Using Ontologies for Enterprise Architecture
Model Alignment

Gonçalo Antunes¹, Artur Caetano¹,², Marzieh Bakhshandeh¹, Rudolf Mayer³, and José Borbinha¹,²

¹ Instituto Superior Técnico, University of Lisbon,
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal,
² Information Systems Group, INESC-ID,
Rua Alves Redol 9, 1000-029 Lisboa, Portugal,
³ SBA Research, Favoritenstraße 16, 1040 Wien, Austria
{goncalo.antunes,artur.caetano}@ist.utl.pt

Abstract. One of the primary goals of enterprise architecture aligning
the business with the underlying support systems. An architecture de-
scription encompasses an heterogeneous spectrum of domains, such as
business processes, application components, metrics, people and tech-
nological infrastructure. Views express the domain elements and their
relationships from the perspective of specific concerns relevant to the
system stakeholders. Thus, each view needs to be expressed in the de-
scription language that best suits its concerns. However, enterprise ar-
chitecture languages often specify meta-models that cross-cut distinct
architectural domains. This hinders extensibility and adds complexity
to the language. Ob the other hand, describing each domain through a
specialized language and then integrating the multiple languages raises
challenges at the level of traceability and consistency. This paper pro-
poses using ontologies to integrate different enterprise architecture do-
mains and to analyse the resulting models. This goal is realized through
a core domain-independent language that is extended by several domain-
specific languages, each focussing on a set of specific concerns. The ap-
proach contributes to the alignment of the different domains while en-
suring traceability and model consistency. The proposal is demonstrated
through an evaluation scenario that employs ArchiMate as the domain-
independent language extended with a set of domain-specific languages.
The demonstration shows that the different domains can be consistently
aligned and analysed through the use of ontologies.

Key words: enterprise architecture, alignment, separation of concerns,
ArchiMate, ontology, OWL

1 Introduction

Enterprise architecture (EA) is defined by Lankhorst as “a coherent whole of
principles, methods, and models that are used in the design and realization
of an enterprise’s organizational structure, business processes, information sys-
tems, and infrastructure” whose models “focus on alleviating the infamous business–information technology alignment problem” [1]. Alignment results from applying models, methods, patterns and best practices to the specification and governance of the different domains of an organisation [2, 3, 4, 5]. Managing dependencies is fundamental for supporting the communication between the different stakeholders and to maintain the consistency at model and meta-model level [6, 7]. Moreover, EA governance requires the ability to analyse artefacts [1, 8] and it is also required to assist business analytics [9].

Despite the efforts for developing comprehensive approaches to architecture, such as TOGAF [10], a “one model fits all” is unable to address the specific domains of an organisation [11, 12]. Different organizations have specific needs and thus place particular demands on the EA models’ artefacts. As such, the development of an architecture description language entails ensuring the consistency and traceability between the language concepts [13, 1]. On the other hand, creating a consistent and comprehensive architecture description language that deals with specific domains is a challenge despite existing situational method approaches to enterprise architecture management [14].

According to the ISO 42010 standard, the usage of multiple views is fundamental to describe a system’s architecture. A system has multiple stakeholders, each with specific interests on the system. An architecture description should aggregate multiple views, materialized in a set of models, that are formulated according to viewpoints expressing the concerns of the stakeholders of the system-of-interest [15]. In this way, an architecture works as a communication agent between stakeholders, as each is presented with its own view over the system of interest. Creating viewpoints may actually require using different meta-models, tools, and validation mechanisms. This creates hurdles to consistency checking and traceability. The integration and extension of models and underlying meta-models is common [16, 17]. But such approaches bring challenges at the level of traceability and consistency because it is difficult to trace concepts between different domains, a problem that is aggravated as the models evolve [18]. Moreover, the integration of different meta-models poses challenges to its validation [19].

This paper is concerned with the question of how to enhance enterprise architecture modelling with regards to consistency and traceability. In this context, consistency means the quality of the artefacts being logically related. The goal is to integrate multiple enterprise architecture description languages as a means to assist alignment. Specifically, we investigate whether ontology technologies can be used to specify, integrate and analyse multiple enterprise architecture models and the underlying meta-models.

