

Expert Interview

First Part

When: 18/07/2018

Where: Zurich

Expert: Roman Brun

Role: Enterprise Architecture and Business Process Management Consultant

Company: BOC Group

Company Website: <https://ch.boc-group.com/>

Personal Website: <http://www.romanbrun.ch/>

In Business Process Management consulting projects:

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

Most clients have difficulties in modeling with standard languages in BPM such as BPMN. This is mainly due to the too many modeling constructs (i.e. modeling elements and relations) a standard language has.

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

See above.

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

By simplifying the standard language, i.e. removing unnecessary modeling constructs.

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?*

In BPM it's good to stick to the standard BPMN as stakeholders are familiar with it. A DSML is certainly eligible to address the problem identified in question 1. However, according to the past projects, the need is more to perform restrictions on the modeling language rather than extension. Namely, removing unnecessary modeling constructs and adding constraints in the metamodel. The latter can be in the form of new relation types or new attribute values.

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, at the beginning of every project the modeling language is adapted to meet the requirements of the stakeholders. As above mentioned, the adaptation mainly refers to a restriction on objects that reside in the metamodel.

An adaptation that is often performed is such that a process hierarchy can be modeled. This is not foreseen by the BPMN but required in every project. Also we distinguish between the process landscape and process diagram. Whereas the latter follows the BPMN notation, the former has its own notation.

Also, a BPMN metamodel extension embedded in ADONIS regards the modeling relation "cross-reference". This allows linking modeling elements that belong to different models.

An example of an adapted modeling standard is the Swiss government standard «eCH» <https://www.ech.ch/vechweb/page>, where BPMN has been simplified, featuring model guidelines for modeling objects and an overall reduction of objects.

Also, there was the need to adapt modeling languages that are not standards. For instance, the Analysis Meta-Model was adapted to accommodate a new modeling element called «Processing Activity». This adaptation was done to comply with the new European regulations «General Data Protection Regulation» (GDPR) on data protection and privacy. A separate document has been sent to explain the case.

- a. *If not, Why?* Not applicable.
- b. *Could a functionality for an on-the-fly customization of modeling constructs be useful in projects with domain-specific target?*

Yes, but the adaptation of the modeling language should be performed only at the beginning of the project. Afterwards, the modeling language should remain the same to avoid inconsistency issues.

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)*

Projects focusing on BPM face less adaptation on the modeling language than Enterprise Architecture projects. Besides simplifications (on BPMN), an adaptation that recurs often take place in the design level. That is, additional semantics is likely to be added while designing models, for instance adding colors to modeling elements for further distinctions, adding the company logo and explanatory notes.

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

I have provided the GDPR use case example.

Comments on the Agile and Ontology-aided Modeling Environment

BPM consultants are continuously under pressure to quickly adapt modeling languages to accommodate needs from different stakeholders' categories. Therefore, an approach that allows quickly developing consistent metamodels could definitely help.

If specific constraints (rules) for the metamodel arise that are not yet implemented in ADONIS, the consultant can request the BOC development team to insert them. Such constraints are written in a dedicated script (ADOscript). It might require sometime until the new constraints are accommodated. If a functionality could support the consultant in this task, it would speed up the creation/adaptation of the metamodel. For example, a graphical interface that allows easily inserting the constraints.

Second Part

1. Do the operators derived in the paper (see below) make sense for you?

Yes, however, it is critical to “delete” modeling constructs after the project has started. This is for two reasons: (1) models that have been already created will be inconsistent and (2) the metamodel can get destroyed.

2. Do you suggest other operators/actions on the language for adaptation purpose?

Maybe actions that allows inserting pictures and notes. Additionally, actions that allows changing the graphical notation easily. For instance, changing color and visibility of information (e.g. attributes) from a graphical notation.

An action to hide modeling elements and relations would also be useful.

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.

