

# Starting Ontology Development by Visually Modeling an Example Situation - a User Study

Marek Dudáš<sup>1</sup>, Vojtěch Svátek<sup>1</sup>, Miroslav Vacura<sup>1,2</sup>, and Ondřej Zamazal<sup>1</sup>

<sup>1</sup> Department of Information and Knowledge Engineering,

<sup>2</sup> Department of Philosophy,

University of Economics, W. Churchill Sq.4, 130 67 Prague 3, Czech Republic,  
{marek.dudas|svatek|vacuram|ondrej.zamazal}@vse.cz

**Abstract.** This paper describes a user study aimed at comparing the common approach to developing an OWL ontology, using the Protégé editor alone, with an ontology development workflow starting by building a so-called PURO ontological background model in visual terms, using the tool PURO Modeler. The background model represents a complex example situation to be covered by the ontology, from which a seed of the ontology is semi-automatically generated. The evaluation suggests that starting from the background model might lead to an ontology that better covers the domain and might also alleviate some OWL encoding difficulties such as those tied to n-ary relations. On the other hand, it is more time-consuming and the user interface of the tool supporting it needs much improvement.

## 1 Introduction

In the semantic web realms, the prevailing practice of formalizing ontologies is creating them, from the onset, in OWL (with editors like Protégé), merely starting from textual specifications and informal charts. The advantages of OWL as uniform representation of ontologies throughout all ‘formal’ phases of their development lifecycle are its thorough standardization, solid support by authoring tools, and powerful reasoning abilities allowing formal consistency checking of the models. On the other hand, the direct transition from informal specifications to OWL puts quite high demands on ontology engineers. Ontology engineers directly defining OWL entities based on informal specifications have to deal with two problems at the same time: (A) “What are the entities and relations inherently described in the specification?” and (B) “How to represent them with OWL constructs?” Moreover, the latter question often has several possible answers – choosing different *OWL encoding styles*,<sup>3</sup> i.e., representing the same situation with different combinations of OWL constructs.

We have recently proposed a possible solution [4]: starting ontology development by creating a visual model in PURO language [8] representing the real world situation that is to be described by the ontology, thus answering the question A, and then configuring its automatic transformation to OWL following the desired encoding style, i.e., dealing with question B. The result of the transformation is an ontology *seed* consisting

---

<sup>3</sup> In previous publications (e.g., [4]), we used the term *OWL modeling styles*.

of classes, properties and domain, range and subClassOf axioms. This seed is then finalized by adding necessary axioms, labels, comments etc. in common IDE like Protégé. Our proposal does not replace common ontology development, it just allows making the first steps more explicit and supported by graphical tools: PURO Modeler [5] for the first step, and OBOWLMorph [4] for the transformation from PURO to OWL.

In this paper, we present an evaluation of PURO Modeler with users. Their performance creating the model in PURO is compared with creating an OWL ontology directly in Protégé.

*PURO Language* PURO<sup>4</sup> is an ontological modeling language recently drafted as common interlingua for different encoding styles in OWL. A model built in PURO is denoted as *ontological background model* (OBM). PURO inventory is very similar to that of OWL, assuring easy understandability and mappability to OWL. It is based on two distinctions: between particulars and universals and between relationships and objects (hence the PURO acronym). There are six basic entity types: B-object (particular object), B-type (type of object/type), B-relationship (particular relationship), B-relation (type of relationship), B-valuation (particular assertion of quantitative value) and B-attribute (type of valuation). An OBM consists of named entities of these types, plus of subTypeOf and instanceOf relationships. It always represents an example of a specific situation, i.e., the modeling should start from instances.

## 2 PURO Modeler and Its Possible Benefits

PURO Modeler is basically a web-based diagramming tool<sup>5</sup> designed for the PURO language. Its UI consists of a palette and a canvas. The palette serves for selecting ‘tools’ for adding instances of PURO terms and relationships between them, represented by nodes of different shapes and links of different styles. It is quite simple, there are 4 types of nodes and 3 types of links, which is enough to cover the whole PURO language. Figure 1 shows a partial screenshot of PURO Modeler including the palette and a part of an OBM.

