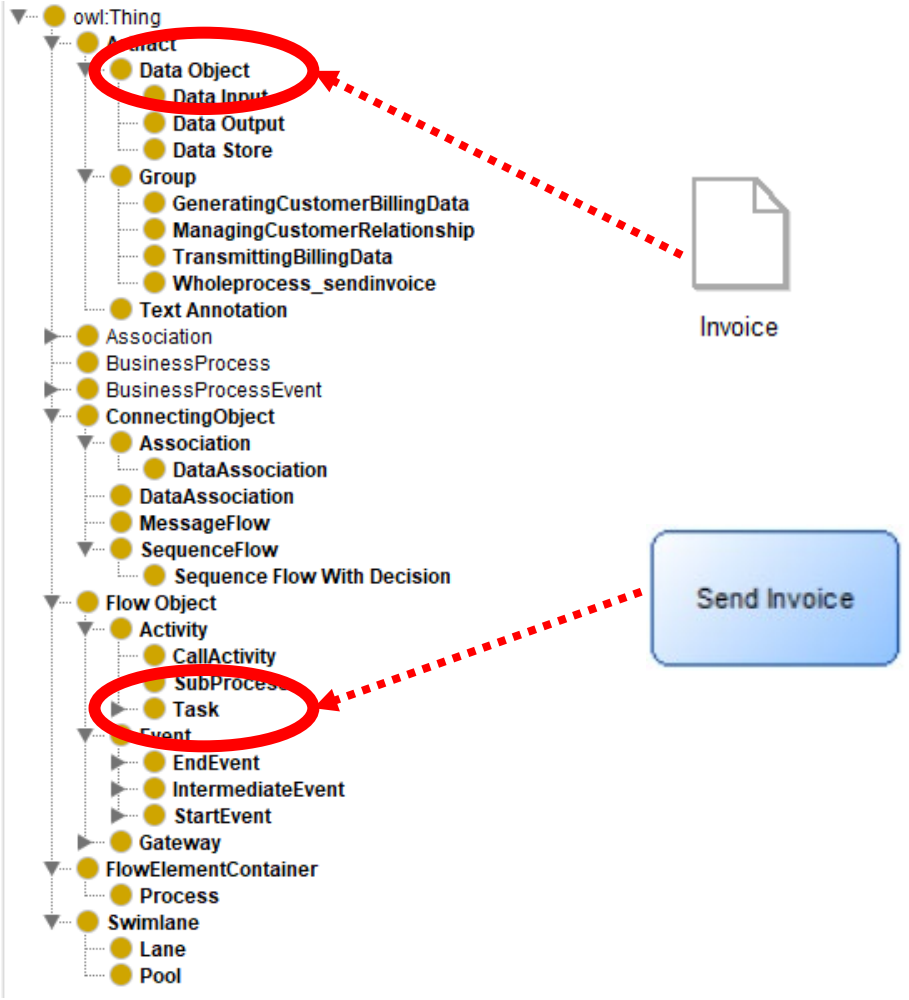


Hinkelmann, K., Laurenzi, E., Martin, A. Thönssen, B. (2018). Ontology-Based Metamodeling. In R. Dornberger, ed. Business Information Systems and Technology 4.0 - New Trends in the Age of Digital Change. Springer Berlin Heidelberg.

https://link.springer.com/chapter/10.1007/978-3-319-74322-6_12

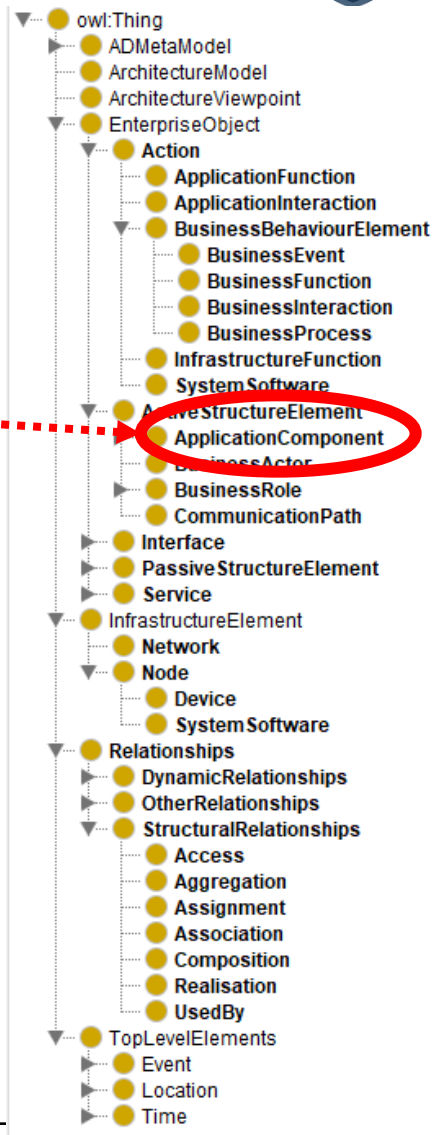
Example of Metamodel Ontologies and Graphical Instantiation

BPMN



Invoice

Archimate



Integration of Linguistic view and Domain View in the Ontology-based Metamodelling

Knowledge Representation of Content: Linguistic View vs. Domain View

- **Linguistic View:** the specification that relates to a modelling language.
- **Domain View:** the specification that relates to a domain of discourse.

ERP (Enterprise Resource Planning) is…

…an Application Component in ArchiMate.

…a type of software system that organizations use to manage day-to-day business activities such as accounting, procurement, project management, risk management and compliance, and supply chain operations. […] Among the most widely used ERPs there are Oracle NetSuite, Microsoft Dynamics 365, and Oracle ERP CloudSAP ERP.

C. Atkinson and T. Kuhne, "Model-driven development: a metamodeling foundation," in *IEEE Software*, vol. 20, no. 5, pp. 36-41, Sept.-Oct. 2003, doi: 10.1109/MS.2003.1231149.

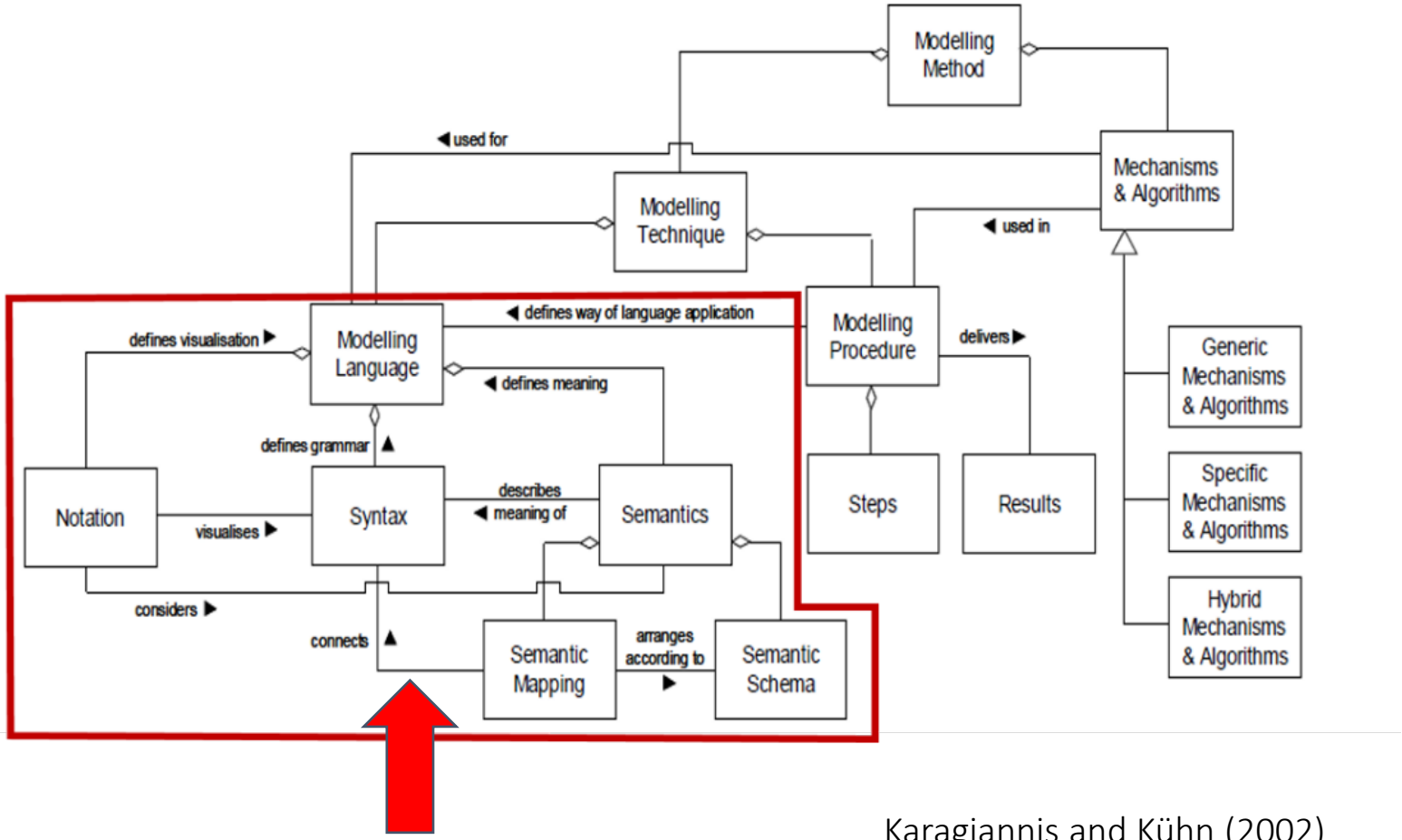
Integration of the Domain View (semantics)

- Semantics of a modelling language can be specified in:
  - the meta-model
  - the semantic domain (akas domain view), that provides information regarding the domain of discourse,
  - the semantic mapping, that maps the abstract syntax into the semantic domain.
- The related mathematical formula for the semantic mapping is as follows (Harel & Rumpe 2000):

  - where the semantic mapping (M) relates concepts from the abstract syntax (L) to the domain semantic (S).