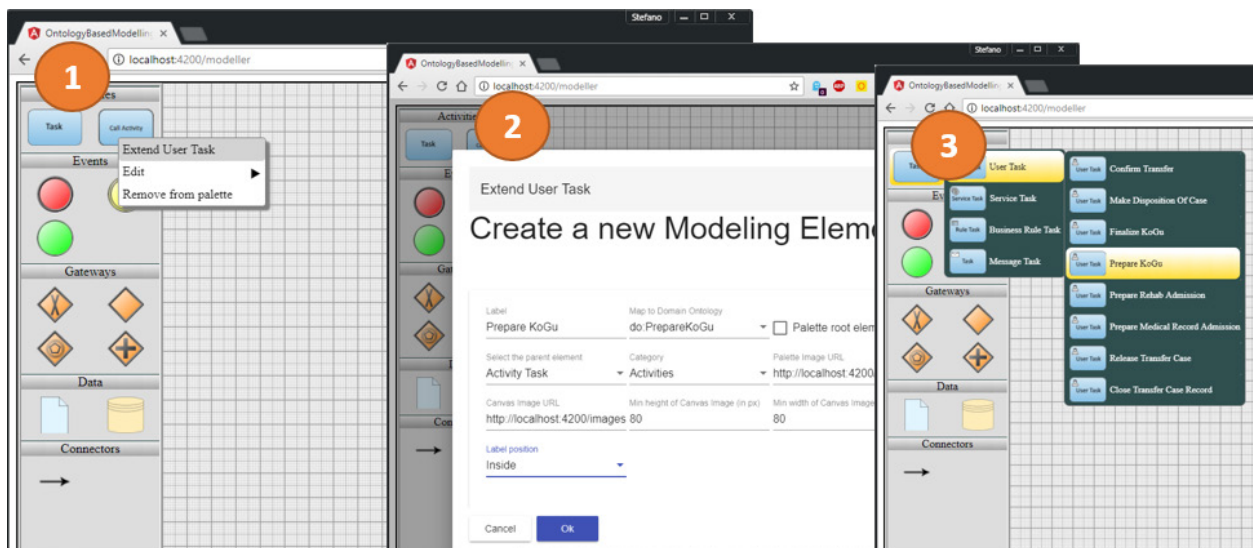


Figure 1. Operator from 1 to 3

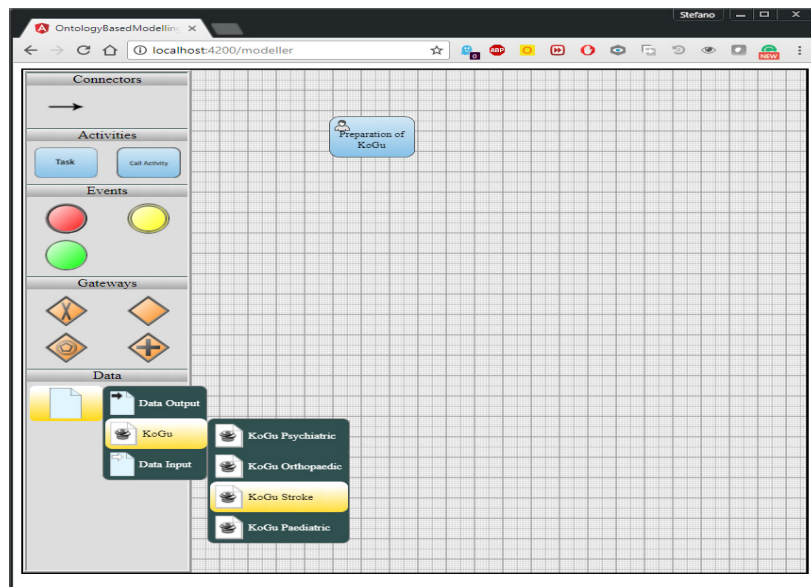


Figure 2. Operator 1 applied on "Data Object"

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.

