Formal modelling of ontologies within Event-B

Yamine Aït-Ameur, Idir Ait-Sadoune, Kahina Hacid, Linda Mohand Oussaid

Formal modelling of ontologies within Event-B*

Yamine AIT AMEUR
IRIT/INPT-ENSEEIHT
Toulouse, France
yamine@enseeiht.fr

Idir AIT SADOUNE
LRI/CentraleSupelec
Gif Sur Yvette, France
idir.aitsadoune@centralesupelec.fr

Kahina HACID
IRIT/INPT-ENSEEIHT
Toulouse, France
kahina.hacid@enseeiht.fr

Linda MOHAND OUSSAID
LRI/CentraleSupelec
Gif Sur Yvette, France
linda.mohandoussaid@centralesupelec.fr

This paper reports on the results of the French ANR IMPEX research project dealing with making explicit domain knowledge in design models. Ontologies are formalised as theories with sets, axioms, theorems and reasoning rules. They are integrated to design models through an annotation mechanism. Event-B has been chosen as the ground formal modelling technique for all our developments. In this paper, we particularly describe how ontologies are formalised as Event-B theories.

1 Introduction

Nowadays, it is well accepted that formal ontologies are commonly used as support for the axiomatisation of the knowledge describing a domain of interest. In particular, for domains in the engineering area where concepts are well mastered by the different stakeholders, ontologies play a major role for knowledge exchange and heterogeneity reduction.

Meanwhile, we observe that defining a formal framework for integrating both ontologies represented by knowledge models and design models of particular systems did not draw the attention of many researchers in system engineering.

Approaches like those of [11][3][4][5][7] supporting the integration of both ontologies and design models contribute to strengthen these design models by offering the capability to design models to borrow knowledge from ontologies, using a particular annotation relationship. As a consequence, the design models are enriched and strengthened with axioms, theorems or invariants issued from the used ontologies.

This paper presents a summary of the work achieved in the context of the French ANR IMPEX research project. Ontologies are formalised as theories with axioms, theorems and reasoning rules. Event-B has been chosen as the ground formal modelling technique for all our developments.

2 Need to embed ontologies in formal developments

When design models are produced, designers use domain knowledge in order to formalise the concepts and components of the system to be designed. Usually, this knowledge is not made explicit and is used in an empirical manner. There is no complete formalisation for the reasoning that can be associated to this knowledge.

*The work reported in this paper has been supported by the ANR project IMPEX ref : Projet-ANR-13-INSE-0001

To appear in EPTCS.

Embedding ontologies in design models in a modular way makes it possible to use ontology concepts and associated reasoning rules in the design models. The interest is to strengthen the models as shown in our previous work [11][3][4][5][7]. The Event-B method has been set up to show how our approach works.

When integrating ontologies and design models, the main difficulty consists in defining a sound integration operation in order to overcome the difficulties resulting from possible semantic gaps that may occur due to the use of ontologies in formal development models. For example, we have adopted the closed world assumption that fits with the studied systems.

3 Ontologies as theories

As mentioned above, ontologies are formalised as theories integrated to formal system modelling languages. In the context of the IMPEX project, we have identified two approaches to define ontologies as formal theories. These two approaches use two different modelling processes: shallow and deep modelling.

3.1 Shallow modelling: Ontologies as contexts [9]

The approach that uses shallow modelling consists in modelling the ontology concepts directly in the target modelling language without keeping trace of the structure of the ontology modelling language concepts. One way to integrate the ontology concepts into a specific formal method development process is to express the ontologies languages constructs into the target formal language by means of transformation rules. In our case, a shallow modelling approach consists in encoding the ontology concepts (classes, properties, ...) directly in an Event-B context by using abstract sets, constants and axioms.

3.2 Deep modelling: Ontologies as instances of ontology models [8, 6]

The approach that uses deep modelling consists in modelling the ontology concepts together with the concepts of the modelling language that were used to define the ontology concepts. Here, ontologies are defined as instances of ontology models. Two steps are required. First, an ontology model is formalized and then ontologies are defined as specific models corresponding to the defined ontology model. In our approach, we consider that both ontology modelling concepts and ontologies are explicitly modelled.

We have used the Event-B method to formalize these concepts. More precisely, as we consider ontologies as theories, we have used Event-B contexts to formalize such concepts.

