

Ontology Transformation of Enterprise Architecture Models

Marzieh Bakhshadeh¹, André Morais¹, Artur Caetano, and José Borbinha^{1,2}

¹Information Systems Group, INESC-ID, Rua Alves Redol 9, 1000-029 Lisboa, Portugal.

²Instituto Superior Técnico, Technical University of Lisbon, Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal
{marzieh.bakhshandeh, andre.coutinho, artur.caetano, jlb}@ist.utl.pt

Abstract. Enterprise architecture supports the analysis and design of business-oriented systems through the creation of complementary perspectives from multiple viewpoints over the business, information systems and technological infrastructure, enabling communication between stakeholders. However, enterprise architecture modelling languages lack representation schemas that support the computable assessment of its models. This paper applies model transformation to address this issue. The proposed approach translates models specified using ArchiMate into OWL. The resulting ontological representation is therefore computable, allowing for the analysis of the consistency and completeness of the enterprise architecture models. The applicability of the approach is shown through a case study.

Keywords: enterprise architecture, model transformation, ontology, ArchiMate, OWL.

1 Introduction

Enterprise architecture is defined by Lankhorst as [1] “a coherent whole of principles, methods, and models that are used in the design and realisation of an enterprise’s organisational structure, business processes, information systems, and infrastructure”. Enterprise architecture languages are usually specified without formal semantics or representation schemas that facilitate analysing its models. For instance, TOGAF is specified using a combination of textual descriptions and object-oriented models [2] and the BPMN metamodel [3] lacks a formal semantics that hinders checking the correctness of models [4]. The same issue occurs with the object-oriented metamodel of the ArchiMate language [5]. ArchiMate lacks a formal representation schema that could help to facilitate the analysing of its models.

There has been a growing interest in Ontology Engineering during the last decade [6]. The most widely used definition characterizes ontologies as “formal, explicit specification of a shared conceptualization” [7]. Ontologies are becoming more important, by their role on the so called Semantic Web [8] [9] [10]. In different fields such as software engineering [11] [12] [13]. The use of ontologies and associated

techniques in EA is increasing, with proposals of EA based on ontologies for improving the models and their semantics, as witnessed in [14] [15] [16] [17] [18] [19].

In the past few years a number of different approaches for evaluating EA models have been described. In [20], and [21], the authors have implemented a tool for analysing EA models, which guides the creation of enterprise information system scenarios in the form of enterprise architecture models and generates quantitative assessments of the scenarios as they evolve. In [22], the authors propose an uniform approach for capturing quality attribute requirements and analysing system and software architecture. However, such approach did not considered the business architecture [23].

This paper presents an application of model transformation to translate the standard ArchiMate enterprise architecture language into the OWL ontology language [24]. OWL is the latest and the most complex ontology language presented by W3C. With OWL ArchiMate metamodel can be represented with ontologies. Ontologies can assist architects by depicting all the consequences of their model. Formal ontology machineries also help architects to view and understand the implicit consequences of explicit statements and can help to ensure that a model is consistent. Logical reasoning can be applied to check the inconsistencies on the models and for inferring different dependencies between different elements of enterprise architecture models.

In this sense, we try to contribute to the following:

- Given Archimate metamodel described in a single XML document. How can we transform XML to OWL;
- Define a set of techniques on how to specify enterprise architecture models in order to improved meta-model conformance verification of models, through the verification of logical inconsistencies present in models;

The paper is organized as follows: in Section 2 the relationship of our topic to collective awareness systems is described. Section 3, the ArchiMate meta-model is introduced and the transformation process of the ArchiMate metamodel and ArchiMate models are presented. After that, in Section 4, a validation based on reasoning using a predefined set of competency questions on a specific use case has been performed. Finally, in Section 5 we conclude remarks drawn from the work and our perspectives for further research.

2 Collective Awareness Systems

Collective awareness is a critical part of collaboration within communities; especially computer-mediated communities [25] in our case, collective awareness can be achieved by analyzing the stakeholders view points, as such, enterprise architects need to conceive views from the viewpoint of the stakeholders and to address their concerns and requirements. However, the existence of semantic gaps between architects and stakeholders may produce conceptual misalignments which can negatively affect the architecture.