$$M: L \xrightarrow{maps} S$$

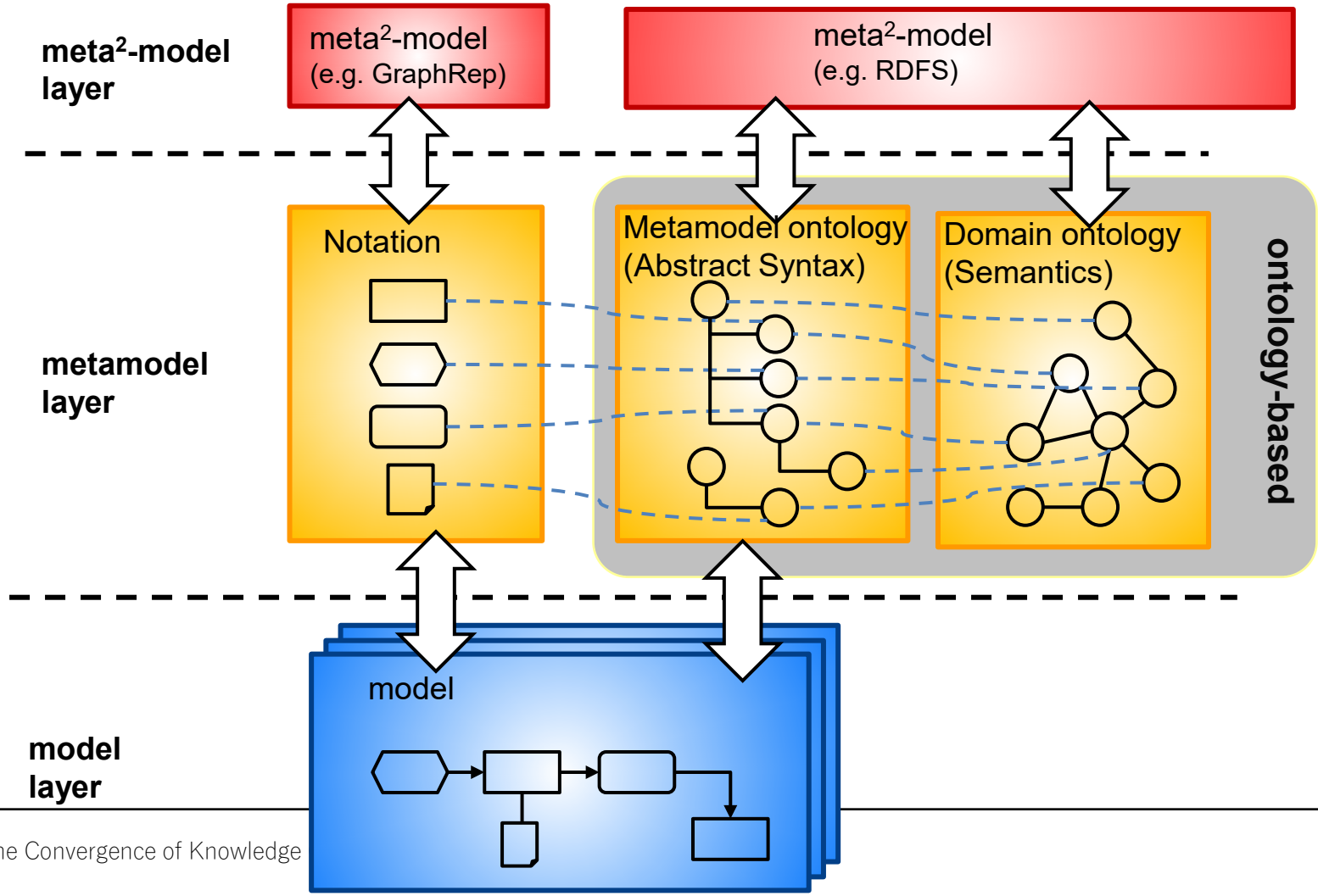
Modelling Method Framework



Karagiannis and Kühn (2002)

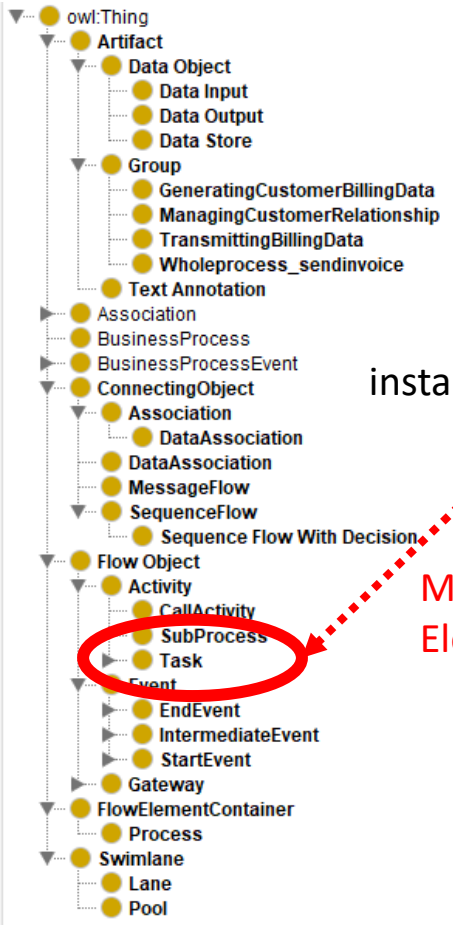
Visualisation of the Semantic Mapping

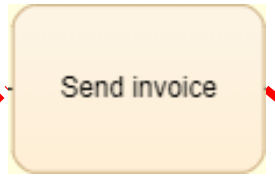
Ontology-based Metamodeling (2): Ontologies for Linguistic View and Domain View



Example of Metamodel Ontology and Domain Ontologies

Meta-model Ontology (BPMN)





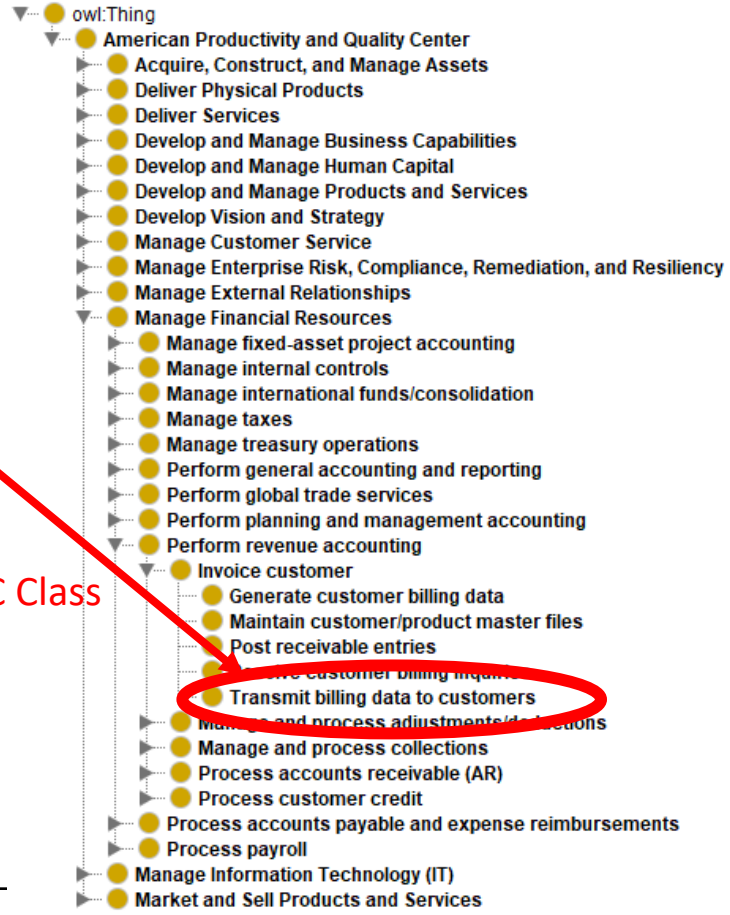
instanceOf

Modelling Element

hasAPQC

APQC Class

Domain Ontology:
APQC Process Classification Framework



Retaining knowledge about graphical notation

Palette Ontology (an excerpt)

- po:PaletteCategory (43)
- po:PaletteConstruct (277)
  - po:PaletteConnector (28)
  - po:PaletteElement (248)

po:SubReceiveActivity	po:PaletteElement
po:Subprocess	po:PaletteElement
po:Subway	po:PaletteElement
po:SystemSoftware	po:PaletteElement
po:Table	po:PaletteElement
po:Tablet	po:PaletteElement
po:Task	po:PaletteElement
po:Task_4DSML4PTM	po:PaletteElement
po:Team	po:PaletteElement
po:TechnologyArtifact	po:PaletteElement
po:TechnologyCollaboration	po:PaletteElement
po:TechnologyDevice	po:PaletteElement
po:TechnologyEvent	po:PaletteElement
po:TechnologyFunction	po:PaletteElement
po:TechnologyInteraction	po:PaletteElement
po:TechnologyInterface	po:PaletteElement

Resource Form

Name: po:Task

Annotations

rdfs:label
S Task

Other Properties

po:paletteConstructHasHeight
I 70

po:paletteConstructHasModelImage
S Task.png

po:paletteConstructHasPaletteThumbnail
S Thumbnail_Task.png

po:paletteConstructHasParentPaletteConstruct

po:paletteConstructHasWidth
I 100

po:paletteConstructIsGroupedInPaletteCategory
po:Category_Activities4BPMNProcessModelingView

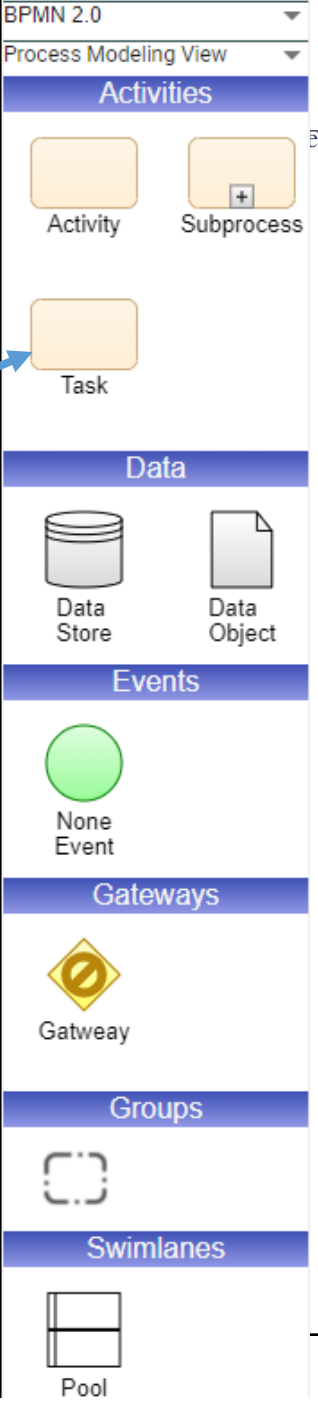
po:paletteConstructIsHiddenFromPalette
B false