4 The OntoEventB plug In

The *OntoEventB* plug-in [10] has been developed to automatically support the translation of ontologies models, described using ontology description languages such as OWL, PLIB or RDFS, into Event-B Contexts [1]. It takes as input an ontology description file and generates, according to the selected approach (shallow or deep), the corresponding Event-B contexts. The *OntoEventB* plug-in is developed according to an architecture composed of three components: Input, Pivot and Output Models (Figure 1).

The Input Models component. This component is devoted to the processing of the input models described using different ontology description languages such as OWL, PLIB, RDFS ... It browses the

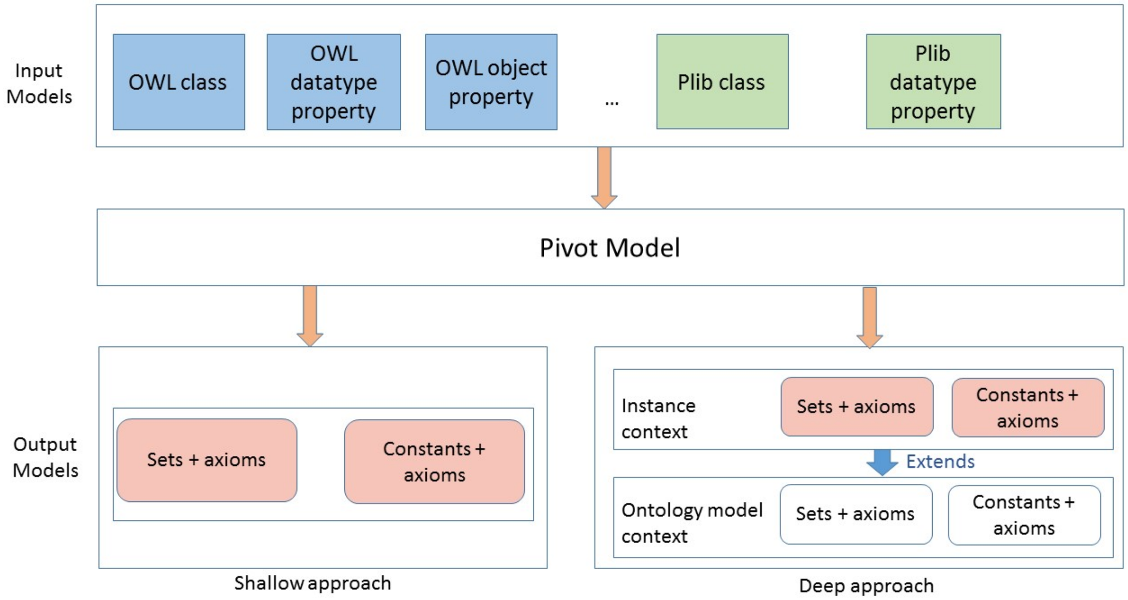

Figure 1: The OntoEventB internal architecture.

input models files in order to extract ontological concepts descriptions (e.g. OWL classes, OWL data type properties and OWL object properties in the case of OWL models) and to send them to the Pivot Model component.

The Pivot Model component. This component is an intermediate operational model, which summarizes the common relevant concepts used by ontology description languages (classes, properties and data types). It defines generic concepts that integrate all specific concepts that can be received from the Input Model component. The Pivot Model can be extended to integrate other generic concepts that can be identified if a new language is added as input model in the Input Models component.

When different ontological concepts are produced from Input Model components (e.g.OWL classes, OWL data type properties and OWL object properties in the case of OWL models), the Pivot Model component translates them into its generic concepts (classes, properties and data types). After this first translation step, the obtained generic concepts are ready to be treated by the next process handled by the Output Model component.

The Output Model component. This component has as input the generic concepts computed by the Pivot Model component and translates them into Event-B Context elements (sets, constants and axioms). This process uses transformation rules that formalise each ontological concept by an Event-B definition following the two approaches proposed and described in section 3 (Shallow and Deep modelling approaches). The user of the *OntoEventB* plug-in can choose one of them.

The use of this architecture allows us to extend the *OntoEventB* plug-in by taking into account new input ontology description languages without redefining the Event-B formalisation rules between Pivot Model component and Output Model component. Indeed, as soon as the new concepts defined by these new languages are translated into generic concepts of the Pivot model, they are be directly formalized in the Event-B Context elements without redefining new transformation rules.