There are three main differences in OBM-started ontology development compared to creating an ontology directly in OWL ontology editor such as Protégé. First, an OBM represents a specific example from the modeled domain. In other words, it is modeled at the level of instances, but including their types. On the other hand, when developing an ontology directly in OWL, the designer usually focuses on the T-Box. By our experience, ontology engineers think about example situations while creating the T-Box anyway, however, only implicitly. OBM allows to make such example situations explicit, which we assume might lead to achieving the intended coverage of the domain more easily. The results from the evaluation suggest this assumption is valid.

Second, the PURO language abstracts from specific aspects of OWL encoding. The most obvious example are n-ary relations, which have to be represented through reification in OWL. PURO allows to model n-ary relations as single objects, thus making the modeling easier and less error prone, as suggested by the evaluation – encoding of the

<sup>4</sup> Please refer to our previous publications ([8] and [4]) for more information.

<sup>5</sup> Available at <http://protegeserver.cz/puromodeler-v3.5>

## Starting Ontology Development by Visually Modeling an Example Situation - A User Study

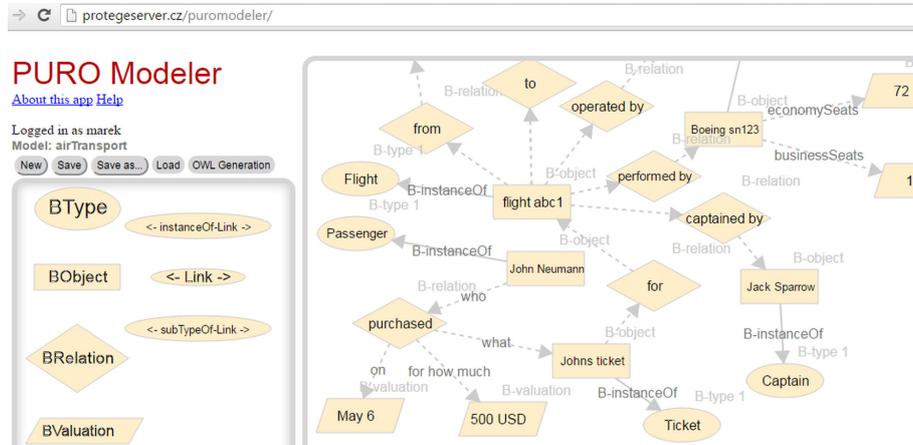


Fig. 1. The palette and a part of an example model in PURO modeler.

n-ary relation into OWL is done later by the automated PURO-to-OWL transformation in OBOWLMorph.

Third, in the OBM, the user can see all entities – types and properties – visualized in one model, while in editors like Protégé each entity type is shown in a separate subwindow. The all-in-one view is preferred by users, as suggested by our evaluation. However, this particular advantage can be achieved also in graphical OWL ontology editors such as OWLGrEd [1], where all entities are also shown and edited in one graph.

### 3 Evaluation

The evaluation<sup>6</sup> was done with 10 undergraduate students of an ontology engineering course. The students had basic knowledge about OWL (understanding of class and their hierarchy, instances, properties and domains/ranges) from the course lectures and had taken 90-minute tutorials about Protégé and PURO Modeler.<sup>7</sup> We prepared 2 example situations to be modeled by the students: an example from air transportation (A) and a human relationships example (B). Five students were asked to model A with Protégé, i.e., to create an ontology that will cover the described example, and to model B with PURO Modeler, i.e., to create an OBM that could be transformed to OWL ontology covering the example. The remaining 5 students modeled A in PURO and B with Protégé. Each student had 45 minutes to accomplish both tasks.<sup>8</sup> Each example consisted of (1) an abstract description of entity and relationship types and (2) an example situation from the domain.

<sup>6</sup> Details about the evaluation are at <http://protegeserver.cz/puroeval>

<sup>7</sup> Based on user and modeling guides, see <http://protegeserver.cz/puroeval>

<sup>8</sup> Due to lack of time, only 2 students finished both tasks. We took even unfinished results into account as they still allowed us to see what errors students made.

