

Expert Interview

First Part

When: 15/08/2018

Where: Zurich

Expert: Dr. Ragnar Nevries

Role: Senior Consultant

Company: Camunda

Company Website: <https://camunda.com/>

Personal Profile: <https://www.linkedin.com/in/ragnar-nevries/>

In consulting projects:

1. *Did users face difficulties in learning standard modeling languages and using them in Camunda?*

Yes, users generally face difficulties in learning standard languages and using them in Camunda. Business people have more difficulties than IT people as the latter are better with algorithms and logic. However, not all the standard language have the same degree of difficulties. For instance, DMN is the easiest standard to learn from both Business and IT people. CMMN is the hardest standard to learn for Business people as it is logic-driven. There have been situations where CMMN was difficult to IT people too. When teaching CMMN, 1 person out of 10 gets it right, whereas when teaching BPMN, 9 people out of 10 get it right. Thus, we encourage to use BPMN as much as possible. At present there is no a single Camunda customer using CMMN.

- a. *If yes, what are the reasons for that?*

In addition to the previous reasons, semantics of standard languages is very broad.

- b. *How could this issue be overcome?*

We provide training to teach how to use modeling languages. Usually it's a 2-full days workshop, where attendees are mainly guided through best practices and how not to model.

Additionally, simplifying the language could be an option. Namely, reducing unnecessary modeling constructs. For instance, in Camunda the parallel start event is not implemented.

In both research and industry, there is the recent trend of adapting standards (or existing) modeling languages to address a specific domain. In result a domain-specific modeling language is developed (DSML). This has the following benefits:

- It decreases the error prone while modeling. This is due the injection of semantics (i.e. abstract syntax and constraints) in the metamodel, which decreases the degree of freedom of modelers.
- It enhances understanding of models by domain experts. This is due to the graphical notations targeting a specific domain.

Developing a DSML through domain-specific adaptation of existing modeling languages has the following benefits:

- Modeling expert-friendly. This is due to the reference to already existing modeling standards.
- Total or partial reusability of the resulting language (i.e. DSML) within the modeling community. This is also due to the reference to already existing modeling standards.
- It fosters the quality of the modeling language. Established experience and lessons learned from existing modeling languages can be taken into account. Additionally, semantics and syntax can be borrowed.

2. *Could a DSML address one of the issues/problems identified in question 1? Are there more problems that can be addressed?*

Yes, a DSML can definitely help addressing the issues above identified. It sounds very useful to have a tool with the possibility to add constraints in the meta-model. Thus, restrictions on the modeling language make sense. Also specifying modeling constructs with new modeling constructs targeting a specific domain is needed. However, I do not see the usefulness of adding new modeling elements from scratch.

3. *Have there been situations, where the modeling languages were not sufficient and where an adaptation could have made sense? Namely, adapting language constructs to fit a more specific domain?*

Yes, from time to time it would have been needed to adapt modeling languages to a targeted domain. For instance, looking back to my past projects, it would have been of great help to be able to specify a system type, e.g. ERP, salesforce etc.

a. *If not, Why? -*

- b. *Could a functionality for an on-the-fly customization of modeling constructs be useful in projects with domain-specific target?*

Yes, especially in domains like IT-system, Pharma and Finance.

4. *For the situations, where the language was not adequate, what kind of modifications on the modeling language would have helped? e.g. creating/deleting/update a modeling construct (i.e. class, attribute and relation on the metamodel level)*

All modifications from operator 1 to 10 make sense.

5. *Could you provide at least a use case where domain-modeling adaptation would have made sense?*

Yes, I will provide a BPMN model built in a project where an adaptation occurred. Data will be anonymized.

Second Part

1. Do the operators derived in the paper (see below) make sense for you? **Yes, they do. However, we never faced the need of operator 11, i.e. no need to distinguish between an instance and a class as all the modeling elements in the model are considered as classes, which are then instantiated by the workflow management system.**
2. Do you suggest other operators/actions on the language for adaptation purpose? **No.**

Operators:

Operator 1: Create sub-class. This operator is applied on modeling elements and modeling relations to create new modeling elements and new modeling relations. This operator is also applied to integrate modeling elements (classes) from different modeling languages. For example, the operator would allow to connect "Discretionary Task" from CMMN as a subclass of the "User Task" from BPMN.