po:paletteConstructIsRelatedToModelingLanguageConstruct
bpmn:Task

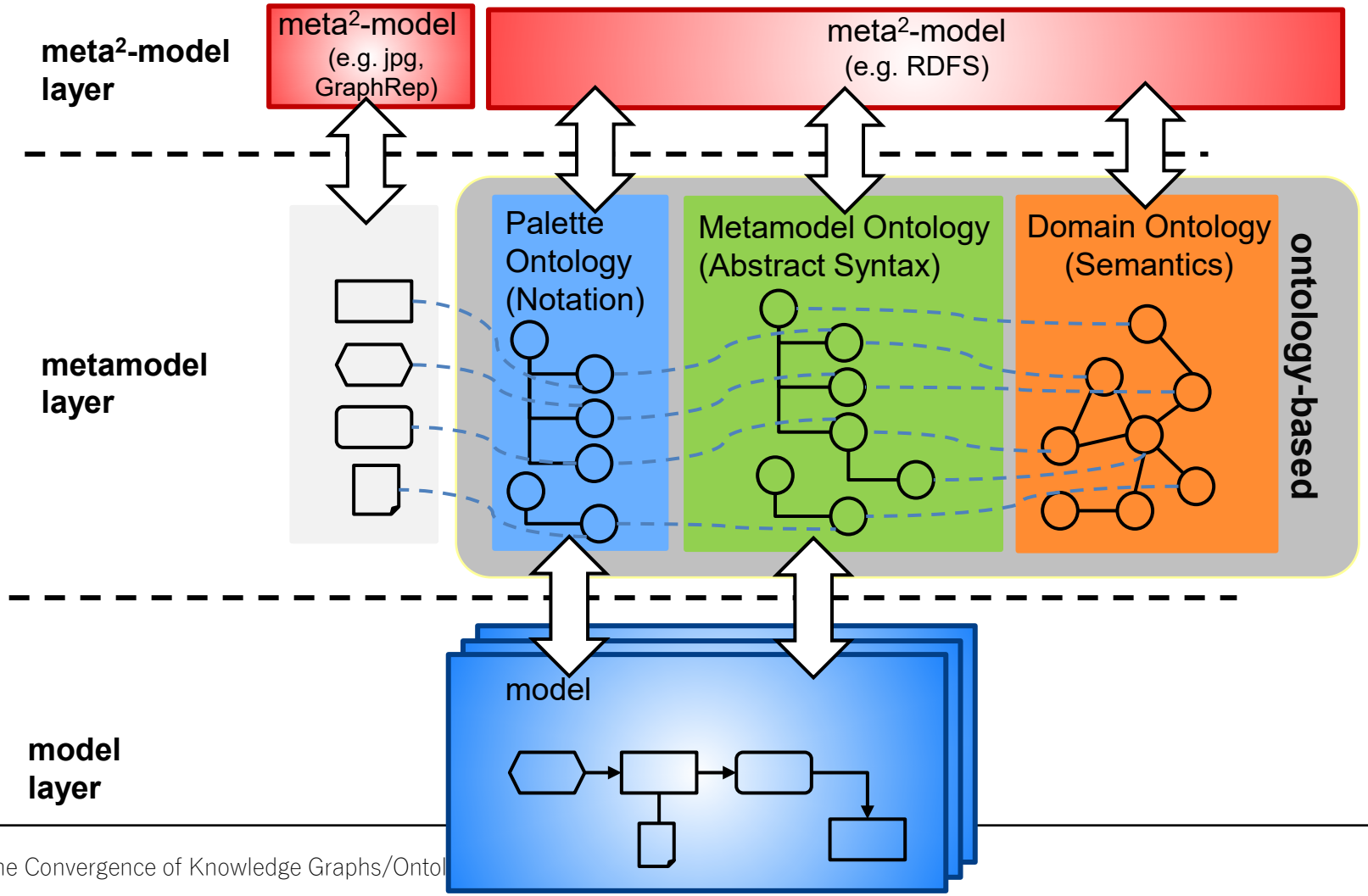
po:paletteElementBackgroundColor
S #c9dcf5

po:paletteElementIconPosition
S top-left

po:paletteElementIconURL
S Thumbnail_Task.png

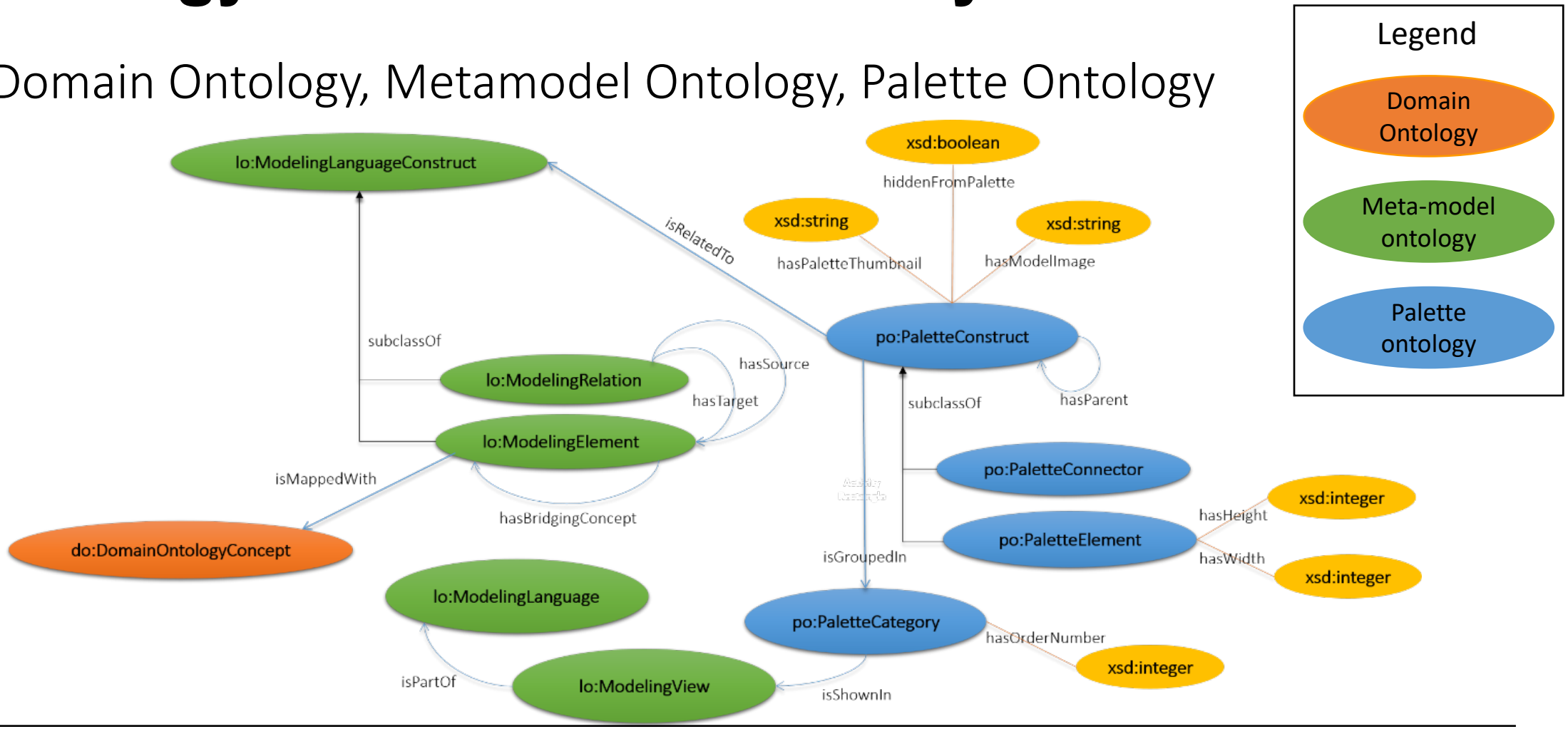


Ontology-based Metamodeling (3): Ontologies for Palette



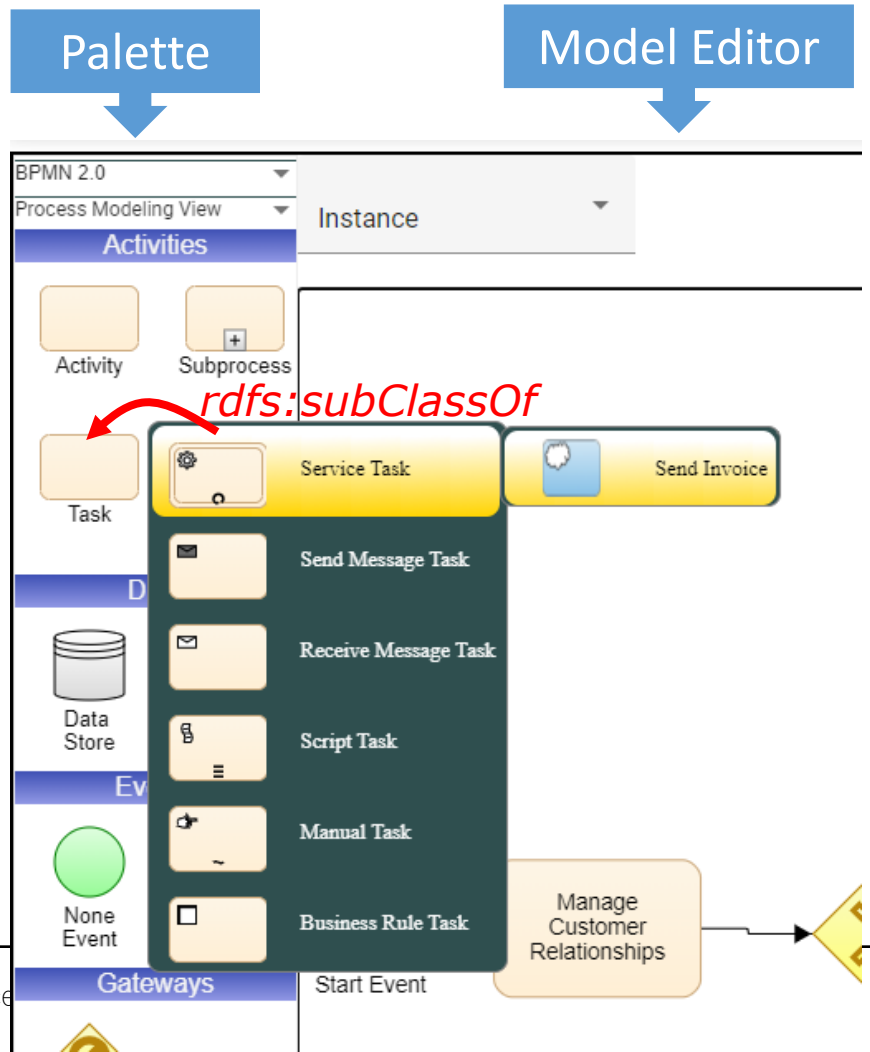
Ontology-based Metamodel Layer

– Domain Ontology, Metamodel Ontology, Palette Ontology

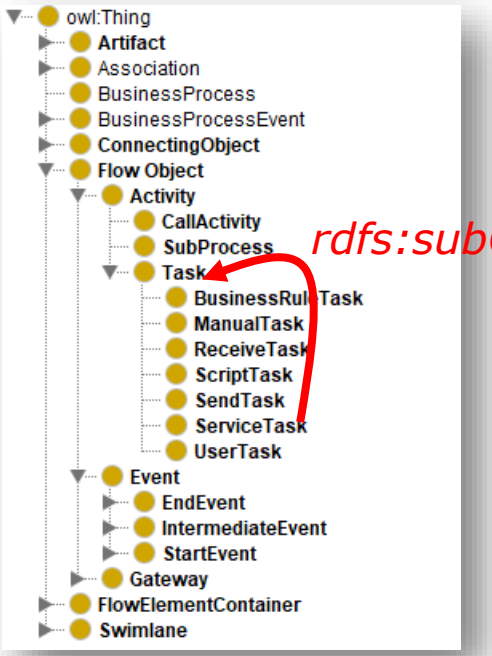


Ontology-Based Modeling in AOAME

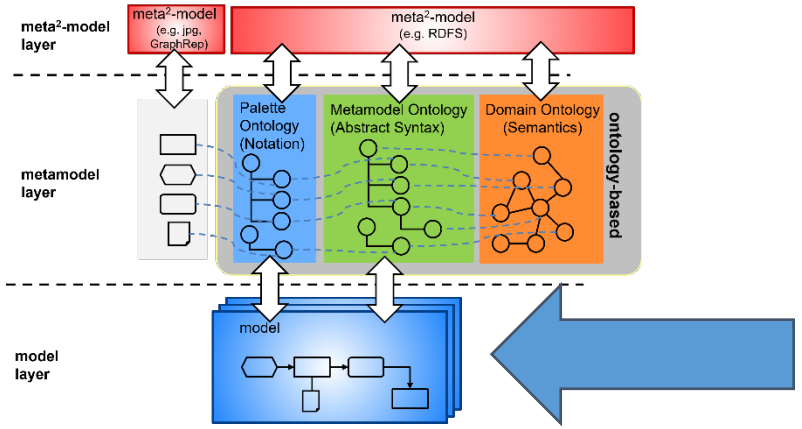
Agile and Ontology-Aided (Meta) Modelling Environment