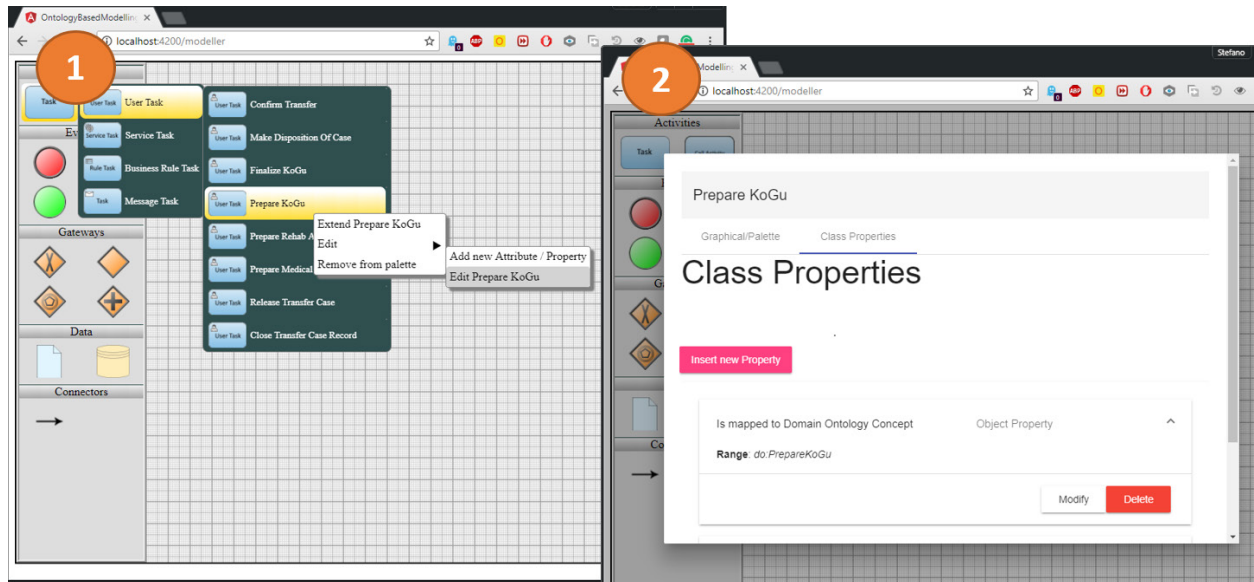


Figure 3. Operators 4 and 5

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

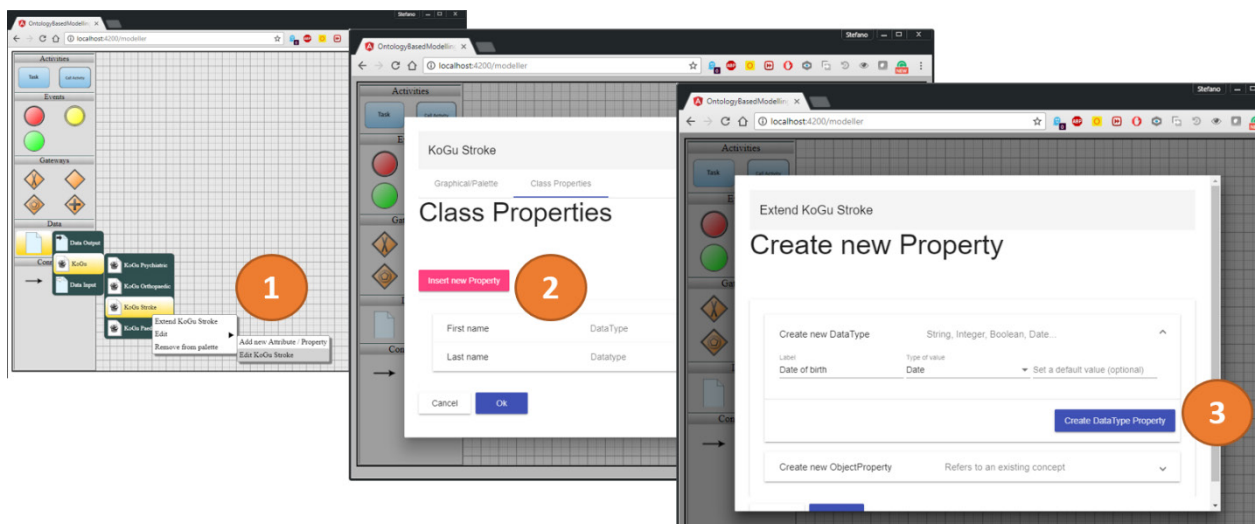


Figure 4. Operator 6 (and 3)

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.