Operator 2: Delete sub-class. This operator is applied on modeling elements and modeling relations to remove unneeded modeling elements and modeling relations from the modeling language.

Operator 3: Create relation (object properties). This operator connects modeling elements and modeling relations to the related Domain Ontology concept.

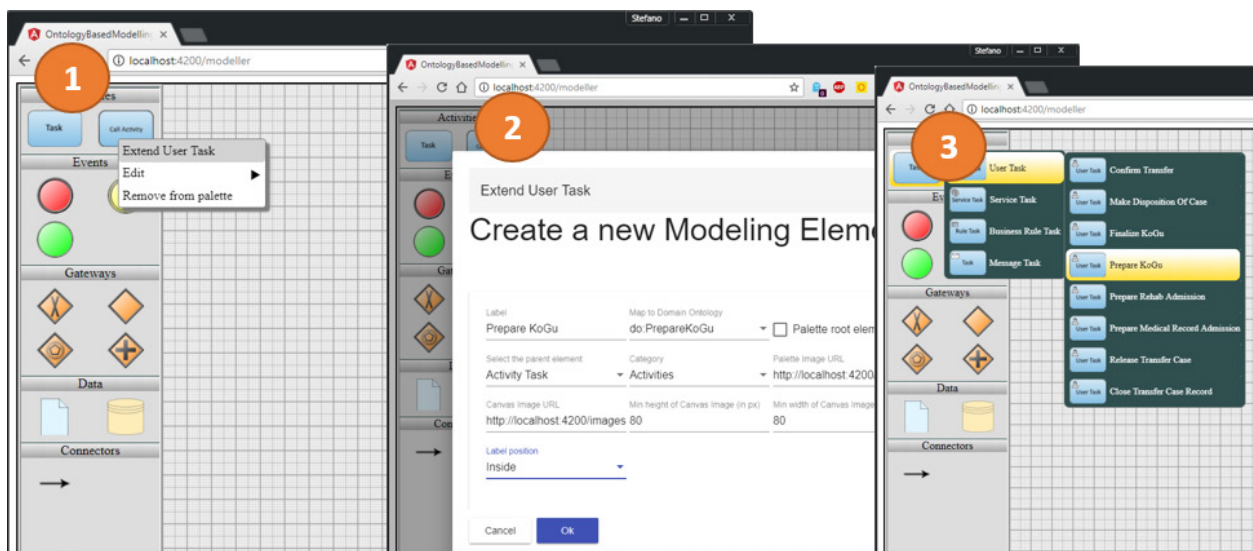
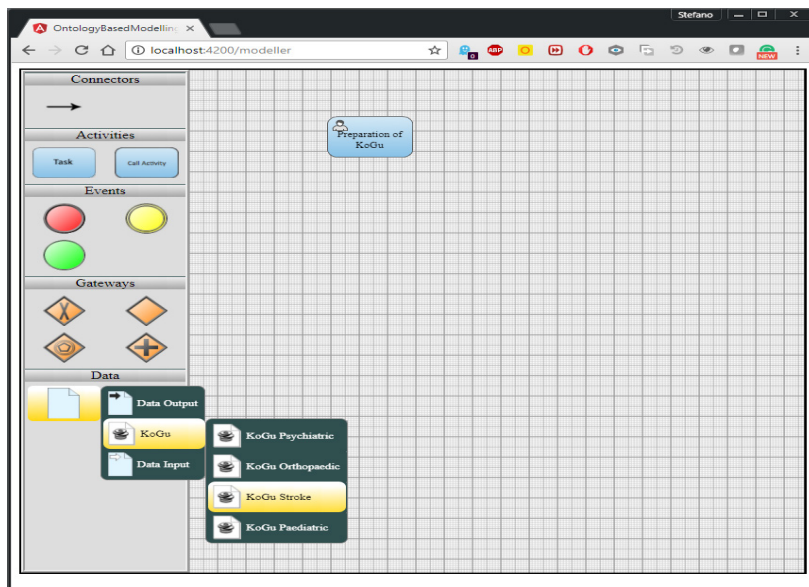


Figure 2. Operator from 1 to 3

Figure 1. Operator 1 applied on "Data Object"



Operator 4: Update relation (object properties). This operator is applied on as it allows updating existing connections between modeling elements/relations and the related Domain Ontology concepts.

