Web Interoperability For Ontology Development And Support With crowd 2.0

Germán Braun[1,2*] , **Giuliano Marinelli**[1,2] , **Emiliano Rios Gavagnin**[1] , **Laura Cecchi**[1] and **Pablo Fillottrani**[3,4]
[1]Universidad Nacional del Comahue
[2]Consejo Nacional de Investigaciones Científicas y Técnicas
[3]Universidad Nacional del Sur
[4]Comisión de Investigaciones Científicas de la provincia de Buenos Aires
{german.braun, giuliano.marinelli}@fi.uncoma.edu.ar, emiliano.rios@est.fi.uncoma.edu.ar,
lcecchi@fi.uncoma.edu.ar, prf@cs.uns.edu.ar

Abstract

In this work, we treat web interoperability in terms of interchanging ontologies (as knowledge models) within user-centred ontology engineering environments, involving visual and serialised representations of ontologies. To do this, we deal with the tool interoperability problem by re-using an enough expressive ontology-driven metamodel, named KF, proposed as a bridge for interchanging both knowledge models. We provide an extensible web framework, named crowd 2.0, unifying the standard conceptual data modelling languages for generating OWL 2 ontologies from semantic visualisations. Visual models are designed as UML, ER or ORM 2 diagrams, represented as KF instances, and formalised as logic-based models. Reasoning results may be newly incorporated into the shared KF instance to be visualised in any of the provided languages.

1 Introduction

The Semantic Web aims at generating machine-understandable resources on the web to be shared among agents [Berners-Lee *et al.*, 2001]. Ontologies take central stage in this framework as a common terminology for such agents. However, one of the barriers for the adoption of these technologies is the weak support of integrated tools for addressing the ontology development and support activities. In particular, they must deal with the interoperability problem [Vigo *et al.*, 2014; Braun *et al.*, 2019], which refers to the ability of two or more tools to exchange ontologies and manipulate them. Ontology Engineering (OE) methodologies must integrate distinct tasks and related tools that support them, then interoperability is an important issue [Garcia-Castro and Gómez-Pérez, 2010]. These scenarios require working on the interaction of complex and heterogeneous representation formalisms, with possibly different expressiveness power and paradigms, during the ontology life cycle.

In this paper, we focus on the interaction between visual and serialised models in the context of a framework for ontology development and support tasks [Keet, 2018]. The two concrete contributions are: (1) a full-fledged Ontology Engineering (OE) workflow for semantics visualisation of ontologies. Our approach is based on the re-use of an enough expressive ontology-driven metamodel, named KF [Keet and Fillottrani, 2015], proposed as a bridge for interchanging visual and serialised knowledge models, preserving their respective semantics. (2) a web framework, which implements the workflow. This framework unifies the standard UML, ER and ORM 2 for generating and reasoning over OWL 2 ontologies [Horrocks *et al.*, 2000] from visual diagrams. First versions of it has been published in [Braun *et al.*, 2020; Gavagnin *et al.*, 2020], where we proposed a reference architecture, two independent UML/ORM 2 editors together with Description Logics (DL)-based automated support. In this work, we have refactored the whole framework and the editors, and integrated a new ER editor together with a KF web service to provide inteoperabilty among them. Moreover, we have implemented a new DL encoding relative to the metamodel and thus reasoning over models regardless of how they are visualised. The last version is available at http://crowd.fi.uncoma.edu.ar:3335.

