2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)

Ontology-Supported Enterprise Architecture Analysis

Murat Pasa Uysal
Department of Management Information Systems
Başkent University
Ankara, Turkey

Murat Karakaya
Department of Computer Engineering
Atılım University
Ankara, Turkey

***Abstract* — Today, processing integrated information within and between enterprises is increasingly becoming more and more critical, and so is the implementation and evaluation of an Enterprise Architecture (EA). The review of literature on EA evaluation shows several issues. However, the evaluation of EAs has not attracted sufficient attention, and thus, this research area has not been explored thoroughly yet. We believe that in order to ensure consistency, interoperability and computational inferences among EAs, a complete and holistic approach, rather than monolithic, should be developed. Therefore, in this study, we propose an ontology-supported process model for the evaluation of EAs, and present the implementation details. The main contributions of the present study are the improvements realized in the expressiveness, extensibility, and computable power of EAs, and their evaluation techniques. Although the proposed model requires gathering empirical evidences and investigating applications in concrete cases, the first implications of the proposed model indicates its validity and feasibility, and, hence, the initial results are promising for continuing future studies.**

Keywords — System Analysis, Enterprise Architecture; Ontology

I. INTRODUCTION

In an information technology (IT)-driven world, processing integrated information within and between enterprises based on standards is increasingly becoming more and more critical. Therefore, the Enterprise Architecture (EA) models and their implementation has been an important research area recently. Being a relatively new field, there are several definitions about EAs. The Open Group Architecture Framework (TOGAF), as an industry practice, defines EA as "*the crossing multiple systems and functional groups within an enterprise that encompasses all of its business services, processes, applications and infrastructure*" [1]. Another definition regards an EA 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 systems, and infrastructure*" [2].

There are different models to the EA implementations, which are reported by the review studies [3]. These approaches usually represent the frameworks or methods for modeling EAs while they describe guidelines and plans for enabling EA artefacts at different abstraction layers. As the research area on EA continues to mature, there have been also studies exploring the methods and techniques for evaluating EAs [4, 5]. Evaluation, of course, is not an easy task and it may require a range of different activities. This process becomes even more complex when regarding the dissimilarities between enterprises. For instance, one issue is the timing of an evaluation: should it be conducted on the current state or future, or both? As another example; what types of criteria should be considered for the evaluation of EAs?

The results of the limited studies on the evaluation of the EA approaches show that it can be qualitative or quantitative in nature. In a broader context, evaluation may have a technical, business, or financial focus. It can be done through the use of measurements, methods, models, or these may overlap. In most of the studies, the elements in an EA, such as IT projects, services, applications and business processes, are usually evaluated, rather than the architecture itself [2, 4-5]. As indicated by [5], most of the research on EA evaluation adopt a monolithic approach to the evaluation processes, rather than a holistic, and thus, they are usually concerned with the technical aspects of EAs. Moreover, the assurance of interoperability and communication between EAs is also another important issue. While an expected contribution of EAs is the support for IT related decisions, their conceptual and semantic foundations for these processes are still amongst the relatively less-explored research topics.

In terms of evaluating the EA itself, there may be some additional issues [5]. The management of dependencies and consistency between different information models and artifacts is an instance. Ensuring its traceability, measuring the potential impacts of changes to the EA is another challenge. The automatic validation of EA models and their conformance to reference models also need empirical evidences. Evaluating how an EA supports the governance [6] and decision making relevant to IT [7] is, therefore, in the research gap. However, EA evaluation is still attracting less attention within the academic literature.

In this paper, therefore, we propose an EA evaluation process model and present its potential impacts. The main contributions brought by this research are as follows:

- An ontology-supported process model and evaluation techniques for EA evaluation,
- Improvement of the expressiveness, extensibility and computable power of EAs through the use of ontological tools and techniques.

The next parts of this paper include the method, case study and conclusion sections.

II. METHOD

The study is conducted according to the guidelines and principles of Design Science Research (DSR) [8]. Thus, DSR provided a set of research activities for the development of proposed model. This research method mainly focuses on the

978-1-5090-3006-4/17/$31.00 ©2017 IEEE

creation of IT artifacts when solving real-world problems in Information Systems (IS) and Software Engineering (SE) domains [9]. Our research rigor relies on the foundations of IS and Ontology knowledge domains. The research output can be defined as the "application of a new solution (ontology-supported model) to a known problem (EA evaluation)" [10]. We followed the following steps during the development and evaluation of the proposed model (see Fig. 1).

