

# Ontology-based Validation of Enterprise Architecture Principles in Enterprise Models

Devid Montecchiari<sup>[0000-0002-8969-1973]</sup>

FHNW University of Applied Sciences and Arts Northwestern Switzerland, School of Business, Switzerland  
UNICAM University of Camerino, School of Advanced Studies, Italy  
[devid.montecchiari@fhnw.ch](mailto:devid.montecchiari@fhnw.ch)

**Abstract.** Enterprises use Enterprise Architecture Principles as a guiding set of rules to provide a basis for decision making. These principles are described using natural language and are not machine-interpretable. The validation of these principles in models is a complex and time-consuming task. The goal of this research is to help humans in this review. Annotating enterprise architecture models with an enterprise ontology and representing architecture principles as rules, it is possible to automatically check architecture principles. The proposed approach is to combine both the domain knowledge and the modeling language knowledge to reason about models, allowing the automatic check of architecture principles.

**Keywords:** Enterprise Ontology · Enterprise Architecture · Enterprise Architecture Principle · Business Rule · Shacl.

## 1 Introduction

The Open Group Architecture Framework (TOGAF) [5] lists a variety of ways in which companies use Enterprise Architecture Principles (EAPs) e.g. *"As drivers for defining the functional requirements of the architecture"*. It also proposes a recommendation for their template and a definition of their characteristics, classifications. The declaration and validation of EAP is a complex manual task. EAPs affect each other and may have hierarchies and ramifications, leading to conflicts and inconsistencies also with the modeled enterprise architecture.

ArchiMate [11] is the OpenGroup standard, adopted by several modeling tools and consulting firms, for the representation of an enterprise architecture. ArchiMate models can be semantically annotated, enriching them with machine-interpretable knowledge [8,10]. Instead, EAPs are defined in statements using natural language. Thus, currently, it is not possible to automatically validate whether the enterprise models of a company are compliant with a specific set of principles.

The proposal is to decompose EAPs into a set of machine-interpretable rules (in an enterprise ontology) that are reusable, verifiable, and measurable. The rules should combine both the domain knowledge (e.g. derived from enterprise architects and literature) and the modeling language knowledge (i.e. syntax and

semantics of ArchiMate [11]). Using this set of rules in the ArchiMEO ontology, it can be possible to recognize EAPs inconsistencies in enterprise models. Further, having the EAPs in the ontology would foster their re-usability and interoperability.

## 2 Research Methodology

The Design Science Research (DSR) framework [7,24] was chosen as it supports the development of artefacts with the aim of better understanding or improving a theory. The artefact is a set of rules representing EAPs in the enterprise ontology ArchiMEO [9] - that is currently being developed - combining both the domain knowledge and the modelling language knowledge.

The DSR consists of five research phases: awareness of the problem, suggestion, development, evaluation, and conclusion. Given the main research question, “*How can we create a set of rules based on EAPs to check the alignment of enterprise models?*” different sub-research questions were identified and are provided in the following chapters according to different DSR phases.

## 3 State of the Art

There exist different studies on models machine interpretability, usually focusing on the Business Process Management Notation like in [3]. Hence, the state of the art was investigated to answer the following research question in the awareness phase of the DSR.

*RQ1: What knowledge is currently machine-interpretable from enterprise architecture models and architecture principles?*

The Enterprise Engineering Knowledge Space [12] was coined for Knowledge Engineering in Business Process and presented with its four dimensions form, content, interpretation, and use. It is possible to apply this framework also to ArchiMate models.

Form: Syntax and semantic of the ArchiMate modeling language. Content: Domain in which knowledge engineering is applied. Use: Stakeholders and their concerns determine the relevant subset of the knowledge and reasoning Interpretation: Graphical models typically are cognitively more adequate for human interpretation and ontologies can be interpreted by machines.