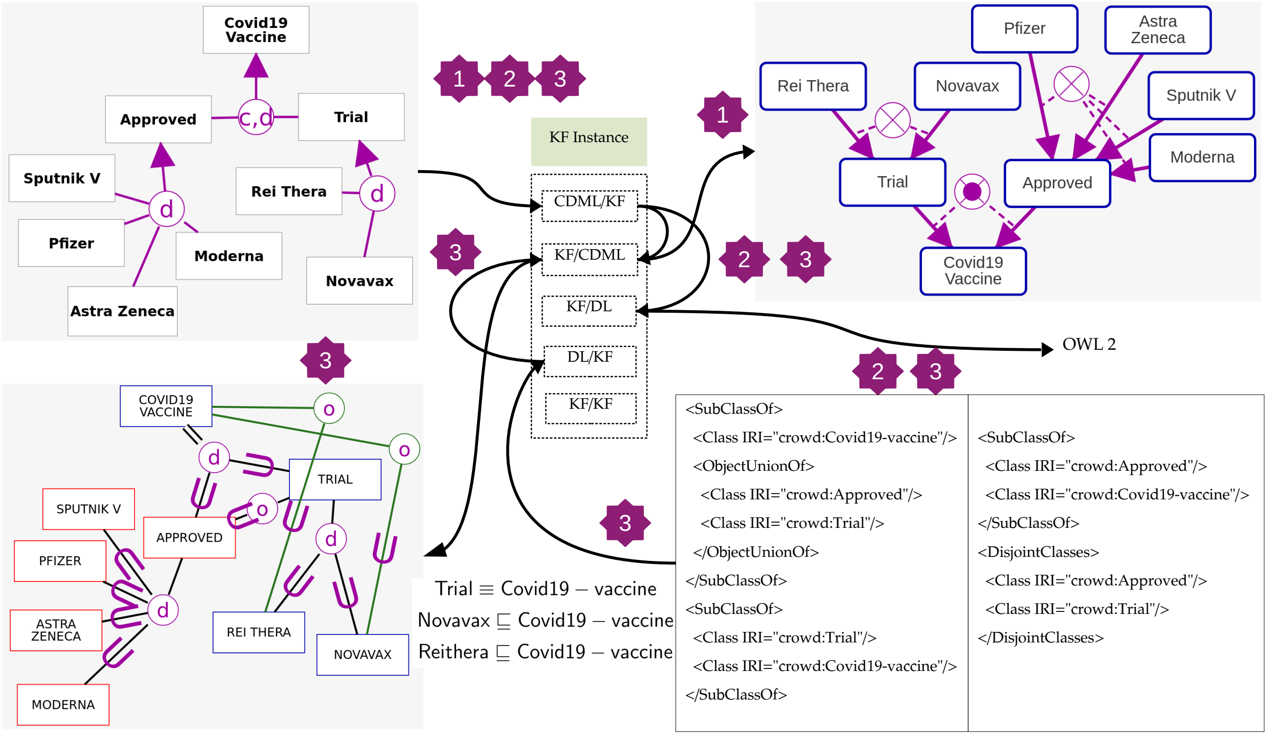
The remainder of this paper discusses about the application domain (Section 2), the problem scenario (Section 3), the technical details (Section 4), and finally contributions (Section 5) and conclusions (Section 6).

2 Application Domain

We revisit the following concepts related into the context of the interoperability between visual tools (human-oriented) and logic-based tools (machine-oriented) for ontology development and support tasks.

Visual Tools. We can identify two different types of ontology visualisations [Dudás *et al.*, 2018]: *Axiomatic visualisation* (a.k.a ontology visualisation) is about drawing its structure at OWL 2 constructor level, which may be closer to users highly skilled in formal representations. *Semantics visualisation* (a.k.a knowledge visualisation) is involved in depicting the underlying semantics covered by an ontology, i.e. its conceptual model, hiding formal aspects from

[*] Contact Author



$\text{Trial} \equiv \text{Covid19} - \text{vaccine}$
$\text{Novavax} \sqsubseteq \text{Covid19} - \text{vaccine}$
$\text{Reithera} \sqsubseteq \text{Covid19} - \text{vaccine}$

```
<SubClassOf>
  <Class IRI="crowd:Covid19-vaccine"/>
  <ObjectUnionOf>
    <Class IRI="crowd:Approved"/>
    <Class IRI="crowd:Trial"/>
  </ObjectUnionOf>
</SubClassOf>
<SubClassOf>
  <Class IRI="crowd:Trial"/>
  <Class IRI="crowd:Covid19-vaccine"/>
</SubClassOf>
```

```
<SubClassOf>
  <Class IRI="crowd:Approved"/>
  <Class IRI="crowd:Covid19-vaccine"/>
</SubClassOf>
<DisjointClasses>
  <Class IRI="crowd:Approved"/>
  <Class IRI="crowd:Trial"/>
</DisjointClasses>
```