BPMN
Meta-model Ontology

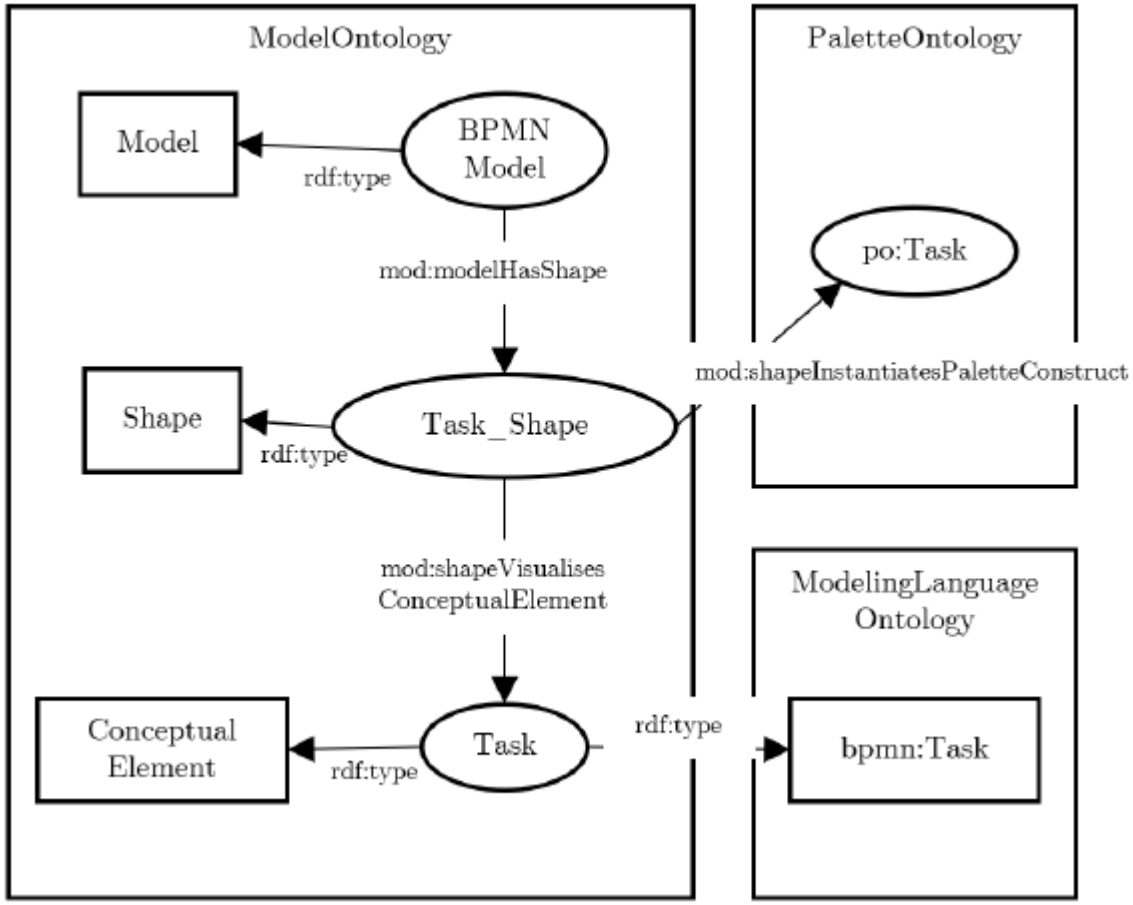


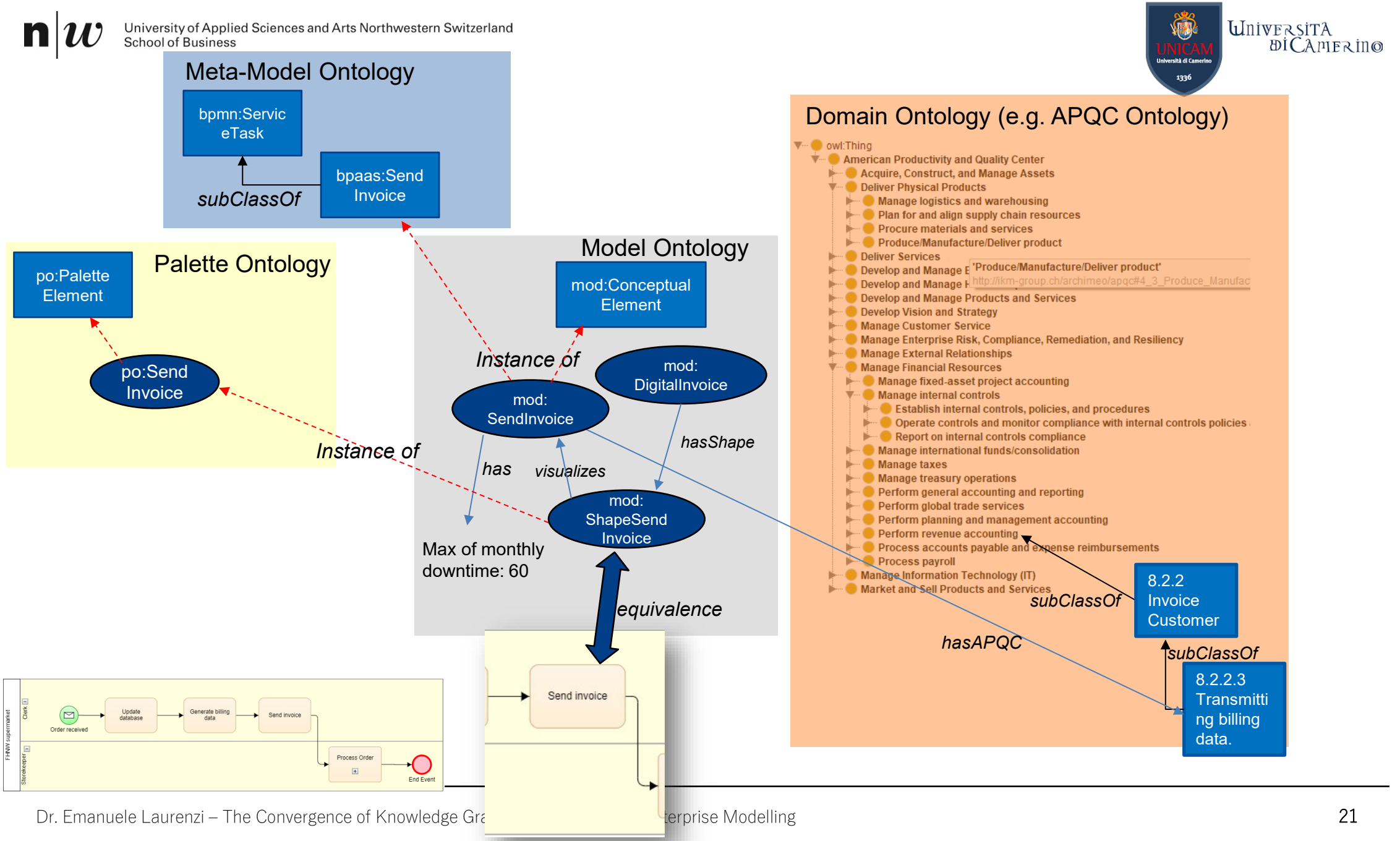
Representing Conceptual Models in Ontology-based Metamodelling