ArchiMEO [9] is a standardized enterprise ontology based on the ArchiMate [11] conceptual model for the creation of Intelligent Information Systems. It is based on a semantically enriched Enterprise Architecture Description (seEAD) composed of four parts: a top-level ontology containing generic concepts as *time, location and space*; application-specific ontologies; a meta enterprise ontology according to the ArchiMate standard and an enterprise upper ontology, enriching the standard. Using ArchiMEO [9] as knowledge base, the Agile Ontology Aided

Modeling Environment (AOAME) [15] can model different modeling languages among which ArchiMate.

Answering the RQ1, a model in ArchiMEO can be semantically annotated with domain knowledge while keeping its syntax in a machine-interpretable form. In AOAME, a modeler can manually associate a modeled element to concepts available in the seEAD or outside ArchiMEO.

### 3.1 Semantic annotation of EAP and enterprise models

Following the awareness phase of the DSR, the outcome of the suggestion phase is a tentative design on how the EAPs can be made machine-interpretable. My proposal is to integrate the EAPs in ArchiMEO, decomposing them in a set of rules. In this phase, I address and describe the following two research questions.

*RQ2: How can enterprise architecture models be semantically annotated?*

The Open Group also defines an ArchiMate Model Exchange File Format Standard "XML Schema Files (.xsd) [6] for ArchiMate 3.1" that "can be used to exchange data between tools and/or systems that wish to import, and export ArchiMate models". Following this schema specification, I created a JAVA parser that automatically generates .ttl files according to the ArchiMEO ontology based on the Archimate .xml files.

The Open Group provides interoperability testing snippets used to validate import and the export functionalities among different modeling tools<sup>1</sup>. I used these interoperability snippets to validate the ArchiMate ontology contained in ArchiMEO and the parser results. During the development of the parser, I also updated the ArchiMate ontology (ca. 350 triples were updated/added) according to the 3.1 specifications of the standard. New ArchiMate layers were added by the Open Group since the previous version of the standard and consequently, several concepts and relationships were updated, made deprecated, and added. Each of the .ttl files generated by the parser was then also tested on the Apache Jena Fuseki server.

Answering the RQ2, the knowledge contained in ArchiMate models can be made available at least in the following two ways: manually-semantically annotated in AOAME, allowing the modeler to associate a meaning contained in the ArchiMEO ontology with the drawback of time-effort; automatically with the risk of imprecision e.g. due to the technical limitations of the parser.

*RQ3: How can enterprise architecture models be semantically annotated?*

There is a wide literature about natural language processing and understanding, however, the goal of this research is not to automatically create rules from natural language. Instead, the research goal is to formalize the principles in form of a set of rules allowing an automatic reasoning process.

---

<sup>1</sup> [https://www.opengroup.org//xsd/archimate/3.1/examples/\\_Snippets/](https://www.opengroup.org//xsd/archimate/3.1/examples/_Snippets/)

A recent survey [1] on 27 years of EAPs literature, strengthened the definition, characteristics, and "key role in guiding the design and the implementation of information system requirement". They state "An architecture principle is a declarative statement, based on, at least, business and IT strategy. It normatively describes a property of the design of an information system, which is necessary to ensure that the information system meets its essential requirements."

Essentially, EAPs are statements and an association to business rules [21] can be made, answering the RQ3. The automation of business rules is widely covered in the literature and different authors have created machine-interpretable business rules. Notably, in [19], it is proposed an "ontology-based approach for detecting the potential semantic error of business process and business rules". In other cases, business rules were written in Semantics of Business Vocabulary and Business Rules(SBVR) [23] and mapped to the Web Ontology Language (OWL) [17] with an automatable and structural-rooted approach [13,20].

## 4 Creation of machine interpretable EAPs

Different semantic web languages can be used to model ontologies. In the development phase, the following (technical) research questions is tackled.