Ontologies describe a domain model by associating meaning to its terms and relations. A more formal and widely used definition is that of Gröber who defines an ontology as a “formal specification of a conceptualisation” [20]. The importance of this technology is evidenced by the growing use of ontologies in a variety of application areas [21, 22] and, especially, by their role on the Semantic Web initiative [23, 24]. Ontology technologies are also used in the field of enter-
Enterprise Architecture Model Alignment

Enterprise architecture to formalize organizational artefacts and to assist with model analysis [25, 26, 27, 28, 29, 30]. In fact, there is a wide body of knowledge that may improve the practice of EA, including ontology matching [31], and model extension and validation [32]. Ontologies facilitate the construction of complex models and can assist model analysis by depicting the consequences of a model. Formal ontology technologies also contribute to viewing and understanding the implicit consequences of explicit statements and can help ensuring that a model is consistent [33].

This paper posits that modelling the different enterprise architecture domains with a set of integrated description languages contributes to their alignment because consistency and traceability become ascertained. The approach entails using ontologies to represent and integrate the multiple architecture description languages and to analyse the resulting models. We argue for the integration of formal ontologies and associated technologies as mechanisms for developing consistent enterprise architecture models. The combination of formally specified models with their analysis via automatic mechanisms contributes to aligning the heterogeneous domains of an EA. One example is the impact analysis of changes from the business on the IT infrastructure and vice-versa. The main contribution of this paper is thus proposing an architecture based on the use of ontologies with the purpose of enhancing the extensibility with domain-specific aspects while enforcing consistency. We demonstrate the applicability of the proposal through the application of formal ontologies to model a set of different EA domains and through the consistent integration of these domains. In particular, we develop an ontology to specify the ArchiMate 2.0 meta-model and then create traceable maps to it from a set of domain-specific languages. We also describe an example that maps the sensor technology domain to ArchiMate in the context of a real-world scenario. This demonstration shows that the application of ontologies to enterprise architecture modelling effectively assists consistently aligning and analysing different domains.

The rest of this paper is organized as follows: section 2 describes a ontology-based proposal to integrate and analyse enterprise architecture models; section 3.1 describes the realization of the proposal; section 3 evaluates the solution using a scenario; finally, section 4 concludes the paper.

2 Using Ontologies to Integrate Enterprise Architecture Meta-Models

This paper proposes an ontology-based framework to formalize and integrate different domains of an enterprise architecture. The design of this artefact adheres to the following architectural principles:

- **Concern orientation.** The architecture represents the concepts that address an explicit set of concerns as a meta-model. The meta-model will not support any concepts that are not derived from the stakeholders' concerns.
- **Viewpoint-orientation.** The architecture supports defining views over sub-sets of its concepts. This facilitates communication because viewpoints act as a separation of concerns mechanism. Viewpoints facilitate addressing multiple concerns and can improve decision-making by isolating certain aspects of the architecture according to the needs of stakeholders.

- **Expressiveness.** The architecture represents a set of unambiguous domain concepts. This entails defining the minimum set of consistent types and relationships to describe the domain.

- **Extensibility.** The architecture supports the integration of multiple domain-specific and domain-independent meta-models while minimizing coupling.

- **Modularity.** The architecture observes high-cohesion and low-coupling. Observing these qualities contributes to the expressiveness and extensibility of the architecture with the goal of minimizing the impact of adding new domain-specific concepts.

The framework uses a core meta-model to formalize the concepts that are domain-independent. This meta-model is formalized as an upper-level ontology and is designated as a *domain-independent ontology* (DIO). The design goal of the DIO is to identify a minimum set of concepts pertaining to the central modeling domain (in this case, enterprise architecture). The concepts of the DIO are extended by defining a variable number of domain-specific meta-models, depending on the particular system concern. Each domain-specific meta-model is formalized as a *domain-specific ontology* (DSO). Thus, a DSO represents a domain-specific language that addresses a particular set of concerns, and should also have the minimum set of concepts required for describing a determined domain. Therefore, separation of concerns, low-coupling and high-cohesion are the primary qualities that affect the DIO and DSO design.