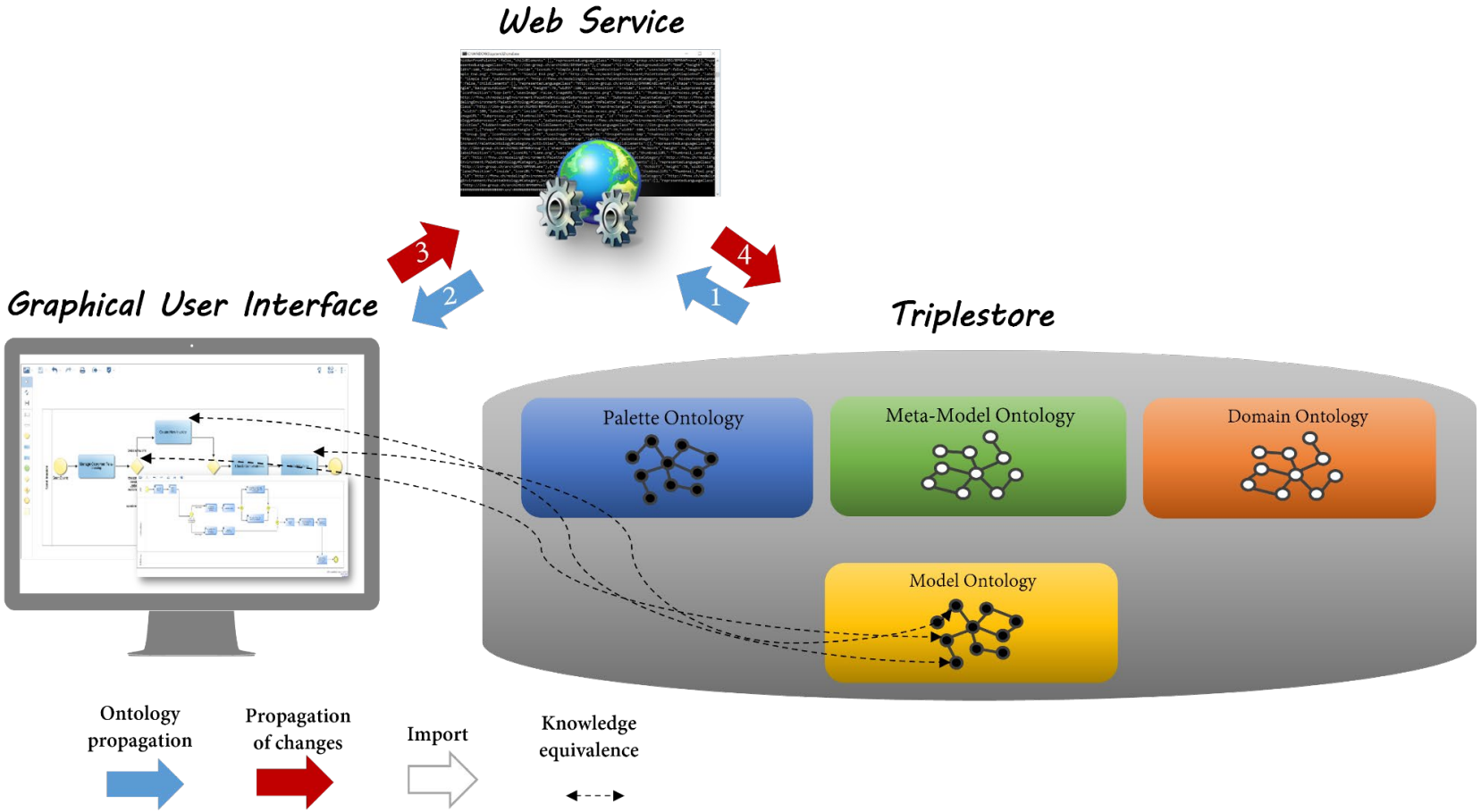
Model Ontology

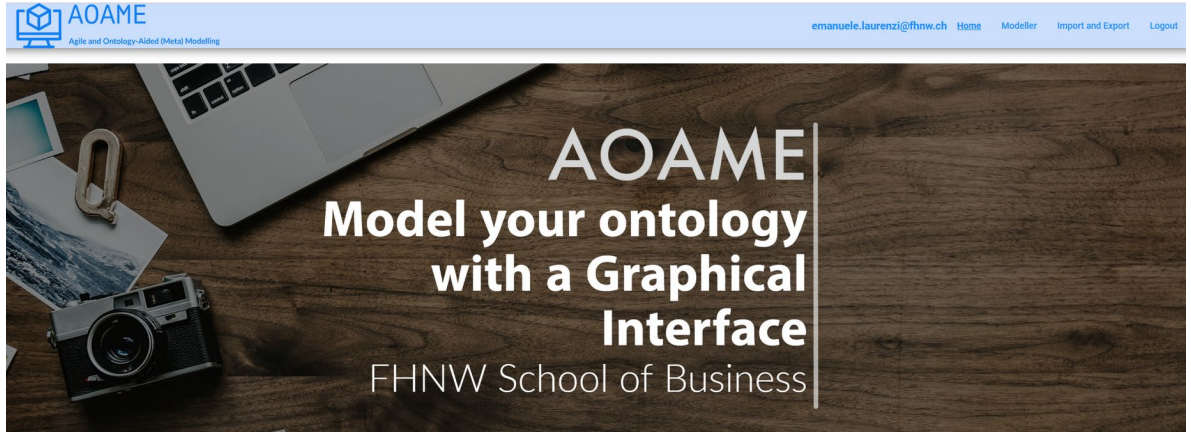
- Models have several elements, named shape
- Each shape visualizes a modeling element
- Each modeling element is related to a meta model construct




Meta-Model Ontology
bpmn:ServiceTask
bpaas:SendInvoice
subClassOf
Palette Ontology
po:PaletteElement
po:SendInvoice
Instance of
Model Ontology
mod:ConceptualElement
Instance of
mod:DigitalInvoice
mod:SendInvoice
hasShape
has
visualizes
mod:ShapeSendInvoice
Max of monthly downtime: 60
equivalence
Domain Ontology (e.g. APQC Ontology)
owl:Thing
American Productivity and Quality Center
Acquire, Construct, and Manage Assets
Deliver Physical Products
Manage logistics and warehousing
Plan for and align supply chain resources
Procure materials and services
Produce/Manufacture/Deliver product
Deliver Services
'Produce/Manufacture/Deliver product'
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
Establish internal controls, policies, and procedures
Operate controls and monitor compliance with internal controls policies
Report on internal controls compliance
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
Process accounts payable and expense reimbursements
Process payroll
Manage Information Technology (IT)
Market and Sell Products and Services
subClassOf
8.2.2 Invoice Customer
hasAPQC
subClassOf
8.2.2.3 Transmitting billing data.
Order received
Update database
Generate billing data
Send invoice
Process Order
End Event
Send invoice

High Level Architecture of AOAME





Demo on AOAME

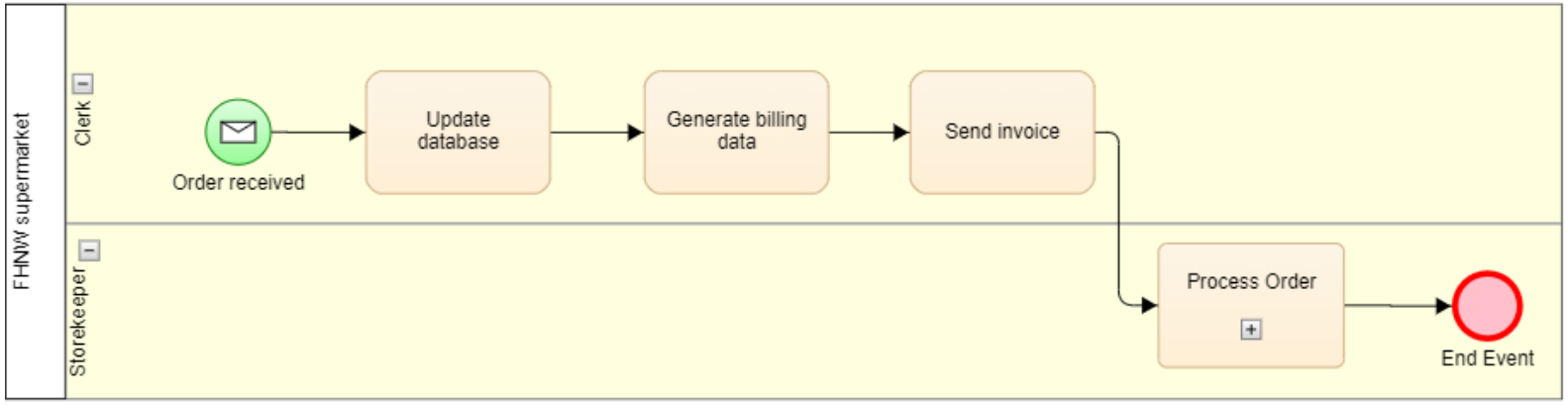
Agile and **O**ntology-b**A**sed (Meta)**M**odelling **E**nvironment.

Use AOAME locally.

Local installation: https://github.com/BPaaSModelling/AOAME_Local_Installation

Exercise in AOAME /2

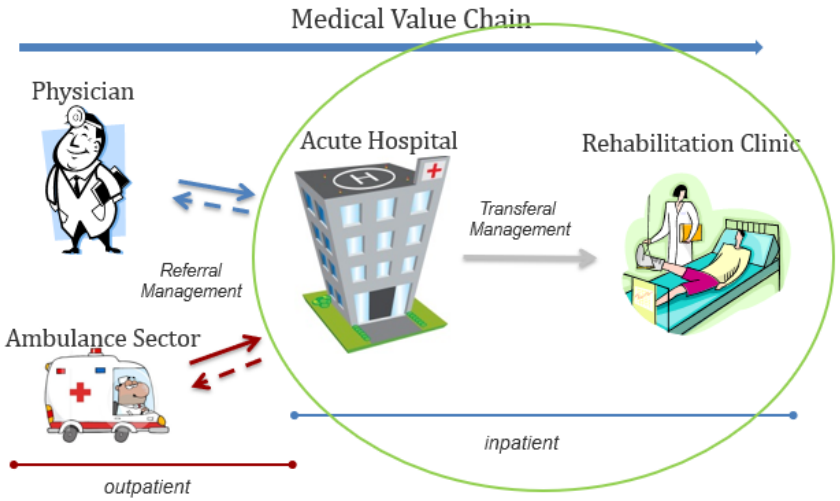
- Create a model called “Order processing”
- Create the below process model
- Prove that the reflecting ontology is created through SPARQL queries.
  - URL to triplestore: https://aoame-fuseki.azurewebsites.net/dataset.html
    - You will create and test the SPARQL here.
  - Follow the walkthrough on ontology-based modelling in AOAME.



Agile Meta-Modelling

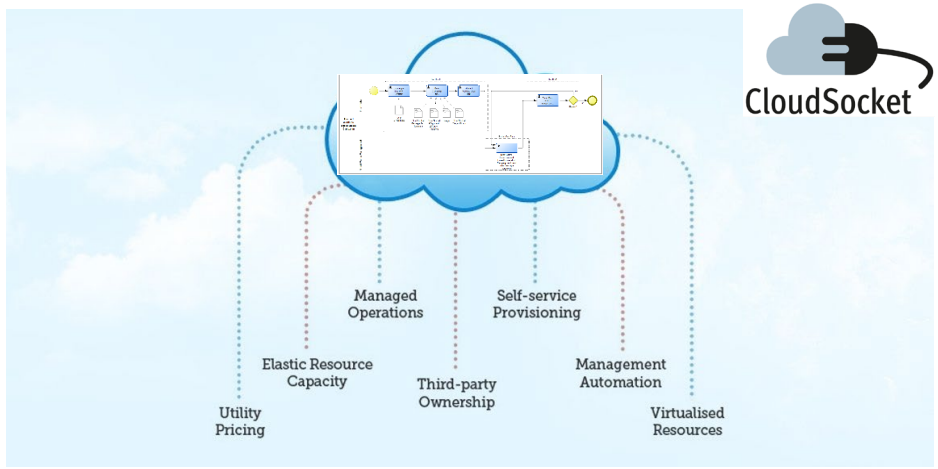
Need for Domain-Specific Modelling Languages (DSMLs) /1

Models are built for a specific purpose



Patient Transferal Management

Purpose of the models:
provide all the relevant concepts (events, activities and decisions).

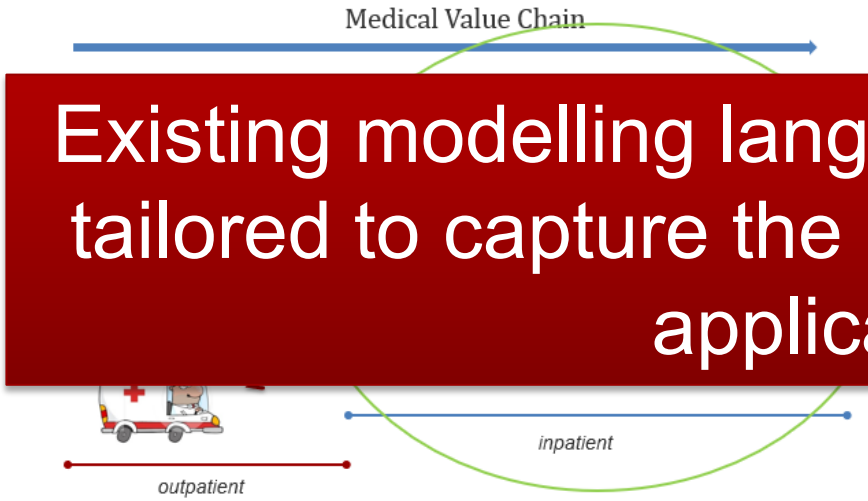


Business Process as a Service (BPaaS)

Purpose of the models:
describe the business process requirements and IT specifications of cloud offerings.

Need for Domain-Specific Modelling Languages (DSMLs) /2

Models are built for a specific purpose



Patient Transferal Management

Purpose of the models:
provide all the relevant concepts (events, activities and decisions).

Business Process as a Service (BPaaS)

Purpose of the models:
describe the business process requirements and IT specifications of cloud offerings.

Existing modelling languages did not offer constructs tailored to capture the representational needs of the application domain

Need for Domain-Specific Modelling Languages (DSMLs)/3

Models are built for a specific purpose

Existing modelling languages did not offer constructs tailored to capture the representational needs of the application domain

Adapt existing modelling languages to offer constructs tailored to capture the representational needs of the application domain!

Patient Transferal Management

Purpose of the models:
provide all the relevant concepts (events, activities and decisions).