3 Transformation Process

3.1 ArchiMate

ArchiMate is an open and independent modelling language, from the Open Group, for enterprise architecture that is supported by different tool vendors and consulting firms [5].

The ArchiMate framework organizes its metamodel in a three by three matrix: the rows capture the enterprise domain layers (business, application, and technology), and the columns capture cross layer aspects (active structure, behaviour and passive structure).

3.2 Transformation of the ArchiMate Metamodel to OWL

The transformation process uses 1) an OWL representation of the ArchiMate metamodel and 2) OWL representations of ArchiMate models.

A set of transformation rules maps each element from the ArchiMate metamodel into the corresponding OWL representation. Each concept is transformed into an OWL Class. Relationships are transformed as OWL Object Properties. ArchiMate metamodel are described in a single XML document represented in Listing 1. Table 1 shows the Transformation Rules for the ArchiMate metamodel. In Table 2 the Archi XML representation element and respective OWL Class is depicted, next in Table 3 Archi XML representation element relations and respective OWL ObjectProperties is shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ArchiMate 2.0 relationship rules -->
<relationships version="2.0">
  <elements>
    <source element="BusinessActor">
      <target element="BusinessActor" relations="cfgostu"
/>
      <target element="BusinessRole" relations="fiotu" />
      .
      .
      .
      <target element="WorkPackage" relations="o" />
      <target element="Deliverable" relations="o" />
      <target element="Plateau" relations="o" />
      <target element="Gap" relations="cgos" />
    </source>
  </elements>
</relationships>
```

Listing 1. XML File structure

Table 1. Transformation Rules for the ArchiMate metamodel

XML	OWL
source element	Class
target element	Class
Relationships	object properties

Table 2. Archi XML representation element and respective OWL Class

XML	OWL
<source element="BusinessActor">	<Declaration> <Class IRI="#BusinessActor"/> </Declaration>

Table 3. Archi XML representation element relations and respective OWL ObjectProperties

XML	OWL
<source element="BusinessActor"> <target element="BusinessObject" relations="ao"/> <target element="Contract" relations="ao" > </source>	<Class IRI="#BusinessActor"/> <ObjectAllValuesFrom> <ObjectProperty IRI="#accesses"/> <ObjectUnionOf> <Class IRI="#BusinessObject"> <Class IRI="#Contract"/> </ObjectUnionOf> </ObjectAllValuesFrom>