A. Problem Identification and Solution Specification

To the best of our knowledge, any ontology-supported EA evaluation process model has not been proposed so far. Thus, this was identified as the first objectives of our solution to the research problem. Then, the main criteria that the solution model should meet were specified as follows:

- It should both regard and reflect the concerns and views of all stakeholders relevant to EAs.
- It should allow the integration of multiple domain information models, such as business, application and IT infrastructure,
- The model should be measurable, extensible, most importantly, it should support and allow computable architectural decisions.

B. Development of EA

TOGAF and Architecture Development Method (ADM) provided the required guidance for the EA development procedures [1]. ArchiMate was determined as a modeling language and IDE as well [2]. As can be seen from Fig. 1, architecture and ontology domains include overall activities conducted in a cross-cut and parallel fashion during the evaluation. First, the process starts with determining the stakeholders' needs as well as specifying the architectural and quality requirements. At this stage, the IDE's meta-model is determined as a domain independent ontology.

C. Analysis of EA

The second stage includes the current-state (as-is) of the EA, manual expert architectural analysis (for comparison purposes), and transformations of the EA to domain-specific (DSO) ontologies, which are also formed by extension of the domain-independent ontology (DIO) [11]. The transformations can be applied manual or using appropriate software. In this study, we use a plugin for converting the ArchiMate files into Web Ontology Language (OWL) formats [12]. Thus, the core architectural components, their behaviors and relationships can be represented in the form of ontological constructs.

D. EA Analysis and Evaluation

At the final stage, ontological analysis and evaluations are performed mainly for measuring the dependency and consistency between the components of the proposed EA. It is known that ontologies can be used in various application domains for different purposes. Such that, organizing the knowledge, making computational inferences, providing interoperability and communication between different knowledge-based systems are some of the examples. They are used for the semantic evaluations of knowledge representations as well. While regarding an EA as a conceptual and visual representation of an enterprise, in our study, we use the ontologies: (1) to form a consistent and theoretical foundation, and (2) to identify its goodness from the viewpoint of a specified ontology [13]. The ontology-based verification and validation procedures, and resolving architectural issues are performed in the Protégé IDE [14]. Exploring how well the proposed EA is able to represent the architecture, making computational inferences, and reflecting the required changes back to the EA are conducted respectively.