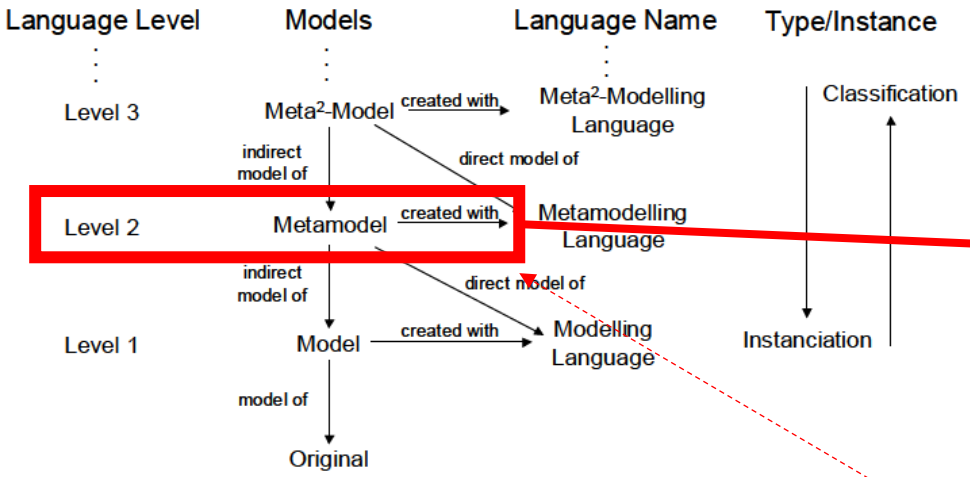
(BPaaS)

Purpose of the models:
describe the business process requirements and IT specifications of cloud offerings.

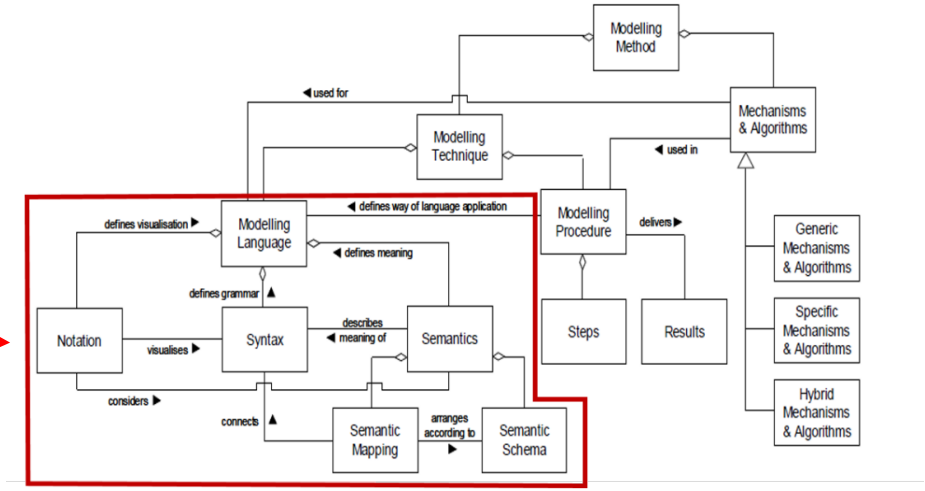
Benefits of DSMLs created through adaptations

- **Benefits of considering existing Modeling Languges:**
  - Reuse of established experience and lessons learned,
  - Syntax and semantics can be borrowed,
  - Fosters reusability within the modeling community or across projects.
- **Benefits of Domain-Specific Modelling Languages (DSML):**
  - High expressiveness of concepts, conciseness -> High productivity of modelling
    - Concepts, relations and their notations are tailored to a specific problem domain (domain-specificity).
  - Better understanding of models -> Support in decision-making
    - Graphical notations familiar to domain experts.
  - Designing models in a meaningful and less error-prone way -> High quality of models
    - Higher degree of semantics in the modelling language
      - Modeling constructs already constrained

Domain-Specific Adaptations of Modelling Languages = Meta-modelling



(Strahringer, 1996)



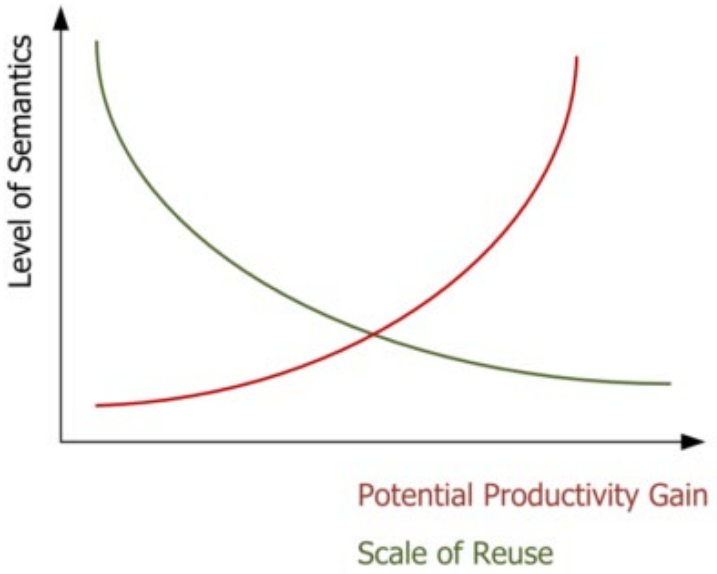
(Karagiannis & Kühn, 2002)

Domain-specific adaptations of modelling languages

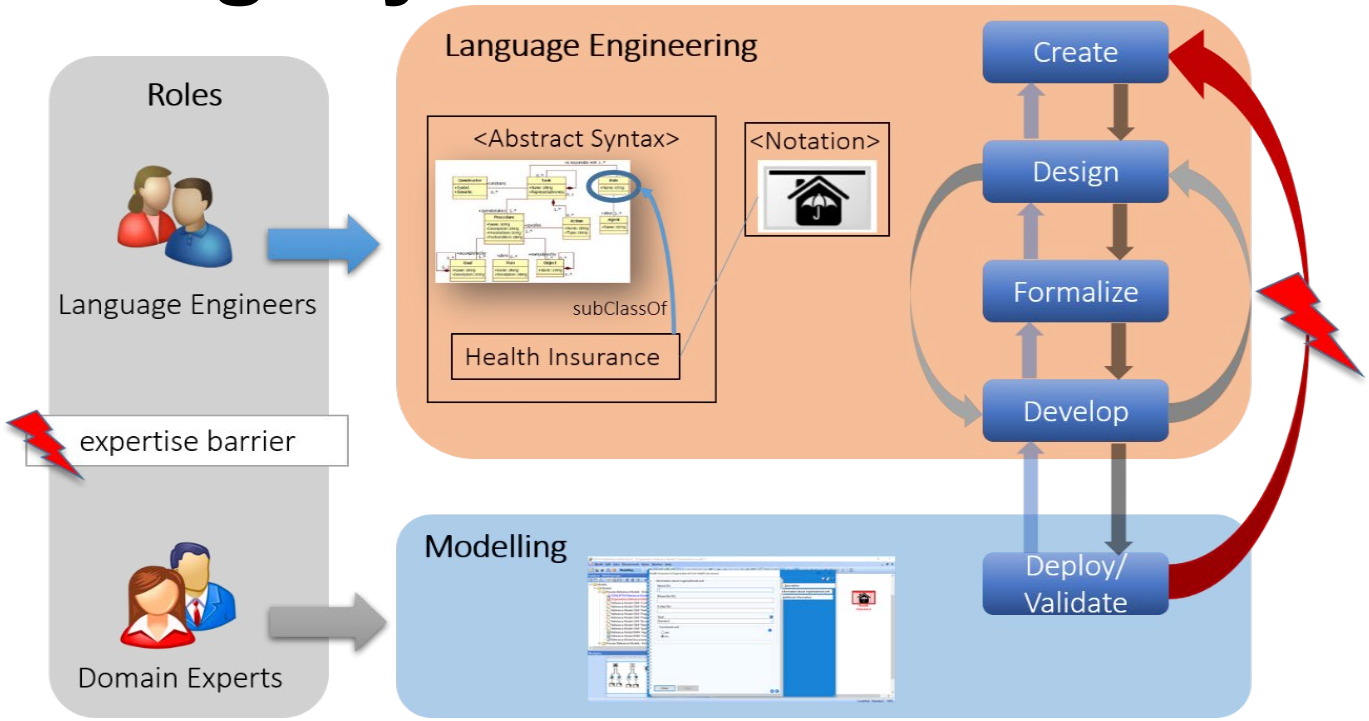
The creation of DSMLs is challenging

- Both domain and language engineering expertise are required.

- Difficult to determine an appropriate domain-specificity of the language.
  - Productivity vs. Reusability



Lack of agility in current meta-modelling tools

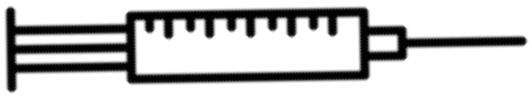


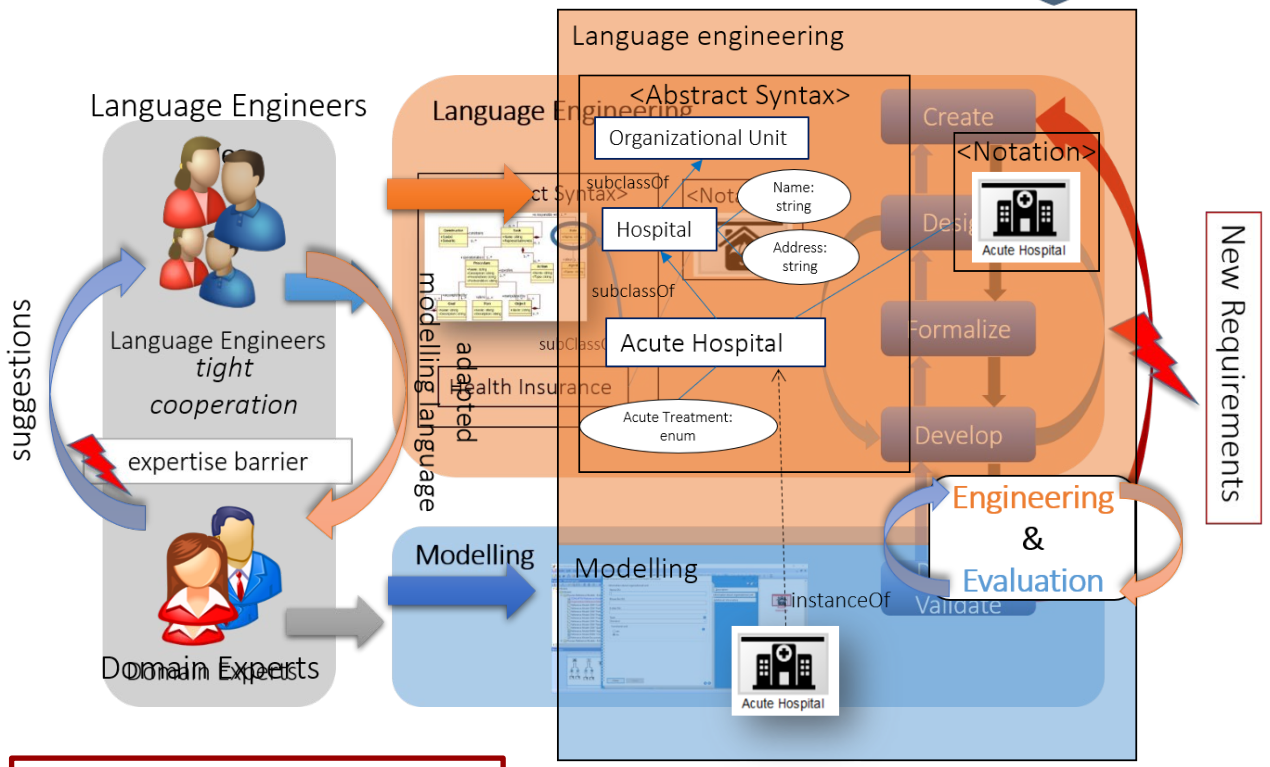
1. Separation of the two key roles
   - Threat to the quality of the DSML as it requires the joint and cooperative effort of both experts!
2. Sequential accommodation of new requirements
   - Time-consuming!!!