Operator 5: Delete relation (object properties). This operator allows deleting existing connections between modeling elements/relations and the related Domain Ontology concepts.

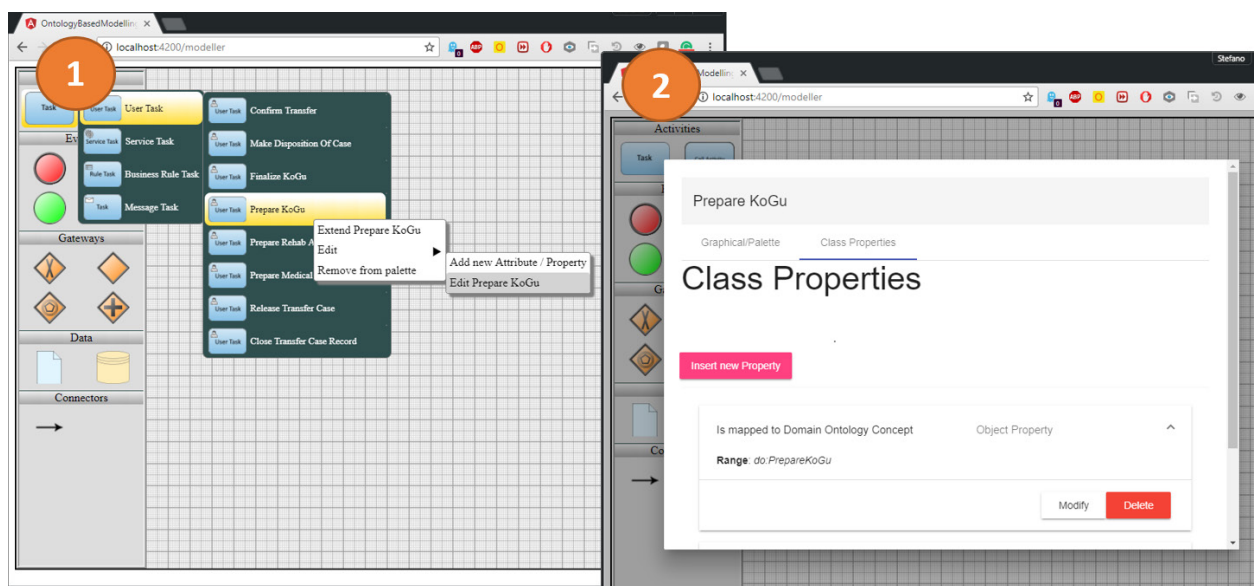
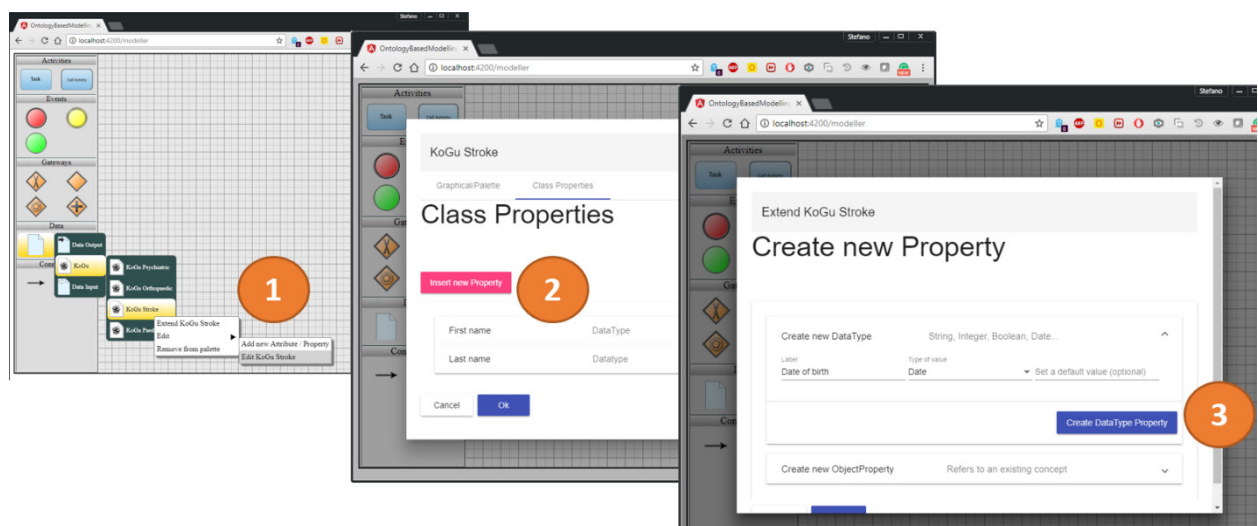


Figure 3. Operators 4 and 5

Operator 6: Create attribute (datatype properties). This operator allows adding new attributes to modeling elements and modeling relations.