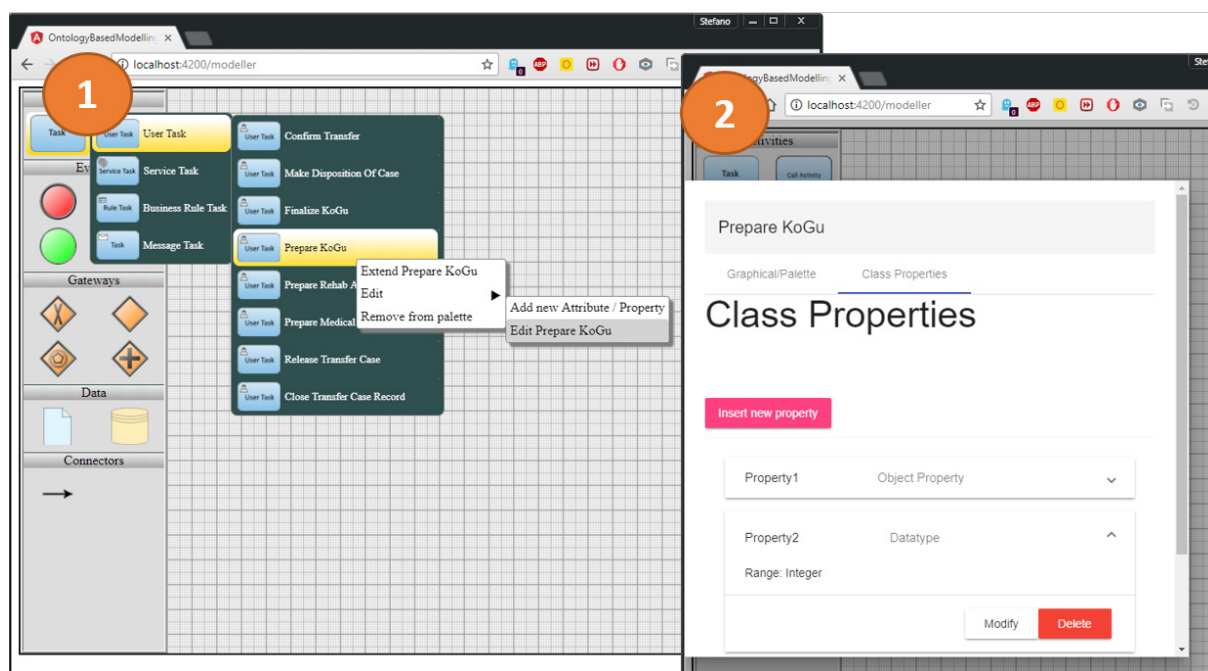


Figure 5. Operators 7 and 8

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

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

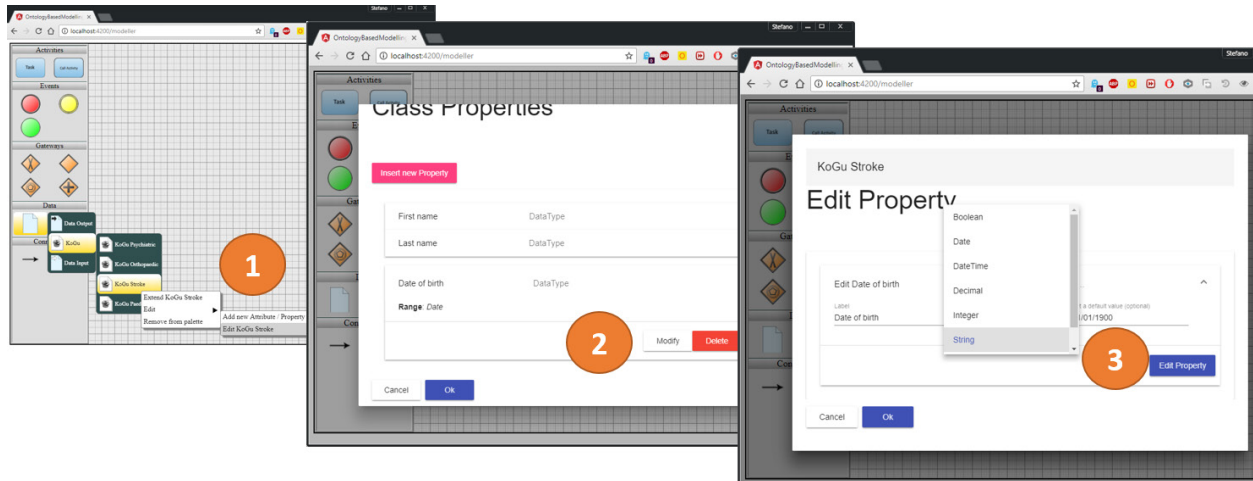