Figure 1: COVID19-Vaccine. (1) scenario 1. (2) scenario 2. (3) scenario 3. Rules CDML/KF, KF/CDML, KF/DL, DL/KF, KF/KF represent set of rules that are used to transform/generate into another model.

domain experts. We found diverse *ad-hoc* languages and tools as VOWL [Lohmann *et al.*, 2016], Eddy [Console *et al.*, 2014], Graffo [Falco *et al.*, 2014] and Protégé plug-ins (OWLViz[1], OntoGraf[2], etc), as examples of the first concept. Similarly, the set of tools related to the second concept includes: NORMA [Curland and Halpin, 2010], for the highly expressive ORM 2 [Halpin and Morgan, 2008], ICOM [Fillottrani *et al.*, 2012], and lastly, OWLGrEd [Cerans *et al.*, 2012] based on an extended version of UML [Booch *et al.*, 2005]. In last ones, UML, ER/EER and ORM 2, take centre stage. Thus, using axiomatic visualisation would improve the tool interoperability because the visual knowledge model (OWL 2 visual notation) is similar to the formal one (OWL 2) [Garcia-Castro and Gómez-Pérez, 2010]. However, the cost of ensuring this level of interoperability is high, presenting visual ambiguity in equivalent models and/or weak support for detecting non-trivial consequences, such as stricter cardinalities [Braun *et al.*, 2019]. On the contrary, the semantics visualisation requires dealing with a more complex interoperability panorama because the knowledge models involved have different expressive power.

Logic-based Tools. One of the assumptions in the field of Knowledge Representation and Reasoning (KRR) is that an agent's knowledge is represented in a declarative form, suitable for processing by reasoning engines. In particular, machine-understandable ontologies are essentially defined as DL knowledge bases [Horrocks *et al.*, 2000], and specified into a standard ontology language as OWL 2. Nevertheless, the development of 'good' ontologies still depends on human experts. Ontologies are often large and complex, making it difficult to avoid errors. Thus, the automated reasoning provided by DL systems allow modellers to detect errors, such as the inconsistency of the whole ontology, the unsatisfiability of a concept, or relationships do not hold in the domain.

Thus, the benefits of putting together the previous concepts is twofold. First, we must make it easy for human modellers, whom require of visual tools (integrated to automated reasoners) for development, debugging and maintenance of ontologies. Second, to implement machine-readable OWL 2 ontologies from human-oriented visual diagrams as the shared knowledge of agents.

3 Problem Scenario

Our goal was to develop a framework that tackle the tool interoperability problem. We instantiated our approach for treating UML, ER and ORM 2 interoperability for ontology development. We implemented the interaction between these languages and the KF metamodel along with interoperability rules, and also provided a formalisation in DL and its serialisation in OWL 2. Some scenarios are detailed below and depicted in Fig. 1, where we briefly model the current status of some Covid19 vaccines[3]. A short com-

[1] http://protegewiki.stanford.edu/wiki/OWLViz
[2] http://protegewiki.stanford.edu/wiki/OntoGraf
[3] https://www.nytimes.com/interactive/2020/science/coronavirus-vaccine-tracker.html

plementary video showcasing these scenarios is available at https://youtu.be/gcMr424MPjI.

Scenario 1: Providing diverse conceptual views for the same conceptualisation This scenario applies when users and/or domain experts with diverse skills try to communicate each other about the current conceptualisation. Then users load their editors of preference to import the model. The activities involve are: UML/ER/ORM 2 diagram, KF instance (applying the respective CDML/KF rules) and export UML/ER/ORM 2 (applying KF/CDML rules). In Fig. 1, the model about Covid19 vaccines is converted from UML to ORM 2.