Challenge 1: Promote the tight cooperation from an early stage

Challenge 2: avoid sequential engineering phases

Inject Agility





- Interleave of language engineering, modelling and evaluation activities within a joint component
- On-the-fly domain-specific adaptations

Challenge 1: Promote the tight cooperation from an early stage

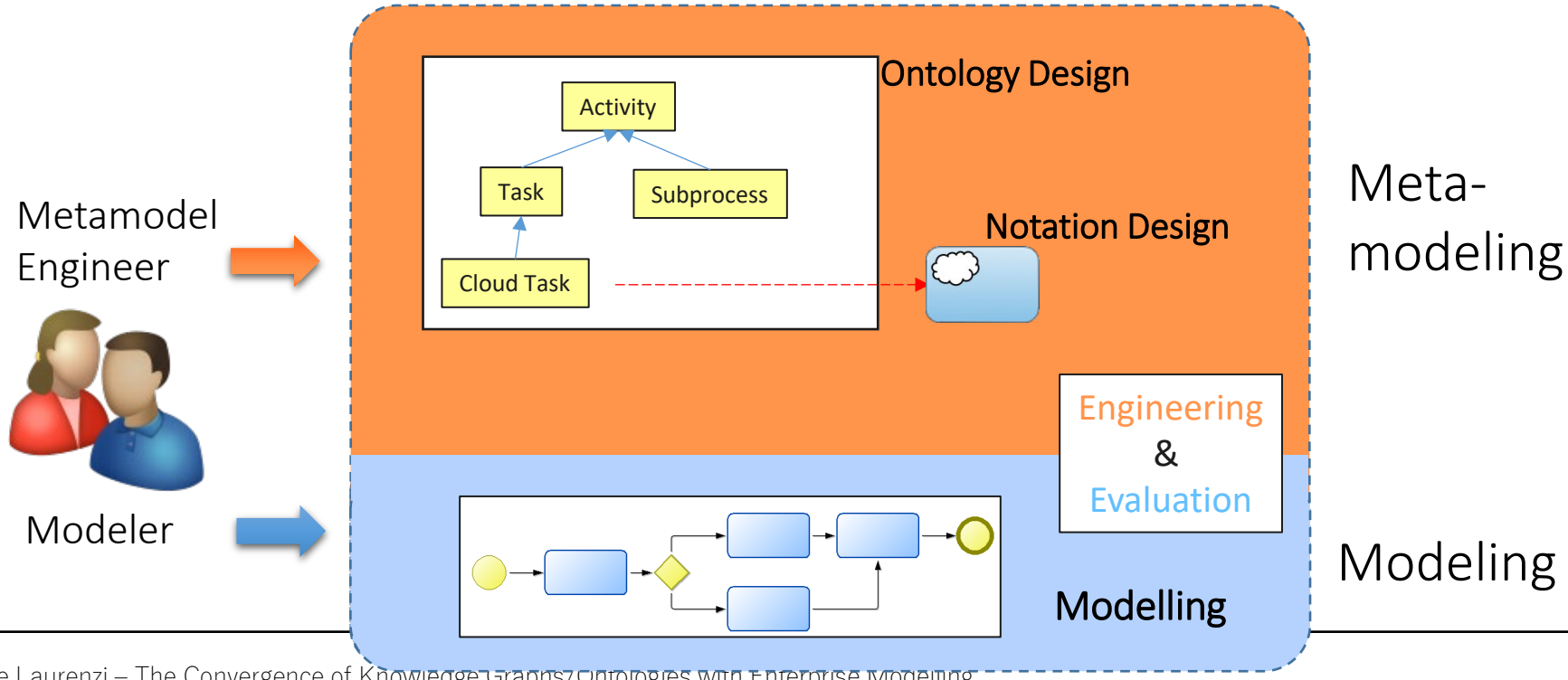
Challenge 2: avoid sequential engineering phases

Drawbacks:

- Ambiguity of new modelling constructs
- Limited automation

Integration Modeling and Metamodeling in a Single Environment

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



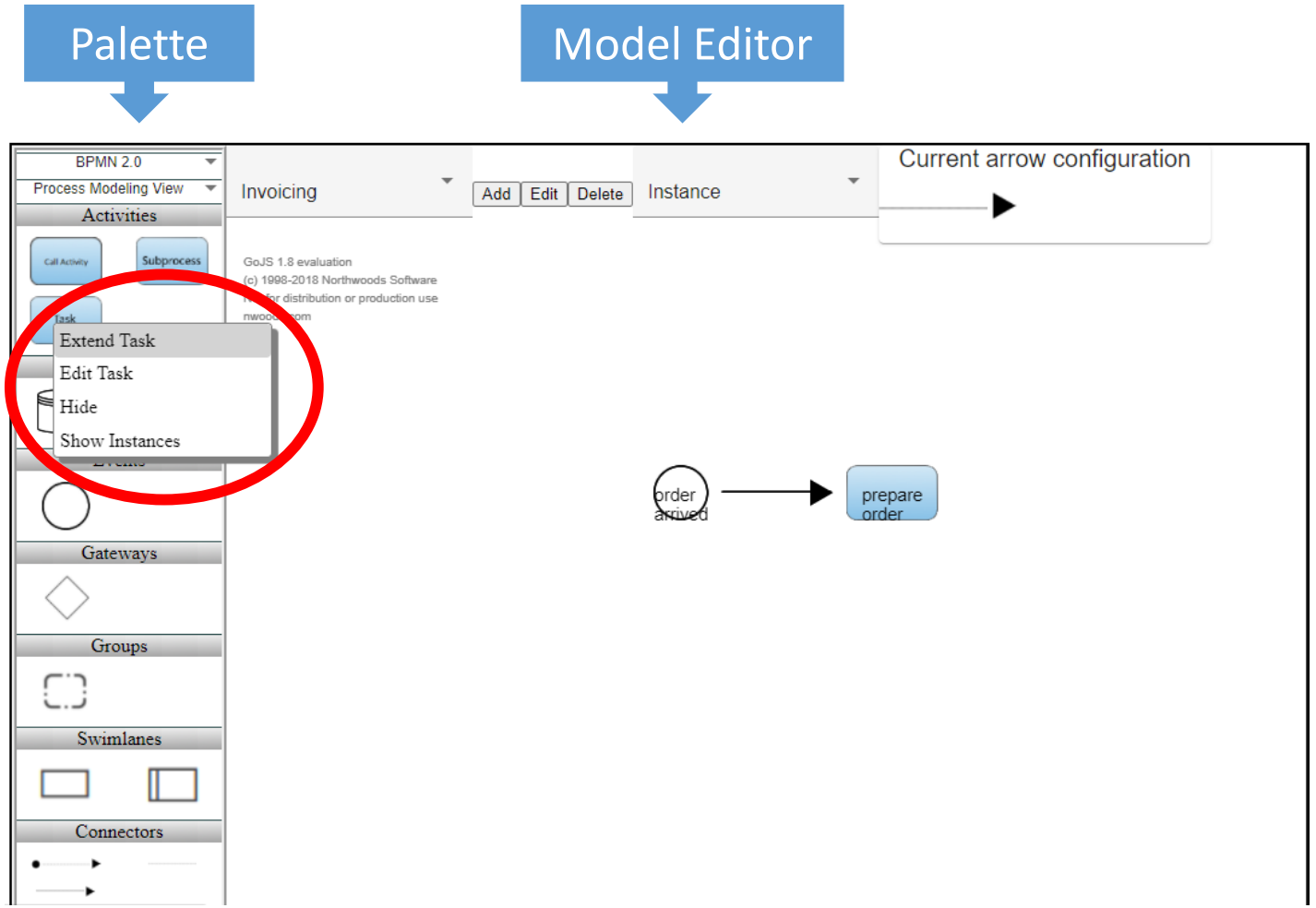
Advantages

- The consistency between the two knowledge representations (the graphical meta-models, models and ontologies) is kept while performing domain-specific adaptations of modelling languages.
- The graphical knowledge representation is both human- and machine-interpretable.
- Ontology-based metamodeling can be used to create and maintain ontologies without ontology expertise.

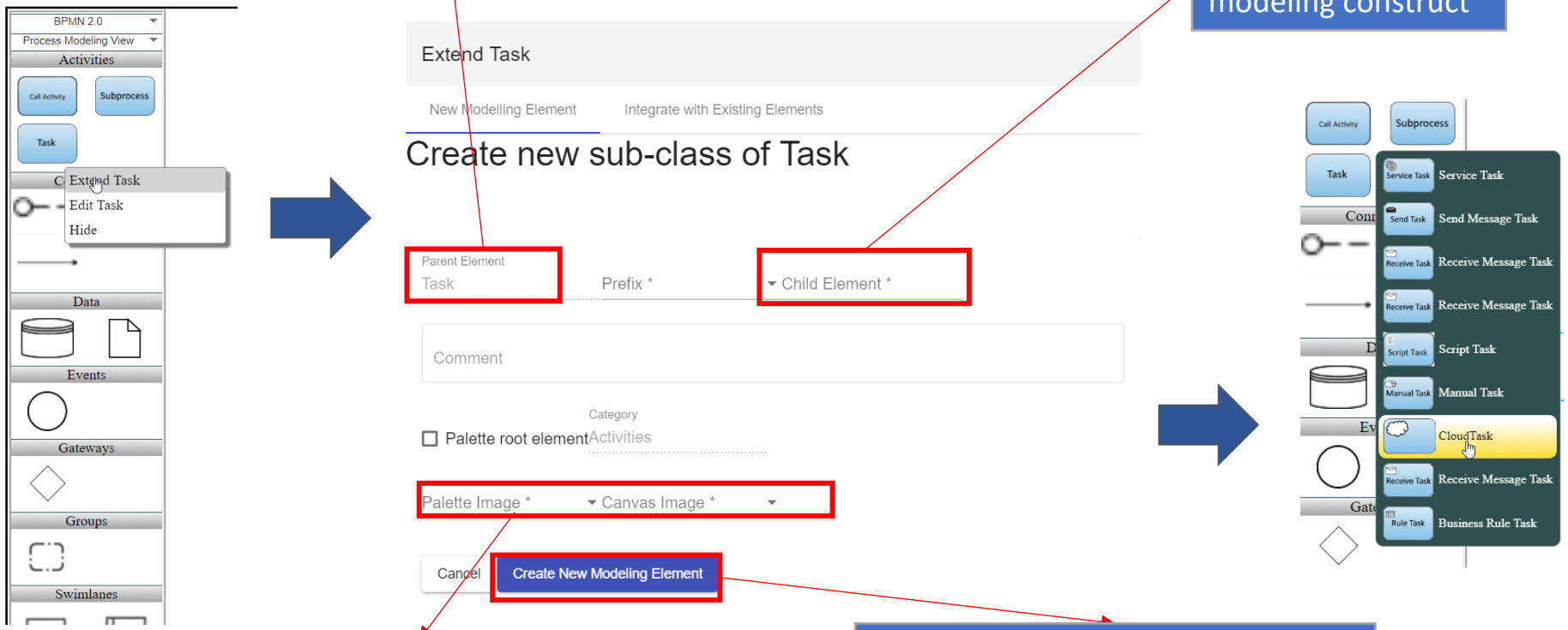
Exercise 2

– Open the file named “Walkthrough and query creation for agile meta-modelling in AOAME” and follow the instructions.