Figure 6. Operator 9 and 10

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.

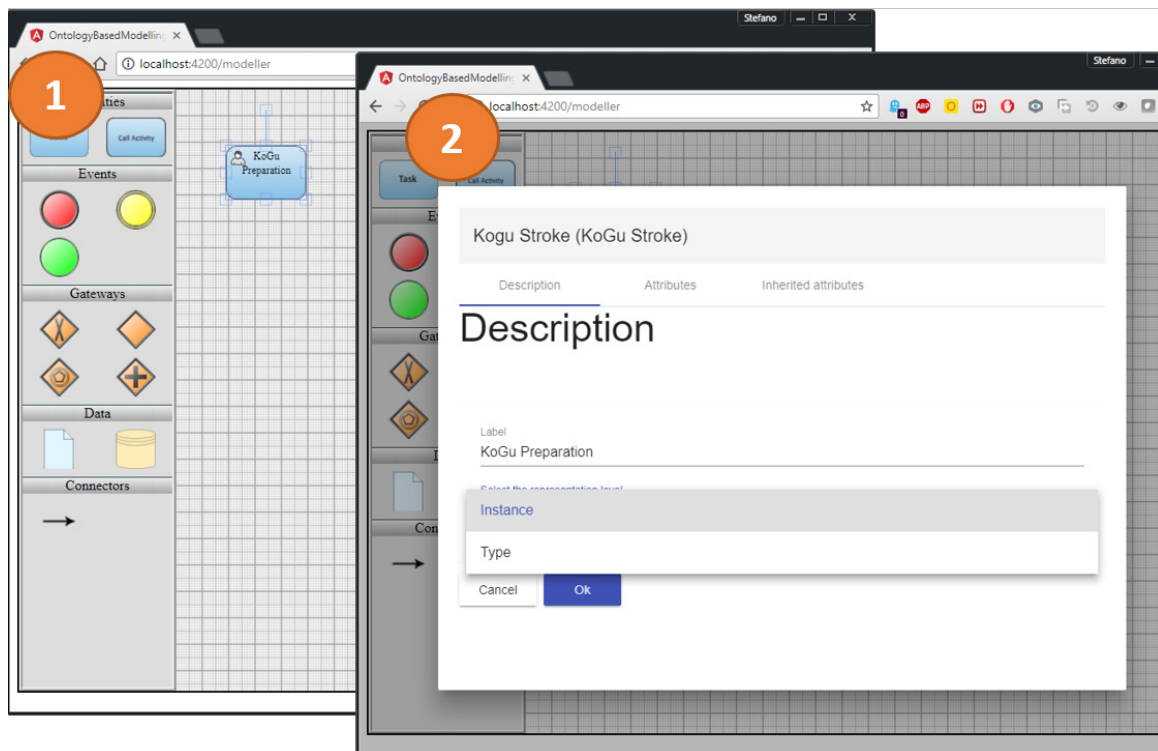


Figure 7. Operator 11

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.