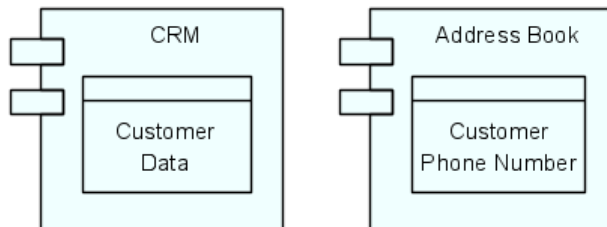
*RQ4: which semantic web language is the most appropriate to model principles as rules in ArchiMEO?*

In [16], we have described how AOAME can be used to support innovation processes, specifically for Design Thinking via Sap Scenes, and introduced a proof of concept to use the Shapes constraint language (SHACL) [14] for plausibility and constraint checking. In a different domain, SHACL NodeShapes and rules (`sh:rule`) were used to constrain the ontology and infer new knowledge utilized in the dialog management [2]. The use of SHACL has proven to allow: generation of validation reports; embedding of rule/constraints written in SPARQL [18] and/or JavaScript [4]; closed world assumption on demand [22].

### 4.1 Proof of concept implementation

The proof of concept implementations uses a fictional ArchiMate model and check if it has a Master Data Management System (MDMS) in place. A snippet of the model is shown below in fig. 1 (simplified for clarity).

The EAP "For all data, there is exactly one application that stores and manages the master version of the data." is a good candidate to explain the need to understand modeling element "content" (according to the definition in [12]). As humans, we can recognize that the *Customer Data* and *Customer Phone Number data objects* may represent duplicated information. A rule, checking if every *data object* in an ArchiMate model is *written* by exactly one application, would fail to recognize the potential inconsistency with the EAP. This, because the Customer Phone Number may be contained also in the *Customer data* within the *CRM*.



**Fig. 1.** Example snippet of an ArchiMate model

The system needs to know if there are at least two *data object* containing/about the same data and more precisely if these data are the master version.

First, the ArchiMate models are parsed using the developed Java ArchiMate parser, and every modeled element are instantiated in a turtle file according to concepts described in the ArchiMEO ontology. Based on the label, in this case, the parser recognizes the keyword "customer" and automatically annotates that this data object is concerning customers.

```

mod:Customer_Data
  rdf:type archi:DataObject ;
  do:dataObjectConcerning do:Customer ;
  rdfs:label "Customer Data" ;

```

Hence, the ArchiMate diagram .xml file gets transformed into a turtle file (.ttl) containing the list of elements, the relationships. This step can also be avoided using AOAME [15] as it allows the semantic annotation of enterprise models.

Second, the created turtle file is loaded in an Apache Jena Fuseki server with the ArchiMEO ontology and the dedicated set of rules representing the MDM EAP. Finally, executing the SHACL reasoner in the triple-store, we can get a validation report containing the list of infringements. Aside from the validation report, we can also get the result of rules that inference new knowledge and perform automatic classifications.

## 5 Conclusion

The proof of concept implementation integrates an EAP in the ArchiMEO enterprise ontology using rules and extending the ontology answering the main research question. The use of SHACL allowed the automated validation of the principle on a fictional ArchiMate model. Further, having the EAPs knowledge in a formalized way allows their re-usability and categorization.

Before the next design cycle, more principles (already available in the literature) are going to be formalized in a set of rules in the ArchiMEO ontology.

In the next iteration of the design cycle of the DSR, a real use case scenario will be introduced, evaluating the approach's applicability and relevance. A possible future extension of the approach is to validate EAP also among new modeling languages (e.g. BPMN [3]) together with ArchiMate.