Scenario 2: Implementing a conceptualisation The implementation of the current conceptualisation can be run in two different situations: (a) the implementation generates an OWL 2 ontology to be shared with ontology engineers; (b) once all the stakeholders agree about a possible version of an ontology, then it must be implemented as a computable specification. This scenario involves: UML/ER/ORM2 diagram, KF instance (CDML/KF rules), DL Encoding (KF/DL embedding rules) and export OWL 2. A brief OWL 2 is shown below.

```
<SubClassOf>
    <Class IRI="crowd:Approved"/>
    <Class IRI="crowd:Covid19-vaccine"/>
</SubClassOf>
<SubClassOf>
    <ObjectIntersectionOf>
        <Class IRI="crowd:Approved"/>
        <Class IRI="crowd:Trial"/>
    </ObjectIntersectionOf>
    <Class abbreviatedIRI="owl:Nothing"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="crowd:Covid19-vaccine"/>
    <ObjectUnionOf>
        <Class IRI="crowd:Approved"/>
        <Class IRI="crowd:Trial"/>
    </ObjectUnionOf>
</SubClassOf>
<SubClassOf>
    <Class IRI="crowd:Trial"/>
    <Class IRI="crowd:Covid19-vaccine"/>
</SubClassOf>
```

Scenario 3: Updating a conceptualisation after running automated reasoning This scenario applies when stakeholders want to evaluate the current conceptualisation using reasoning. They will see the reasoning results over the same diagram. It involves: UML/ER/ORM 2 diagram, KF instance (CDML/KF rules), DL Encoding (KF/DL embedding rules), DL Reasoning, Update KF instance (DL/KF embedding rules) and export UML/ER/ORM 2 (applying KF/CDML rules). In this last task, the resulting diagram might be visualised using any language (possibly different from the initial one).

Fig. 1 illustrates this scenario. Let us suppose that any of stakeholders asserts a subclass constraint between $Approved$ and $Trial$ vaccines, meaning that approved vaccines are also trial of them. Clearly, this is a source of inconsistencies because vaccines could not have the same status at the same time. As another consequence, covid19 vaccines are only trial ones ($\mathsf{Trial} \equiv \mathsf{Covid19} - \mathsf{vaccine}$). No primitive is available for drawing equivalences in ER modelling language, however, two new subtyping constraints $\mathsf{Novavax} \sqsubseteq \mathsf{Covid19} - \mathsf{vaccine}$ and $\mathsf{Reithera} \sqsubseteq \mathsf{Covid19} - \mathsf{vaccine}$ are inferred because of that.

4 Technical Details

The framework developed implements the interaction between a `crowd 2.0 client`[4] and multiple applications on the server side via WebMethod calls. The client is a web application implemented with Angular framework and JavaScript plugins. The server is composed of three main applications: a nodeJS application, for connecting a MongoDB database; a Java application, `KF API`[5], which implements the KF rules and connects with the OWL API for OWL 2 importing; and a PHP application `crowd 2.0 core`[6], in charge of the interaction with the DL reasoners (Racer, Konclude, or any other, via OWLlink protocol), processing the reasoning outputs for updating the KF models (if necessary), and finally, writing OWL 2 specifications from such KF instances.

5 Contribution

Ontology development tools present diverse interoperability levels going from high to low as their knowledge models are less similar in terms of expressiveness. Existing tools do not explicitly addressed this issue but they present alternative solutions such as defining (or extending) new visual representations. Even worst, some of them only provide modelling capabilities at logic level, depending on experts skilled in formal representations. crowd 2.0 addresses this problem.

The innovation here is to guarantee independence from visual languages and at the same time, from logic-based tools and languages. The KF metamodel allows us track the semantic gaps between different conceptual models. Thus, modellers can represent the domain knowledge in a language of preference, delegating both semantic conversions and reasoning over models to crowd 2.0. Lastly, its main limitation is importing arbitrary OWL 2 ontologies to be reused (visualised or edited). Nevertheless, this scenario implies embedding a more expressive subset of OWL 2 axioms into KF instances, which is being currently developed.

6 Conclusion

crowd 2.0 is a web framework, designed to work in early tasks of ontology development and support. To do this, we tackled the tool interoperability problem in the context user-centred tools by providing intermediate instances of an enough expressive metamodel. The framework is explicitly focused on solving UML, ER and ORM 2 interoperability for ontology development, which means that we must consider increasing the whole expressiveness of this approach as future work. We also plan to assess the usability of the tool by evaluating usability factors (SUS) [Lewis and Sauro, 2009]. A video demo is available at https://www.youtube.com/watch?v=0u2-WP6NAuM&t=33s.