Installing and Using OntoEventB plug-in. The *OntoEventB* tool is developed as an Eclipse plug-in to integrate it into a Rodin platform [2], an IDE (Integrated Development Environment) supporting Event-B developments. To use *OntoEventB* plug-in in your Rodin platform instance, you must install the plug-in by using the *Install New Software* menu item[1] for downloading and installing the plug-in automatically.

5 Conclusion

This paper reports on some of the results of the French ANR IMPEX research project. We have discussed the interest of making explicit domain knowledge in design models in order to strengthen them. We also proposed a straightforward approach formalizing ontologies as theories encoded within Event-B contexts. This approach led to the development of Plug-In that produces automatically Event-B contexts from ontologies expressed in different ontology models.

Moreover, the previous work achieved in this project showed the interests of the approach to strengthen models in different areas. We have applied the developed approach to case studies issued from avionic systems, medical devices and electronic voting systems.

This work is still an on-going work. We are currently investigating the possibility to formalise ontologies of behaviours (e.g. ontologies of services) and their use to annotate behavioural components of design models (e.g. events of an event-B model). First results are already available on plastic interfaces [5].

References

[1] Jean-Raymond Abrial (2010): *Modeling in Event-B - System and Software Engineering*. Cambridge University Press.

[2] Jean-Raymond Abrial, Michael J. Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta & Laurent Voisin (2010): *Rodin: an open toolset for modelling and reasoning in Event-B*. *STTT* 12(6), pp. 447–466.

[3] Yamine Aït Ameur, J. Paul Gibson & Dominique Méry (2014): *On Implicit and Explicit Semantics: Integration Issues in Proof-Based Development of Systems*. In: *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications - 6th International Symposium, ISoLA 2014, Imperial, Corfu, Greece, Proceedings, Part II*.

[4] Yamine Aït Ameur & Dominique Méry (2016): *Making explicit domain knowledge in formal system development*. *Sci. Comput. Program.* 121, pp. 100–127.

[5] Abdelkrim Chebieb & Yamine Aït Ameur (2015): *Formal Verification of Plastic User Interfaces Exploiting Domain Ontologies*. In: *2015 International Symposium on Theoretical Aspects of Software Engineering, TASE 2015, Nanjing, China, September 12-14, 2015*.

[6] Kahina Hacid & Yamine Aït Ameur (2016): *Annotation of Engineering Models by References to Domain Ontologies*. In: *Model and Data Engineering - 6th International Conference, MEDI 2016, Almería, Spain, September 21-23, 2016, Proceedings*, pp. 234–244.

[7] Kahina Hacid & Yamine Aït Ameur (2016): *Strengthening MDE and Formal Design Models by References to Domain Ontologies. A Model Annotation Based Approach*. In Tiziana Margaria & Bernhard Steffen, editors: *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th*

[1] OntoEventB update site : `http://wdi.supelec.fr/OntoEventB-update-site/`

International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I, *Lecture Notes in Computer Science* 9952, pp. 340–357.

[8] Kahina Hacid & Yamine Aït Ameur (2016): *Strengthening MDE and Formal Design Models by References to Domain Ontologies. A Model Annotation Based Approach*. In: *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I*, pp. 340–357.

[9] Linda Mohand-Oussaïd & Idir Aït-Sadoune (2017): *Formal Modelling of Domain Constraints in Event-B*. In: *Model and Data Engineering - 7th International Conference, MEDI 2017, Barcelona, Spain, October 4-6, 2017, Proceedings*, pp. 153–166.

[10] Linda Mohand Oussaïd & Idir Ait-Sadoune (2017): *OntoEventB : Un outil pour la modélisation des ontologies dans B Événementiel*. In: *AFADL 2017*, Montpellier, France, pp. 117–121.

[11] David Simon Zayas, Anne Monceaux & Yamine Aït Ameur (2010): *Knowledge Models to Reduce the Gap between Heterogeneous Models: Application to Aircraft Systems Engineering*. In Radu Calinescu, Richard F. Paige & Marta Z. Kwiatkowska, editors: *15th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS 2010, Oxford, United Kingdom, 22-26 March 2010*, IEEE Computer Society.