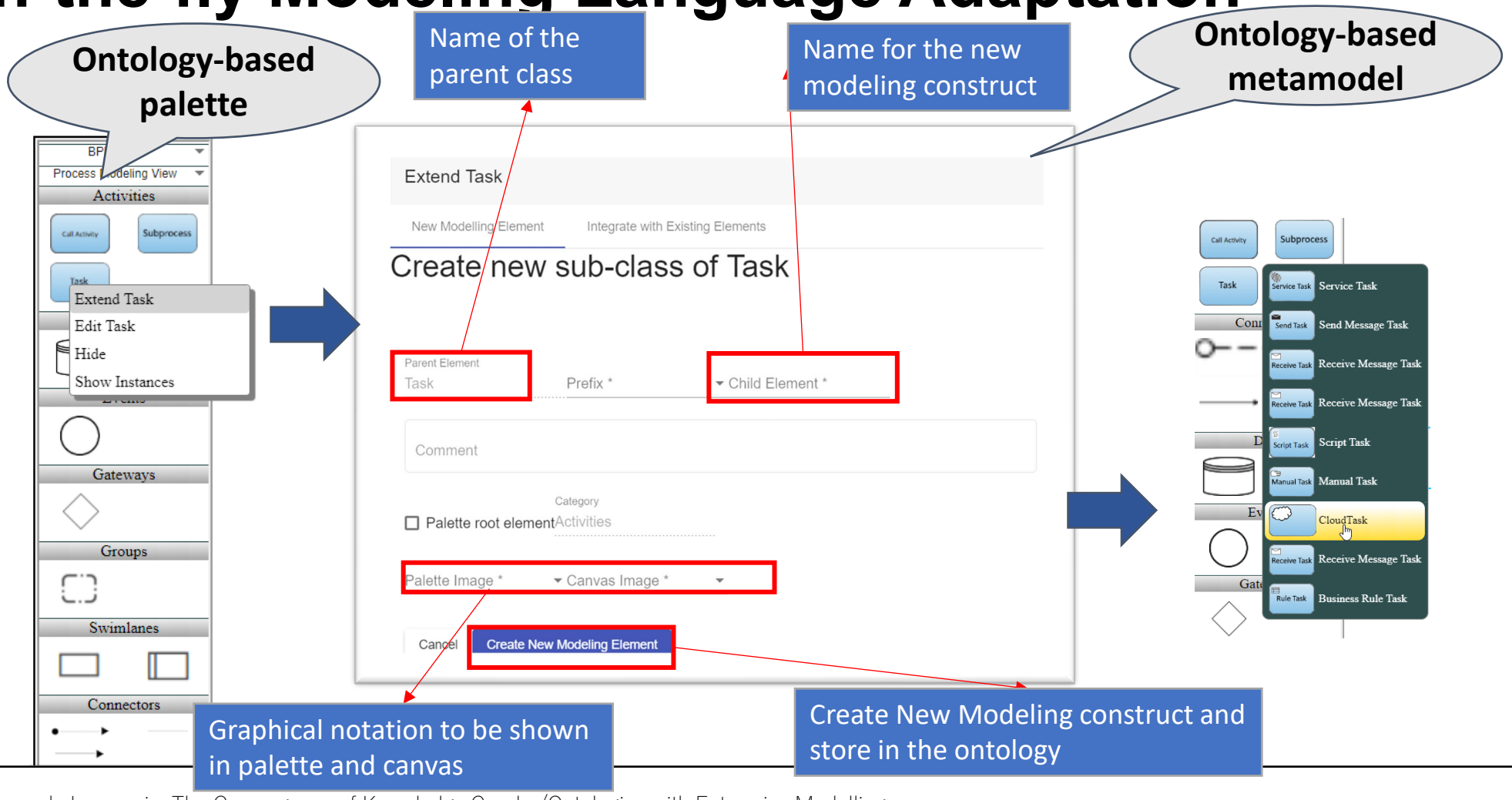
Extending AOAME Modeling Languages – on the fly



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

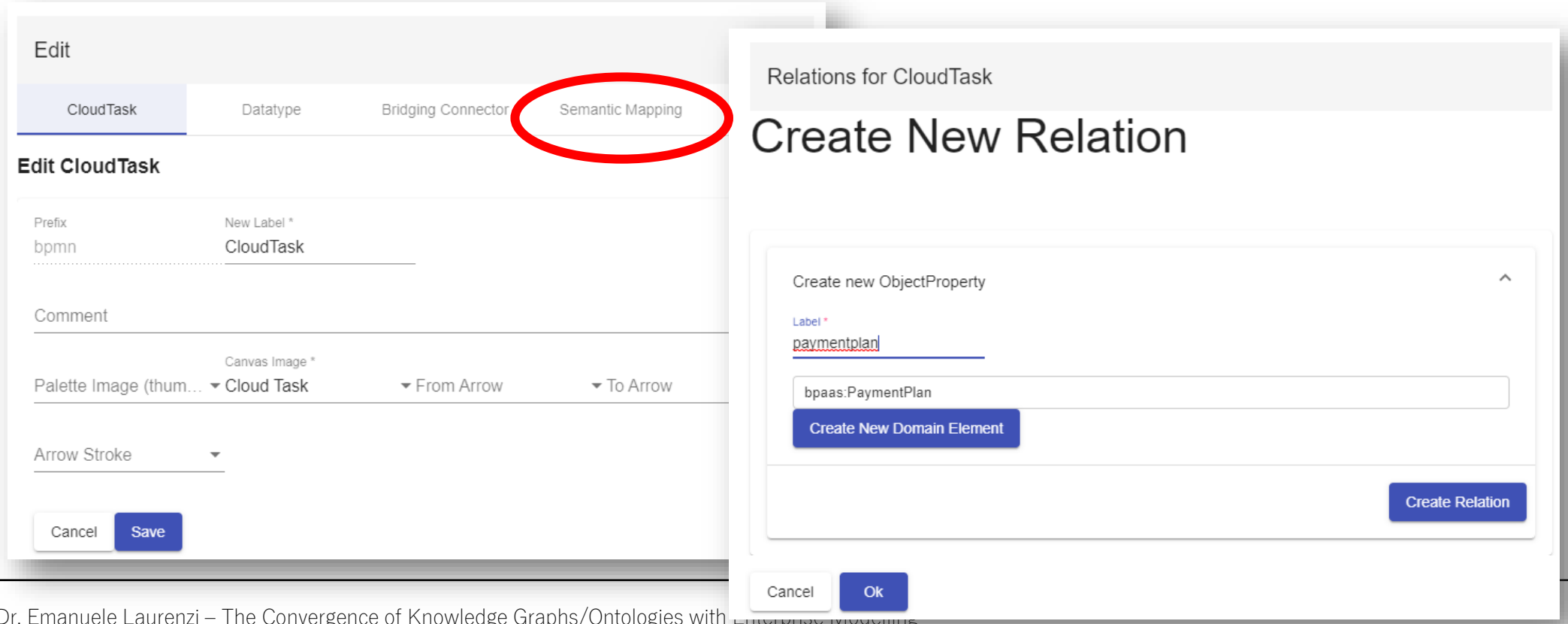


Integration of Meta-modeling and Modeling: On-the-fly Modeling Language Adaptation



Semantic Alignment in AOAME

– With Semantic Mapping modeling elements can be connected to domain ontology concepts.



Exercise 3

– Create the query that retrieve the Cloud Services for the specified business process. Note that we are looking for Cloud Services that have a downtime less than 60 min.

AOAME:
Agile and Ontology-bAsed Modeling Environment

- AOAME is a prototypical implementation for Agile and Ontology-based Meta-modelling created from Emanuele Laurenzi's PhD, which has been supervised by Prof. Dr. Knut Hinkelmann.
- Implementation of the current version (chronologically) by:
  - Emanuele Laurenzi
  - Stefano Izzo
  - Charuta Pande
  - Devid Montecchiari
  - Egemen Kaba
  - Marco Di Ianni
  - Jan Eich
  - Victor Hargrave
  - Kyrylo Buga
- Several master's thesis and PhD built and continue to build on AOAME.

References

- Laurenzi, E. (2024). An Agile and Ontology-based Meta-Modelling Approach for the Design and Maintenance of Enterprise Knowledge Graph Schemas. Special issue on Enterprise Modeling and Knowledge Graph in Enterprise Modelling and Information Systems Architectures (EMISAJ). International Journal of Conceptual Modeling https://doi.org/10.18417/emisa.19.6
- Laurenzi E., Hinkelmann K., Montecchiari D., Goel M. (2021) Agile Visualization in Design Thinking. In: Dornberger R. (eds) New Trends in Business Information Systems and Technology. Studies in Systems, Decision and Control, vol 294. Springer, Cham. https://doi.org/10.1007/978-3-030-48332-6_3
- Laurenzi, E. (2020): An Agile and Ontology-Aided Approach for Domain-Specific Adaptations of Modelling Languages. PhD thesis, University of Pretoria. Available at: https://repository.up.ac.za/handle/2263/73419
- Hinkelmann, K.; Laurenzi, E.; Martin, A.; Montecchiari, D.; Spahic, M. and Thönssen, B. (2020). ArchiMEO: A Standardized Enterprise Ontology based on the ArchiMate Conceptual Model. In Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development - MODELSWARD, ISBN 978-989-758-400-8; ISSN 2184-4348, pages 417-424. DOI: 10.5220/0009000204170424
- Laurenzi, E., Hinkelmann, K., & van der Merwe, A. (2018). An Agile and Ontology-Aided Modelling Environment. In R. Buchmann, D. Karagiannis, & M. Kirikova (Eds.), The Practice of Enterprise Modelling. PoEM 2018. (pp. 221–237). Vienna: Springer, Cham. https://doi.org/10.1007/978-3-030-02302-7_14
- Laurenzi, E., Hinkelmann, K., Izzo, S., Reimer, U., Merwe, A.V.D. (2018): Towards an agile and ontology-aided modeling environment for DSML adaptation. In: International Conference on Advanced Information Systems Engineering, pp. 222–234. Springer International Publishing, Cham
- Harel, D., & Rumpe, B. (2004). Meaningful modeling: what's the semantics of 'semantics'? *Computer*, *37*(10), 64–72. https://doi.org/10.1109/MC.2004.172
- Atkinson, C. and Kuhne, T. (2003) "Model-driven development: a metamodeling foundation," in IEEE Software, vol. 20, no. 5, pp. 36-41, doi: 10.1109/MS.2003.1231149.
- Harel, D., & Rumpe, B. (2000). *Modeling Languages: Syntax, Semantics and All That Stuff, Part I: The Basic Stuff*. Weizmann Science Press of Israel. Retrieved from https://dl.acm.org/citation.cfm?id=903627