Ontology integration is required to link concepts from the DIO to the DSO. Integration combines different ontologies while ensuring consistency and maximum coverage of the domain being addressed. The simplest case is that of integrating the DSO concepts with the core concepts represented in the DIO. Cross-DSO integration occurs whenever more expressiveness power to specific domains. The ontology integration makes use of model transformation, which involves defining a mapping strategy from a source model to a target model. Figure 1 depicts the types of transformation maps between the DIO and the DSOs. A map attempts to relate the concepts from a source to a destination ontology so that there is a one-to-one correspondence between each pair of concepts. Three types of mapping deficiencies may occur. A source concept may map to more than one destination concept resulting in overload, a source concept may not be mappable to any destination resulting in deficit, or several source concepts may map to the same destination concept leading to redundancy.

The models that result from instantiating the integrated DIO and DSO can be analysed and properties and relations can be inferred through reasoners. Four types of reasoning are possible with this architecture: DIO reasoning when inference is limited to the DIO concepts, DSO reasoning, when inference is limited
Fig. 1: Mappings between domain-independent and domain-specific ontologies to the concepts of a single DSO. Cross-DSO reasoning, when inference use concepts from different mapped DSOs, and DIO-DSO reasoning, when inference uses concepts from the DIO and one or more DSOs.

3 Application to Enterprise Architecture

This section describes the specification of a DIO, a DSO and an application of the integrated meta-models. The evaluation scenario concerns a civil engineering safety authority that needs to monitor large structures such as hydroelectric power dams, reservoirs and bridges. Each structure has different sensors that measure physical phenomena and produce data that is analysed to assess the structure throughout its life cycle. The business process that deals with the assessment of a structure includes activities that deal with the data acquisition and data analysis. An instance of this process may be active for decades, from the early construction phases until disposition. Part of this process is supported by an information system that provides the following functions:

- Instrumentation: manages sensor installation, configuration and deployment.
- Transformation: manages the algorithms that transform sensor raw data into information.
- Observation: manages geodetic data, visual inspections data, and the data acquired from monitoring systems.
- Analysis: manages data analysis, visualization and reporting.
- Synchronization: synchronizes data between multiple locations and systems.

The organization is required to acquire and preserve the monitoring data during the entire structure’s life cycle. Therefore, capturing information about the acquisition processes and supporting technological infrastructure is fundamental to attest the provenance and authenticity of the monitoring data. Moreover, historic data can be used to analyse and predict the behaviour of the structure. In this setting, enterprise architecture plays a valuable role to assist with the specification, evolution and the alignment of these processes with the supporting technology.

This scenario was modelled with the ArchiMate 2.0 language. Although ArchiMate is able to specify the different domains of this scenario, at a high-level of abstraction, it has not the expressiveness required to model the domain-specific concerns pertaining to sensors and acquisition. The first step was to
create an ArchiMate model of the scenario (the models were produced with the Archi tool\(^1\)). Figure 2 depicts an overview of the acquisition process and the services supporting it. Figure 3 depicts an overview of the application components and underlying technological infrastructure. The ArchiMate model was then exported from the Archi tool and automatically converted to OWL using a specialized tool.

Fig. 2: Business processes and application infrastructure

Fig. 3: Technological infrastructure

3.1 The ArchiMate Domain-Independent Ontology

ArchiMate describes the domain-independent aspects of the architecture, focusing on the core concepts pertaining to enterprise architecture. The DIO is

\(^1\) http://archi.cetis.ac.uk/
therefore a specification of the ArchiMate meta-model using OWL-DL. OWL-DL enables taking advantage of existing inference and querying mechanisms to analyse the models and assessing their consistency. The ArchiMate ontology was mainly developed according to the ontology engineering methodology defined by Horridge [37]. The steps include:

1. Identification of the concepts and concept hierarchy.
2. Identification of the disjoint concepts.
4. Addition of all the relationships between concepts.
5. Identification of definitions.
6. Addition of annotations.
7. Refinement of the ontology through various iterations of the above steps.

The resulting ontology maps the ArchiMate concepts to OWL Classes and relations to OWL ObjectProperties. Restrictions were added to the properties, such as InverseObjectProperties and SuperObjectProperties, so that derived relationships can be correctly inferred. Figure 4 depicts a partial specification of ArchiMate’s business function as displayed in Protégé 4.3.