Figure 4. Operator 6 (and 3)



Operator 7: Update attribute (datatype properties). This operator is allows updating existing attributes.

Operator 8: Delete attribute (datatype properties). This operator is allows deleting existing attributes.

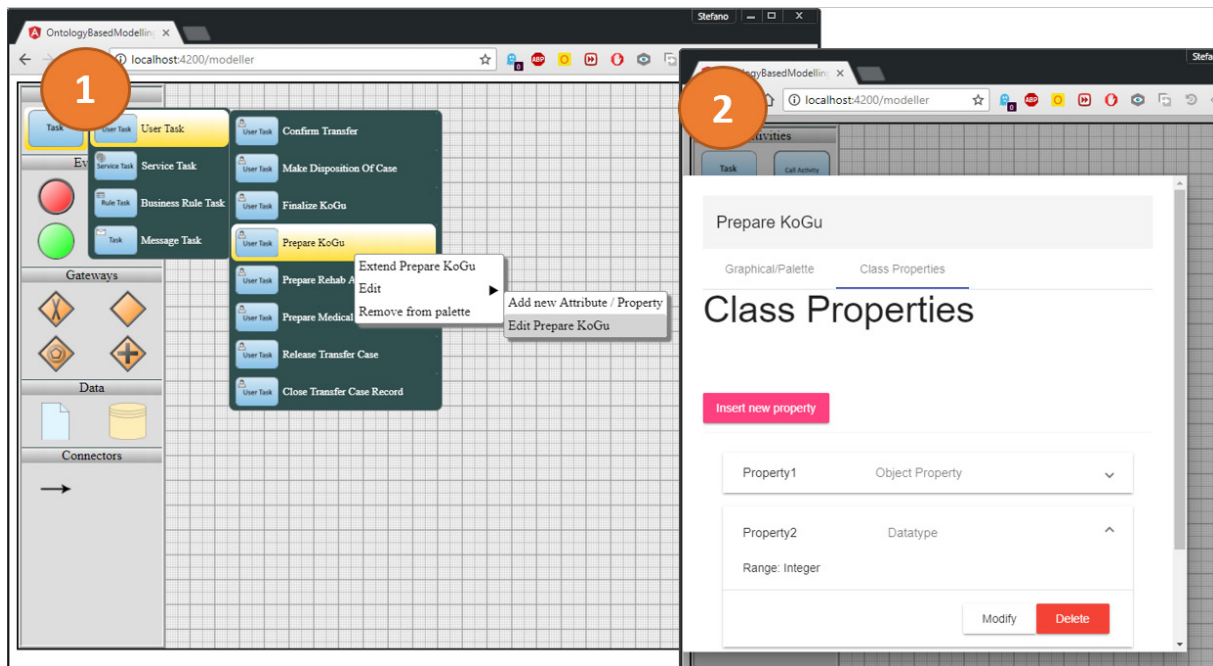


Figure 5. Operatros 7 and 8

Operator 9: Assign attribute type and attribute value. This operator allows assigning value types String, Integer, Boolean to attributes as well as concrete values to attributes of modeling elements.

Operator 10: Update attribute types and attribute values. This operator allows updating types and/or values that are assigned to attributes of modeling elements.