To make the results comparable, we instructed students to create only classes, properties and subClassOf and domain/range axioms in OWL, i.e., omit any possible complex classes, restrictions etc. since these are not created by the PURO-to-OWL transformation in OBOWLMorph. We measured the time each student needed to accomplish each task and then we examined the resulting ontologies and OBMs handed in by students. The students were also given a questionnaire focusing on comparison of PURO and direct OWL modeling at the end of the evaluation.

*Correctness* We classified the errors student made into three levels. Level 1 errors are against the syntax of the language (PURO or OWL). In PURO these are for example missing labels or wrong orientation of links. Level 2 errors are such that are correct syntactically, but do not make sense in the language: for example economy-class being a subclass of plane. Level 3 errors occur when something is modeled differently than in a gold-standard model created by us. We also tried to evaluate the overall severity of errors – whether the errors are critical and affect the whole model, or non-critical where at least part of the model is correct. An example of a critical error in case of PURO is when the students created only the types and did not include the instance-level entities.

There were no level 1 errors in the OWL ontologies as Protégé does not allow to make such errors. PURO Modeler checks for only some such errors and an obvious conclusion is that the application has to check for all syntactic errors as they occur very frequently: 7 out of 10 PURO models contained level 1 errors, in 4 cases critical.

The amount of level 2 errors is quite comparable between the two tools. 6 of 9 ontologies (one student did not hand in the result) and 4 of 6 syntactically mostly correct PURO models<sup>9</sup> contained such errors, which is actually the same percentage. A common critical level 2 error in case of PURO models was modeling only the ‘T-Box’ part.

There were no critical level 3 errors, i.e., all models without critical level 1 or 2 errors could be used without major changes. There was 1 PURO model without error, however unfinished due to lack of time. No OWL ontology was without errors. The students had problems modeling n-ary relationships in OWL. For example, no one was able to model the “is angry at someone because of something” relationship.

*Completeness* Models that did not contain any critical errors were evaluated in terms of completeness, i.e., whether they covered all relationships and entities described in the real world situation. The entities or relationships were considered covered even when there were minor errors in the model. Based on that, we found out that only 1 OWL ontology had complete coverage, in contrast with 3 such PURO models.

*Time* Relevant time measurement is only for the first task (example A), as most students did not finish the second task. The average times were 32 minutes to create an OBM and 26 minutes for ontology created directly in OWL.

*Questionnaire* The questionnaire contained 9 questions. First question was about how often students hesitated about mapping from the textual example description to PURO

---

<sup>9</sup> We did not check syntactically incorrect models for level 2 errors as it is meaningless.

term, with 5-level scale of answers between ‘never’ and ‘very often’. Then there were 5 questions comparing PURO Modeler and Protégé in terms of UI-friendliness, fun, speed, easy understanding and discrete views in Protégé (classes and relationships in separate subwindows) vs. all-in-one view in PURO Modeler. These had also 5-level scale from ‘definitely PURO Modeler’ to ‘definitely Protégé’. Next question asked students whether they used rather the general domain description or the concrete example as the main source for the modeling. Eighth question was about how hard it was to understand PURO language. In the last question the students could write in a free-text what would they improve in PURO Modeler.

The answers to the questionnaire<sup>10</sup> were generally pro-PURO, however, the students could have been biased by desire to speak positively about their teacher’s research. The students were unsure about the description-to-PURO mapping mostly ‘sometimes’, considered PURO Modeler more user friendly, more fun and easier to understand than Protégé. They considered work in PURO Modeler faster than in Protégé, even though the reality is the opposite. The students used rather general descriptions as the source for the model and considered the PURO language rather easy to understand. They preferred the all-in-one view in PURO Modeler over separate windows in Protégé.

*Summary* The evaluation suggests modeling in PURO is a little bit slower and more error prone, partially due to less strict UI, however leads to better coverage of the domain. Both the time consumption and number of errors is quite comparable in OWL and PURO. According to the questionnaire, students generally prefer PURO Modeler over Protégé. Note that the evaluation is focused on PURO Modeler. To obtain an (seed of) actual OWL ontology, the PURO model would have to be transformed in OBOWL-Morph. Such transformation, when using default settings, is fully automated and needs literally just three clicks. When not using default settings, i.e., changing the target OWL encoding style in OBOWLMorph, the evaluation would become much more complicated and we wanted to focus on the first step (PURO Modeler) first, leaving OBOWL-Morph evaluation for future work.