![Fig. 4: Partial OWL-DL specification of an ArchiMate Business Function](image)

### 3.2 The Domain-Specific Ontology

The organizational stakeholders required modelling and analysing specific information about sensors. However, the ArchiMate language lacks the expressiveness to capture the specifics of this domain. Sensors measure values that can be processed and used in structural analysis. There are sensors for making different measurements, which have specific transformation algorithms and calibration.
Fig. 5: Core structure of the sensor DSO

values. Some sensors are geo-referenced and some capture data according to dynamic acquisition rates. As such, the particularities of this domain imply defining a DSO. The organization considered different sensor modelling languages, such as SensorML and TransducerML but decided that their expressiveness was insufficient to address specific concerns. As such, we developed a DSO based on SensorML that addresses the concerns of this organization using the ontology engineering methodology described earlier. The core concepts of the sensor DSO are depicted in figure [5]. The transformation map from the sensor DSO to the ArchiMate DIO contains the relations described on Table 1.

### Table 1: Mappings between the Sensor DSO and the ArchiMate DIO

<table>
<thead>
<tr>
<th>Sensor DSO</th>
<th>ArchiMate DIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor</td>
<td>Node</td>
</tr>
<tr>
<td>GeoLocation</td>
<td>Location</td>
</tr>
<tr>
<td>StructuralLocation</td>
<td>Location</td>
</tr>
<tr>
<td>Algorithm</td>
<td>ApplicationComponent</td>
</tr>
<tr>
<td>Value</td>
<td>Data Object</td>
</tr>
<tr>
<td>AcquisitionRatePerYear</td>
<td>Data Object</td>
</tr>
</tbody>
</table>

### 3.3 Model Analysis

One of the stakeholder concerns relates to the technological infrastructure elements that support the acquisition process. This concern can be addressed

---

2 [http://www.ogcnetwork.net/SensorML](http://www.ogcnetwork.net/SensorML)

3 [http://www.ogcnetwork.net/infomodels/tml](http://www.ogcnetwork.net/infomodels/tml)
through DIO reasoning, i.e. via the ArchiMate meta-model. Figure 6 depicts the question formalized as a OWL-DL query along with the ObjectProperty chains that identify the 19 instances that support the acquisition process. An example of intra-DSO reasoning is depicted on figure 7 that depicts the sensors that are able to make temporal readings. Finally, figure 8 shows the result of DIO-DSO reasoning where the integrated models are queried about which ArchiMate ApplicationComponents rely on the reading of the sensors of type Drain. In the latter case, the reasoner uses the mappings between the Sensor DSO and the ArchiMate DIO to infer the reasoning chains and thus to answer the query.

Fig. 6: Intra-DIO query results

Fig. 7: Intra-DSO query results

Fig. 8: Cross DIO-DSO query results
4 Conclusions

This paper proposes using ontologies to integrate different enterprise architecture domains and to analyse the resulting models. This goal is realized through the specification of a core domain-independent ontology that is extended by a set of domain-specific ontologies, each focussing on specific concerns. The approach contributes to the alignment of the different domains while ensuring traceability, consistency and extensibility. As observed from the case study, formal ontologies can enhance the quality of meta-modelling due to their automated analysis capability that can be used to assess meta-model consistency as well as to assess model conformance. Moreover, ontologies positively contribute to enterprise architecture alignment because multiple meta-models can be integrated and represented in such a way that its information can be traced and analysed and the reasoning consequences be exposed. The proposal was evaluated using ArchiMate as the DIO. To do that, we converted the ArchiMate meta-model to OWL-DL. A scenario was modelled using the ArchiMate DIO and its domain-specific aspects were modelled using a set of DSOs. In particular, this paper partially described one of the DSOs, the Sensor DSO, and exemplified different types of analysis that can be accomplished using this approach. This demonstration shows that the application of ontologies to enterprise architecture modelling effectively assists consistently aligning and analysing different domains.

Our current work focuses on extending the analysis capabilities to support the validation of models and the assessment of models and meta-models. We are also working on a set of automated and semi-automated extractor and process mining tools to instantiate the domain-specific and domain-independent ontologies with operational data to test the conformance of the “should-be” models towards the actual “as-is” models.

Acknowledgements

This project is partially supported by the European Commission under the 7th Framework Programme (FP7/2007-2013) under grant agreement 269940, TIMBUS project (http://timbusproject.net) and by COMET K1, FFG, Austrian Research Promotion Agency. Gonçalo Antunes is partially supported by FCT, Fundação para a Ciência e a Tecnologia, through project PEst-OE/EEI/LA0021/2011 and grant SFRH/BD/69121/2010.

References