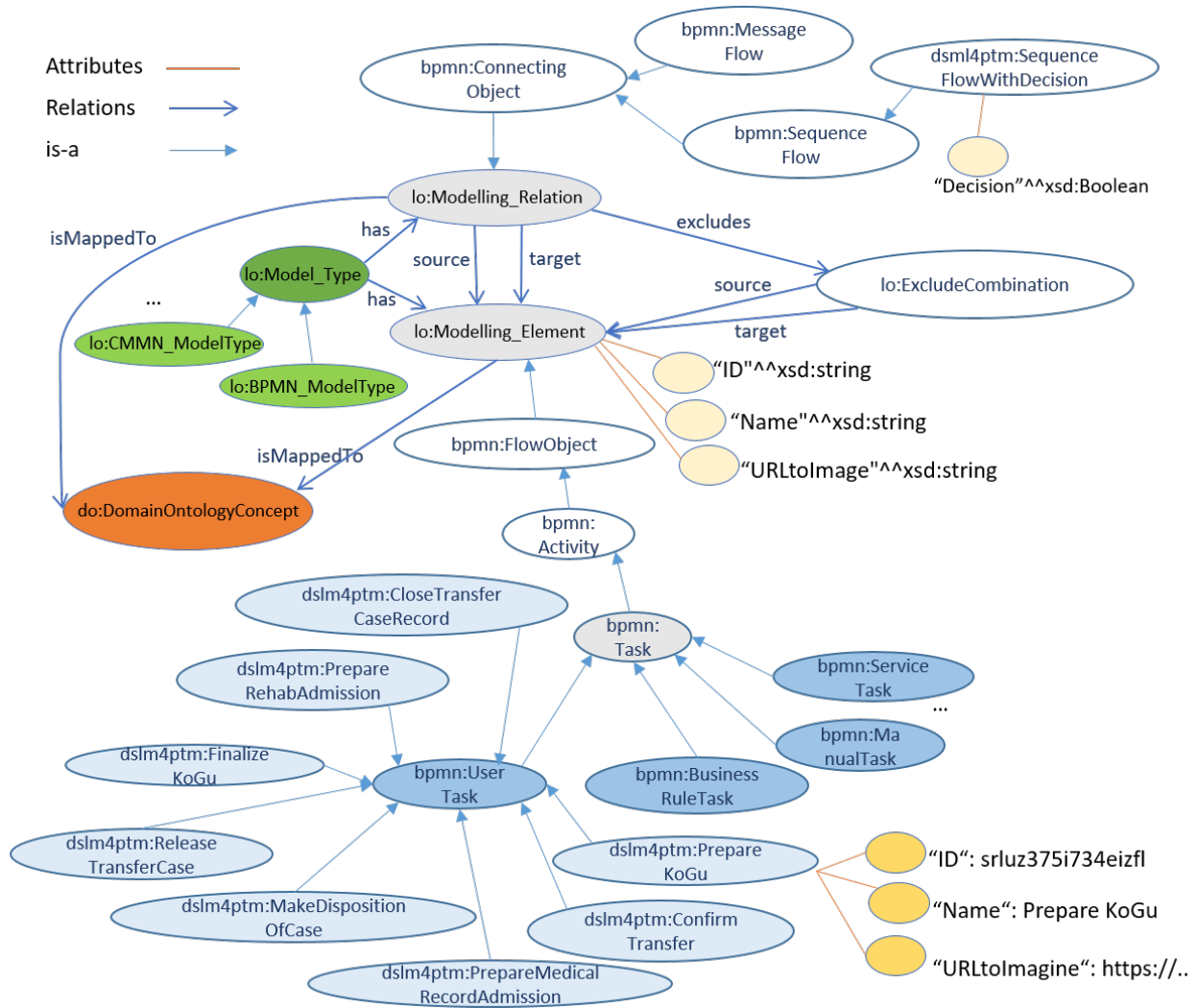


Figure 8. Ontology Excerpt

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