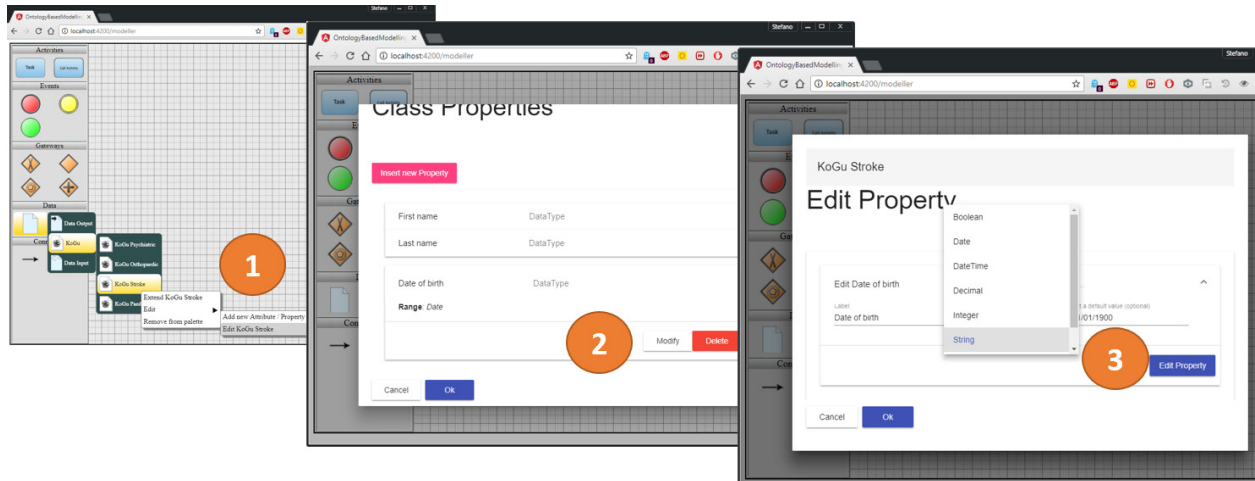


Figure 6. Operator 9 and 10

Operator 11: Enable representation level. This operator allows specifying whether a modeling element is a type or instance. Hence, the modeling environment can model instances as well as classes. Since an class can be an instance of another class, the modeling environment enables to distinguish between different levels of abstractions and not restricted to the class-instance dichotomy of description logics representation.