After the concepts and relationships are represented as classes and object properties, the ontology constraints still need to be included. Restrictions were added to the properties: Inverse ObjectProperties and SuperObjectProperties axioms were added to the OWL ontology, so that derived relationships can be extracted through the use of reasoners. Moreover, two SuperObjectProperty chains were created for modeling dependencies between different elements. The dependsDown ObjectProperty is thus a SuperObjectProperty of the aggregation, composition, assignment, usage, and realization ObjectProperties that resulted from the conversion of the ArchiMate relations, while the dependsUp ObjectProperty fills the same purpose for the counterpart InverseObjectProperties.

3.3 Transformation of ArchiMate Models

The ArchiMate models, i.e. the instances of the metamodel, are converted to OWL using a converter that converts a CSV file, generated using the Archi tool [26] plug-in that export such file, in representation of an ArchiMate model to OWL. Figure 1 shows an OWL representation of the ArchiMate metamodel, along with classes and object properties.

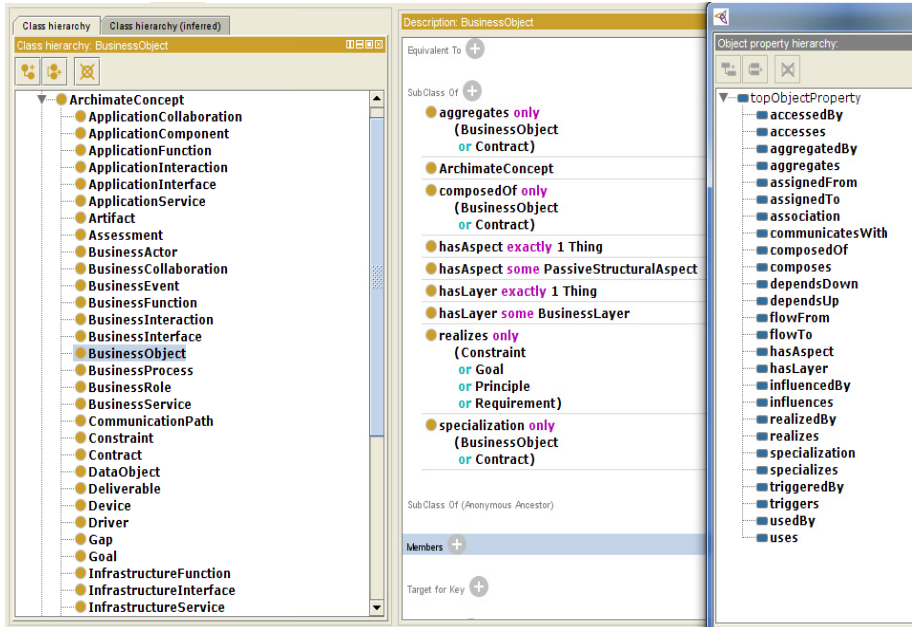


Fig. 1. Business Function class and corresponding properties

3.4 Reasoning

OWL language is based on Description Logics (DL), which is one of the main representational facilities of OWL. OWL-DL enables taking advantage of existing inference and querying mechanisms to analyse the models and assessing their consistency. A Description Logic reasoner performs various inferring services [27].

In the case of ArchiMate, it is possible to take advantage of reasoning in order to check the inconsistencies on the models and for inferring different dependencies between different elements of enterprise models.

4 Case Study

ArchiSurance is a fictitious insurance company used throughout the ArchiMate 2.0 as a case. The scenario was converted to the OWL to be represented as the instances of the ArchiMate metamodel; this example in this section is used to illustrate the capabilities of reasoning, by validating the correctness of the ontology. A set of predefined competency questions were used in order to validate ontology [28]. One of the competency questions defined to validate the integrated ontology is :

- What Services belongs to a Technology Layer are Behavioural Aspect and is used by Financial Application?

The representation of the last competency question is showed in Figure 2.

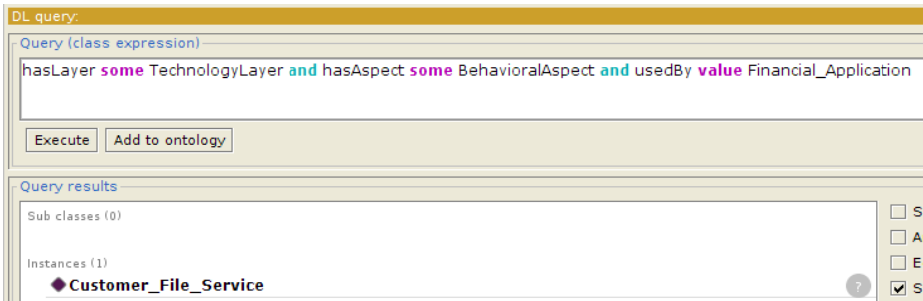


Fig. 2. Query Result

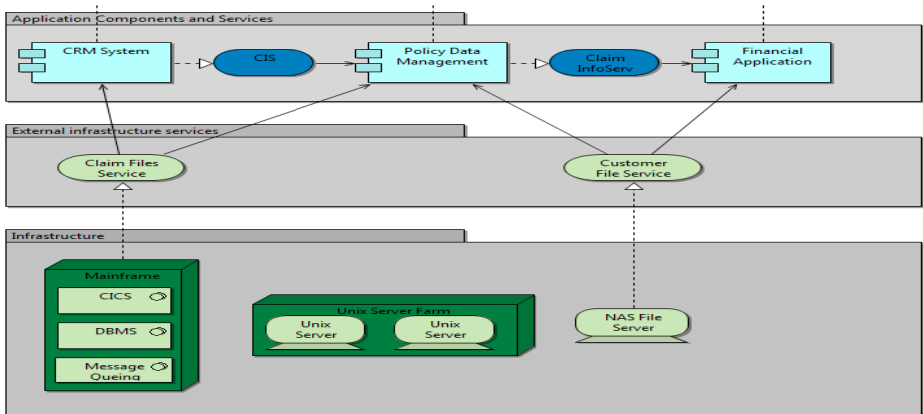


Fig. 3. ArchiSurance Layered View

The model below in Fig. 3 depicts a high-level view of the application components and infrastructure in the ArchiSurance scenario.

5 Conclusions and Future Work

This paper presents an application of model transformation to translate models described with the standard ArchiMate enterprise architecture language into the OWL ontology language. The resulting artefact is an ArchiMate ontology. This ontology was validated by a set of competency question in a use case. The OWL language gives a formal representation of the ArchiMate metamodel and enables reasoning to be applied to enterprise architecture models. As it was proven, ontologies assist to cover the semantic behind enterprise architecture concepts. In this sense, we presented some techniques that can be used for specification and analysis of models, allowing a better understanding of them. Future works will focus on the application of this approach to new scenarios in order to discover the analysis possibilities, considering the usage of different reasoning and querying techniques.