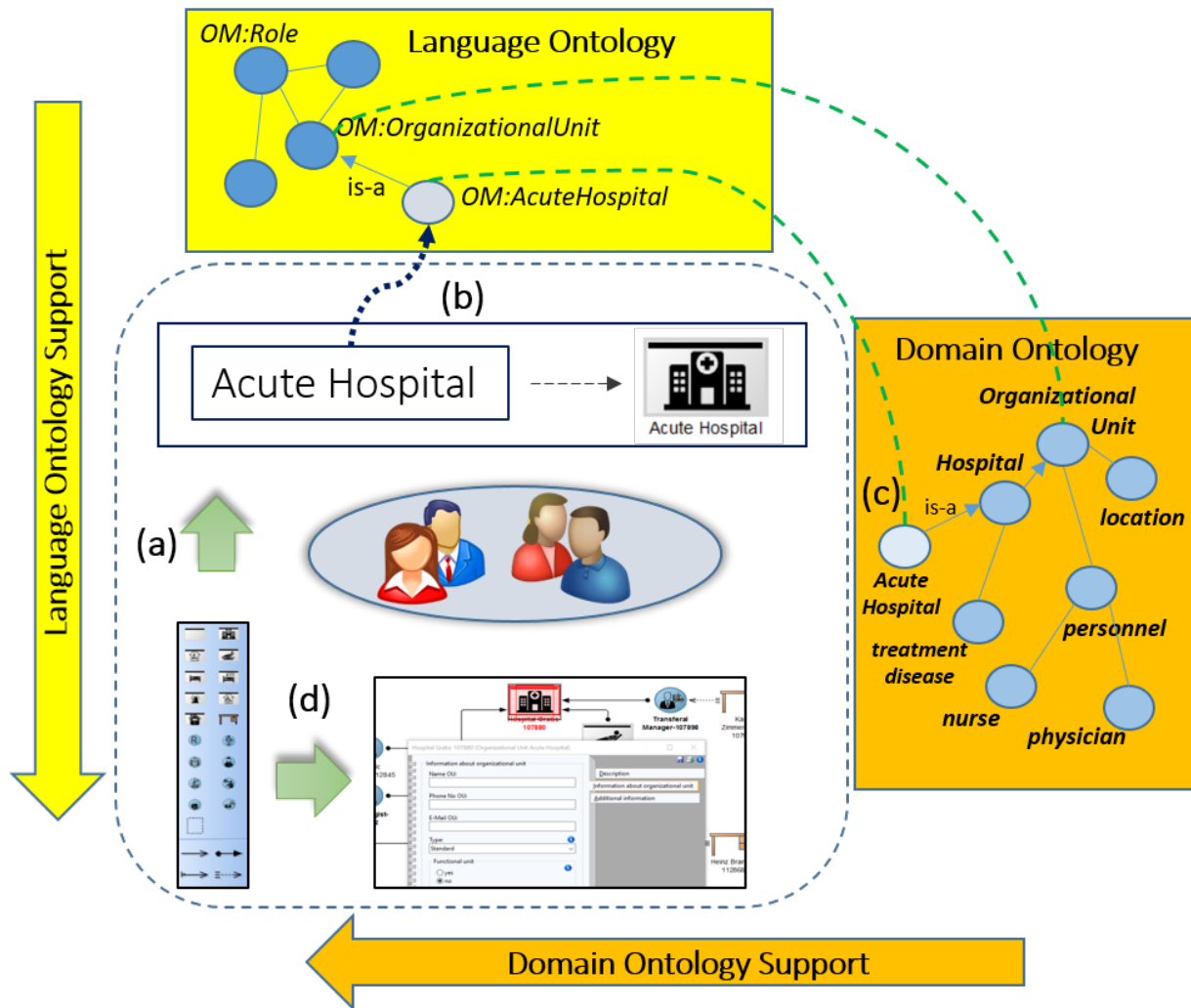


Figure 9. The Agile and Ontology-aided Modeling Environment