## **4 Related Research**

*Starting ontology development from a simplified model* OntoUML [3] is a conceptual modeling language based on UML and grounded in the Universal Foundational Ontology (UFO). OLED, the graphical editor for OntoUML, allows to transform it into OWL fragments. The transformation is hard-coded and each OntoUML element has its single OWL counterpart. Bauman [2] implemented XSLT transformation of conceptual models into XML Schema, while OWL as target is only mentioned as possible future work. The user can choose a sort of encoding style, e.g., whether to transform a concept to an XML attribute or child-element. To allow reusing existing ER diagrams, Fahad [6] designed their rule-based transformation to OWL ontologies. The framework is however not intended as a general ontology development alternative. In all mentioned OWL generation methods, the input model is created at the level of types. In our approach, in contrast, the input model is created as an example situation at the instance level.

---

<sup>10</sup> The whole questionnaire and answers are available at <https://goo.gl/MR6aS1>

*Evaluation of ontology development tools* Lambrix et al. [7] did an evaluation of Protégé 2000, Chimaera, DAG-Edit and OilEd. They asked users to perform specific tasks, but the evaluation was based on a questionnaire rather than the actual user performance. Our situation is somewhat different, as we are comparing different approaches, rather than just different tools. Similarly to our evaluation, they admit omitting tests of scalability – the evaluation was done only using small parts of an ontology.

## 5 Conclusions and Future Work

We compared common ontology development in Protégé with development started from an ontological background model in PURO Modeler in an evaluation with users. Results suggest that users prefer starting from PURO Modeler over Protégé, thanks to its simpler interface and better visualization, and are able to create models better covering the domain. Creating the background model takes however more time and is more error prone in terms of syntactic errors. Future research will include improving PURO Modeler and doing a full-scale evaluation including transformation to OWL in OBOWL-Morph and finalizing the ontology in a common ontology editor like Protégé. We will also compare PURO Modeler with graphical ontology editors like OWLGrEd. A specific aspect that we will have to focus on is scalability – we will have to implement advanced visualization techniques and test development of full-scale ontologies started from OBMs.

**Acknowledgments** *The research is supported by UEP IGA F4/28/2016. Ondřej Zamazal is supported by CSF 14-14076P.*

## References

1. Bārzdīņš, J., Bārzdīņš, G., Čerāns, K., Liepiņš, R., Sproģis, A.: OWLGrEd: a UML style graphical notation and editor for OWL 2. In: Proc. 7th International Workshop OWL: Experience and Directions (OWLED-2010) (2010)
2. Bauman, B.T.: Prying apart semantics and implementation: Generating XML schemata directly from ontologically sound conceptual models. In: Proceedings of Balisage: The Markup Conference 2009
3. Benevides, A.B., Guizzardi, G.: A model-based tool for conceptual modeling and domain ontology engineering in OntoUML. In: Enterprise Information Systems, pp. 528–538. Springer (2009)
4. Dudáš, M., Hanzal, T., Svátek, V., Zamazal, O.: OBOWLMorph: Starting ontology development from PURO background models. In: International Experiences and Directions Workshop on OWL. pp. 14–20. Springer (2015)
5. Dudáš, M., Hanzal, T., Svátek, V.: What can the ontology describe? Visualizing local coverage in PURO modeler. In: VISUAL@EKAW (2014)
6. Fahad, M.: Er2owl: Generating owl ontology from er diagram. In: Intelligent Information Processing IV, pp. 28–37. Springer (2008)
7. Lambrix, P., Habbouche, M., Perez, M.: Evaluation of ontology development tools for bioinformatics. *Bioinformatics* 19(12), 1564–1571 (2003)
8. Svátek, V., Homola, M., Kluka, J., Vacura, M.: Metamodeling-based coherence checking of OWL vocabulary background models. In: OWLED (2013)