References

1. Lankhorst, M.: Enterprise architecture at work: Modelling, communication and analysis. Springer (2013)
2. Group, T.: TOGAF(2009)
3. Allweyer, T.: BPMN 2.0. BoD (2010)
4. Ye, J., Sun, S., Song, W., Wen, L.: Formal Semantics of BPMN Process Models Using YAML. In: Intelligent Information Technology Application, pp. 70–74 (2008)
5. Group, T.: Archimate 2.0 Specification (2012)
6. Jenz, D.: Business Process Ontologies: Speeding up Business Process Implementation (2003)
7. Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisitions 5, 199–220 (1993)
8. Berners-Lee, T.: The semantic web. Scientific American (2001)
9. Shadbolt, N.: The semantic web revisited (2006)
10. Hitzler, P.: Foundations of semantic web technologies. Chapman and Hall/CRC (2011)
11. Gašević, D.: Model driven architecture and ontology development. Springer (2006)
12. Pan, J.: Ontology-Driven Software Development. Springer (2013)
13. Happel, H.-J.: Applications of ontologies in software engineering. In: Proc. of Workshop on Semantic Web Enabled Software Engineering (2006)
14. Uschold, M.: The enterprise ontology (1998)
15. Geerts, G.: An ontological analysis of the economic primitives of the extended-REA enterprise information architecture (2002)
16. Kang, D.: An ontology-based enterprise architecture (2010)
17. Wagner, G.: Ontologies and Rules for Enterprise Modeling and Simulation (2011)
18. Azevedo, C., Almeida, J., Sinderen, M., Quartel, D., Guizzardi, G.: An ontology-based semantics for the motivation extension to archimate. In: 15th IEEE International Enterprise Distributed Object Computing Conference, pp. 25–34 (2011)
19. Almeida, J., Guizzardi, G.: An ontological analysis of the notion of community in the RM-ODP enterprise language. Computer Standards & Interfaces, 257–268 (2013)
20. Johnson, P., E.: A tool for enterprise architecture analysis (2007)
21. Buschle, M., J.: A tool for enterprise architecture analysis using the prm formalism (2011)

22. SoS, A.: A uniform approach for system of systems architecture (2009)
23. Gagliardi, J.: A workshop on analysis and evaluation (2010)
24. McGuinness, D., van Harmelen, F.: OWL Web Language Overview. W3CRecommendation 10, 1–19 (2004)
25. Favier, M.: Developing a Measure of Collective Awareness in Virtual Teams (2007)
26. Archi - Archimate Modelling. In: Overview, <http://archi.cetis.ac.uk> (accessed December 05, 2012)
27. Protégé. In: what is protégé?, <http://protege.stanford.edu/> (accessed December 05, 2012)
28. Fox, M., Gruninger, M.: Enterprise Modeling (1998)