[4] https://bitbucket.org/gilia/crowd-app/

[5] https://bitbucket.org/gilia/metamodelapi/

[6] https://bitbucket.org/gilia/reasoning/

References

[Berners-Lee *et al.*, 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.

[Booch *et al.*, 2005] Grady Booch, James Rumbaugh, and Ivar Jacobson. *Unified Modeling Language User Guide*. Addison-Wesley Professional, 2005.

[Braun *et al.*, 2019] Germán Braun, Laura Cecchi, and Pablo Fillottrani. Taking Advantages of Automated Reasoning in Visual Ontology Engineering Environments. In *JOWO@BOG*, 2019.

[Braun *et al.*, 2020] Germán Braun, Christian Gimenez, Laura Cecchi, and Pablo Fillottrani. crowd: A Visual Tool for Involving Stakeholders into Ontology Engineering Tasks. *KI - Künstliche Intelligenz*, 34(3):365–371, 2020.

[Cerans *et al.*, 2012] Karlis Cerans, Julija Ovcinnikova, Renars Liepins, and Arturs Sprogis. Advanced OWL 2.0 Ontology Visualization in OWLGrEd. In *DB&IS*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2012.

[Console *et al.*, 2014] Marco Console, Domenico Lembo, Valerio Santarelli, and Domenico Savo. Graphol: Ontology Representation through Diagrams. In *27th International Workshop on Description Logics*, 2014.

[Curland and Halpin, 2010] Matthew Curland and Terry A. Halpin. The NORMA Software Tool for ORM 2. In *CAiSE Forum*, Lecture Notes in Business Information Processing. Springer, 2010.

[Dudás *et al.*, 2018] Marek Dudás, Steffen Lohmann, Vojtech Svátek, and Dmitry Pavlov. Ontology visualization methods and tools: a survey of the state of the art. *Knowl. Eng. Rev.*, 33:e10, 2018.

[Falco *et al.*, 2014] Riccardo Falco, Aldo Gangemi, Silvio Peroni, David M. Shotton, and Fabio Vitali. Modelling OWL Ontologies with Graffoo. In *The Semantic Web: ESWC 2014*, 2014.

[Fillottrani *et al.*, 2012] Pablo Fillottrani, Enrico Franconi, and Sergio Tessaris. The ICOM 3.0 intelligent conceptual modelling tool and methodology. *Semantic Web*, 3(3):293–306, 2012.

[Garcia-Castro and Gómez-Pérez, 2010] Raul Garcia-Castro and Asunción Gómez-Pérez. Interoperability results for Semantic Web technologies using OWL as the interchange language. *J. Web Semant.*, 8(4):278–291, 2010.

[Gavagnin *et al.*, 2020] Emiliano Rios Gavagnin, Germán Braun, Laura Cecchi, and Pablo Fillottrani. Towards an Ontology Engineering Framework for IntegratingVisualisation, Metamodelling and Reasoning. In *Proceedings of the 13th Seminar on Ontology Research*, 2020.

[Halpin and Morgan, 2008] Terry Halpin and Tony Morgan. *Information Modeling and Relational Databases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008.

[Horrocks *et al.*, 2000] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Reasoning with Individuals for the Description Logic SHIQ. In *Automated Deduction - CADE-17, Proceedings*, volume 1831 of *Lecture Notes in Computer Science*, pages 482–496. Springer, 2000.

[Keet and Fillottrani, 2015] Maria Keet and Pablo Fillottrani. An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2. *Data Knowl. Eng.*, 98:30–53, 2015.

[Keet, 2018] Maria Keet. *An Introduction to Ontology Engineering*. University of Cape Town, 2018.

[Lewis and Sauro, 2009] James Lewis and Jeff Sauro. The factor structure of the system usability scale. In *Proceedings of the 1st International Conference on Human Centered Design: Held As Part of HCI International 2009*. Springer-Verlag, 2009.

[Lohmann *et al.*, 2016] Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. Visualizing ontologies with VOWL. *Semantic Web*, 7(4):399–419, 2016.

[Vigo *et al.*, 2014] Markel Vigo, Samantha Bail, Caroline Jay, and Robert D. Stevens. Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. *Int. J. Hum. Comput. Stud.*, 72(12):835–845, 2014.