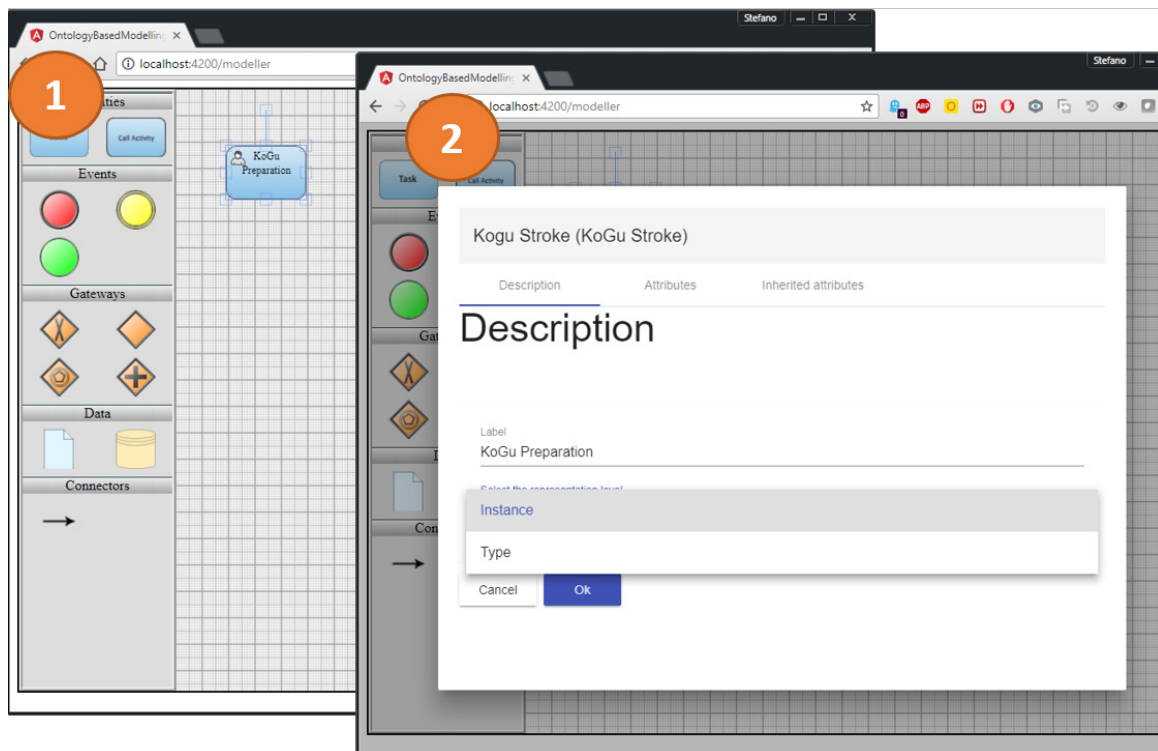


Figure 7. Operator 11

Figure 8 shows an excerpt of the ontology that is behind the modeling environment. It includes elements that belong to the architecture of the ontology-aided modeling environment, i.e. all elements with the “lo” prefix. Elements with the prefix “bpmn” belong to the modeling standard BPMN whereas “dslm4ptm” belong to elements of the created DSML that covers the patient transferal management domain. Finally, the element with the prefix “do” reflects the root concept of the domain ontology, which provides the (language-independent) semantics to both the modeling elements and modeling relations.

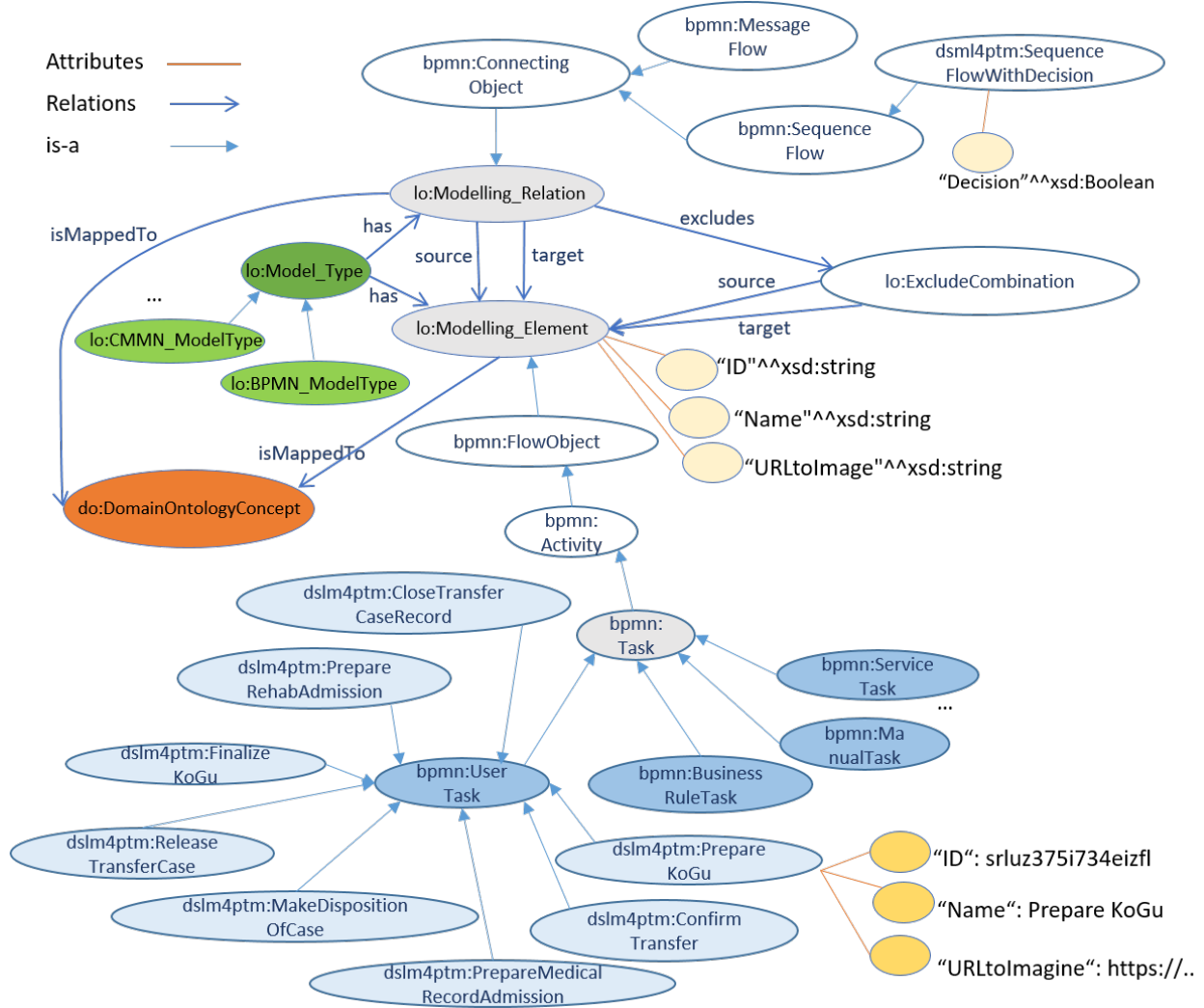


Figure 8. Ontology Excerpt

Figure 9 shows the overall new approach agile and ontology-aided approach.