## References

1. Borgers, M., Harmsen, F.: Strengthen the architecture principle definition and its characteristics—a survey encompassing 27 years of architecture principle literature. In: ICEIS (2). pp. 607–622 (2018)
2. Charuta Pande, Hans Friedrich Witschel, A.M., Montecchiari, D.: Hybrid conversational ai for intelligent tutoring systems. In: AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (2021)
3. Fischer, L.: BPMN 2.0 Handbook First Edition: Foreword by Bruce Silver, vol. 1. Future Strategies Inc. (2010)
4. Goodman, D.: JavaScript bible. John Wiley & Sons (2001)
5. Group, O.: The TOGAF Standard, Version 9.2. Van Haren Publishing (2018)
6. Group, T.O.: Archimate® model exchange file format for the archimate 3.1 modeling language (2019), <https://www.opengroup.org/xsd/archimate/>, accessed on 10.1.2021
7. Hevner, A., Chatterjee, S.: Design Research in Information Systems: Theory and Practice. Springer Science+Business Media, LLC (2010). <https://doi.org/10.1007/978-1-4419-5653-8>
8. Hinkelmann, K., Gerber, A., Karagiannis, D., Thoenssen, B., van der Merwe, A., Woitsch, R.: A new paradigm for the continuous alignment of business and it: Combining enterprise architecture modelling and enterprise ontology. *Computers in Industry* **79**, 77–86 (2016)
9. Hinkelmann, K., Laurenzi, E., Martin, A., Montecchiari, D., Spahic, M., Thönssen, B.: Archimeo: A standardized enterprise ontology based on the archimate conceptual model. In: Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELSWARD,. pp. 417–424. INSTICC, SciTePress (2020). <https://doi.org/10.5220/0009000204170424>
10. Hinkelmann, K., Laurenzi, E., Martin, A., Thönssen, B.: Ontology-based metamodelling. In: Business Information Systems and Technology 4.0, pp. 177–194. Springer (2018)
11. Josey, A.: A Pocket Guide to the ArchiMate® 3.1 Specification. Van Haren (2019)
12. Karagiannis, D., Woitsch, R.: Knowledge Engineering in Business Process Management. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 2, pp. 623–648. International Handbooks on Information Systems, Springer (October 2015). <https://doi.org/10.1007/978-3-642-45103-4>, [https://ideas.repec.org/h/spr/ihichp/978-3-642-45103-4\\_26.html](https://ideas.repec.org/h/spr/ihichp/978-3-642-45103-4_26.html)
13. Karpovic, J., Nemuraite, L.: Transforming sbvr business semantics into web ontology language owl2: main concepts. *Information Technologies* pp. 27–29 (2011)
14. Knublauch, H., Kontokostas, D.: Shapes constraint language (shacl). W3C Candidate Recommendation **11**(8) (2017)

15. Laurenzi, E., Hinkelmann, K., Van der Merwe, A.: An agile and ontology-aided modeling environment. In: IFIP Working Conference on The Practice of Enterprise Modeling. pp. 221–237. Springer (2018)
16. Laurenzi, E., Hinkelmann, K., Montecchiari, D., Goel, M.: Agile visualization in design thinking. In: New Trends in Business Information Systems and Technology, pp. 31–47. Springer (2020)
17. McGuinness, D.L., Van Harmelen, F., et al.: Owl web ontology language overview. W3C recommendation **10**(10), 2004 (2004)
18. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of sparql. ACM Transactions on Database Systems (TODS) **34**(3), 1–45 (2009)
19. Pham, T.A., Thanh, N.L.: An ontology-based approach for business process compliance checking. In: Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication. pp. 1–6 (2016)
20. Reynares, E., Caliusco, M., Galli, M., et al.: Sbrv to owl 2 mappings: An automatable and structural-rooted approach. CLEI Electronic Journal **17**(3), 3–3 (2014)
21. Ross, R.G.: Principles of the business rule approach. Addison-Wesley Professional (2003)
22. Savković, O., Kharlamov, E., Lamparter, S.: Validation of shacl constraints over kgs with owl 2 ql ontologies via rewriting. In: European Semantic Web Conference. pp. 314–329. Springer (2019)
23. Specifications, O.: Semantics of business vocabulary and business rules. Tech. rep., Tech. rep., Object Management Group (2015)
24. Vaishnavi, V., Kuechler, W.: Design Science Research methods and Patterns: Innovating Information and Communication Technology. Auerbach Publications, Taylor & Francis Group, Boca Raton, FL, New York (2007)