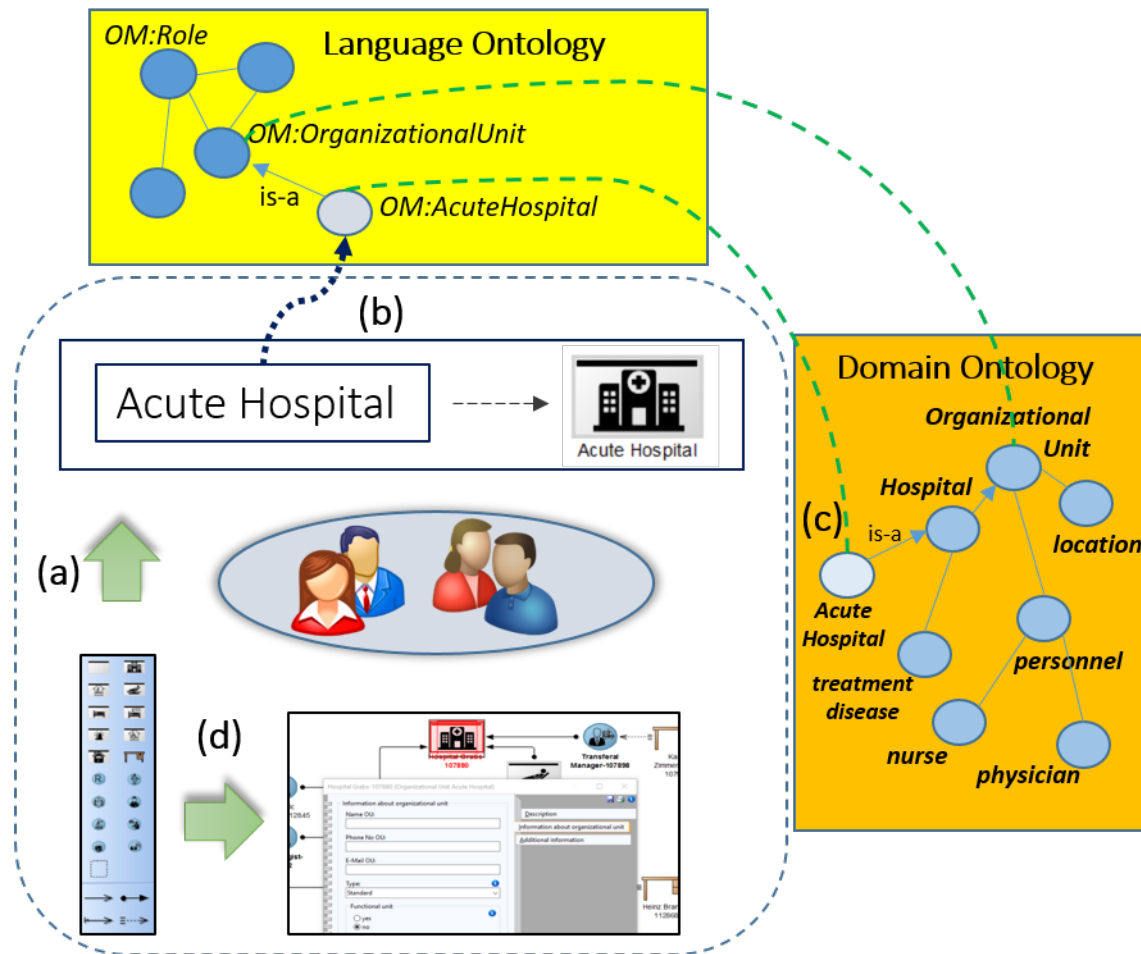


Figure 9. The Agile and Ontology-aided Modeling Environment