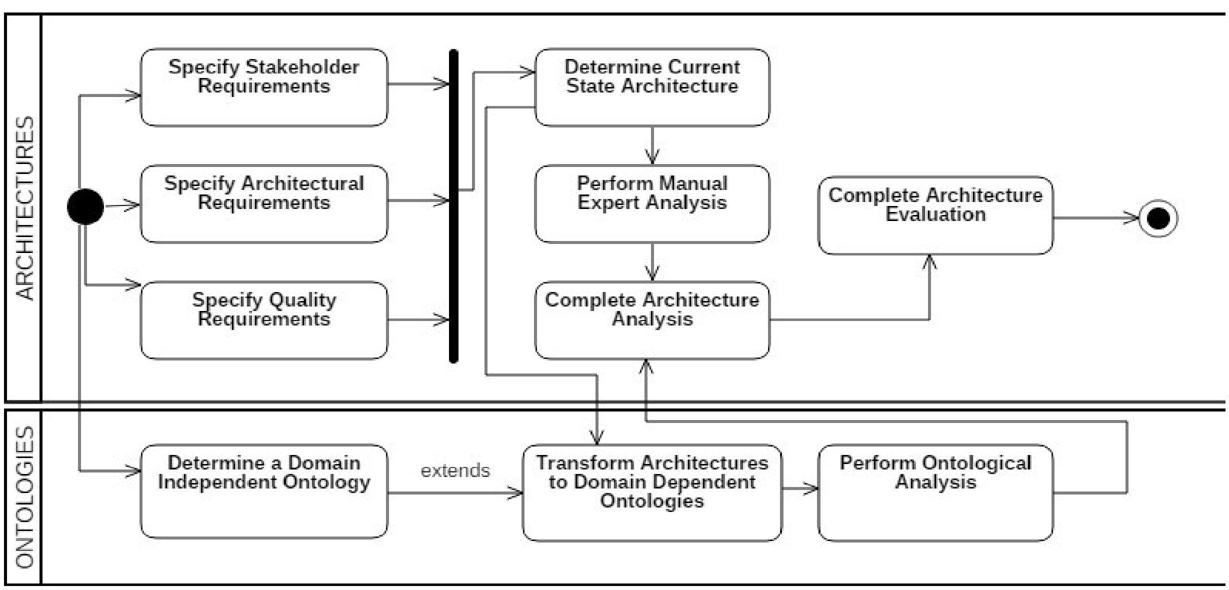


Fig. 1. Ontology-supported enterprise architecture evaluation model

III. A SAMPLE CASE

This sample case study mainly focuses on the analysis of current state (as-is) of a generic EA and leaves the manual expert analysis and complete evaluation as a future work. The details of the case study are provided in the following sections.

A. Determine the Current-State of EA

Supposing that Appendix 1 represents a small to medium size enterprise with a generic EA model, we include only the business, application and technical infrastructure layers, each with simple components, and then exclude the users and roles layer of the EA for simplicity purposes. Service orientation, which is also in line with the current trends in IT and software industry, provides a layered view of an EA. Thus, a service is defined as the unit of functionality, which is made available to its environment and it has some value for certain entities [2]. In this context, services are the main linking points between different layers. The implementation layers realize these services for higher layers. How the implementation layers make use of the services of other layers is shown by the “used by relations”. Moreover, the “realization relations” show how a service is realized in an implementation layer by using the component’s internal functionality.

In our example, three business processes (Business Process-1, 2, 3), one with 2 sub-processes, reside in the uppermost layer (see Appendix 1). The Application Component (Realization) Layer “realizes” the services by its internal application components (Application Component-1, 2, 3, 4). There are also some active (collaborating) and passive (data object) elements in this layer. These four application services are “used by” the corresponding business processes. Finally, the technical infrastructure layer “realizes” the infrastructure services (file access, message service, data service and web service) required by the application components located at its upper layer.

B. Transform the EA to an Ontology

We assume the Archimate’s meta-model as a DIO with a set of concepts required for addressing different scenarios [11]. Later, this meta-model is extended, and thus the generic EA model is transformed to a DSO by using the plug-in installed in Archimate IDE [11], which is represented by OWL descriptions (see Appendix 2). This software explores the EA diagram elements, defines them as classes of objects, and the relationships are transformed into object properties. One important issue is, here, assuring one-to-one correspondence and mapping the constructs between DIO and DSO. The issues pertaining to ontological completeness and clarity (deficiencies in construct overload, redundancy and excess) can be checked, such as, by using the Bunge-Wand-Weber ontological analysis models [13]. Although this is out of the scope of this paper, it is seen that the plugin is able to represent the architectural components and relationships with proper ontological elements.

C. EA Analysis and Evaluation

The ontological analysis and evaluation of the proposed EA comes after the transformation processes. It is suggested that the current structure of ontologies can be analyzed using SPARQL queries [14]. This would allow extracting different types of pieces of information that meet specified conditions. For example, we may be concerned with the types of elements exist at each layer of the EA. Another concern may be the types of relationships between these elements. These information are important to observe and analyze how the EA elements are interdependent, and also, how a change in one element at one layer would affect the corresponding elements both at the same and other layers. Thus, the queries and returning results shown by Fig. 2 give the required information as well as some idea about the structure of the EA. As can be seen, the business layer contains only one type of element while the application layer includes four different architectural elements. The second query, on the other hand, indicates that many of the structural relationships exist in the proposed EA. The elements connected by relations give us the basic structure; who and/or what acts on what.

Query-1:

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?element ?layer
WHERE
{
  ?element rdfs:subClassOf ?layer .
}
ORDER BY ?layer
```

Result-1:

element	layer
ApplicationComponent	ApplicationLayerElement
DataObject	ApplicationLayerElement
ApplicationCollaboration	ApplicationLayerElement
ApplicationService	ApplicationLayerElement
BusinessProcess	BusinessLayerElement
SystemSoftware	TechnologyLayerElement
InfrastructureInterface	TechnologyLayerElement
Network	TechnologyLayerElement

Execute

Query-2:

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?RelationshipsInEA ?LanguageForEA
WHERE
{
 ?RelationshipsInEA rdfs:subPropertyOf ?LanguageForEA .
}
```

Result-2:

RelationshipsInEA	LanguageForEA
realisation	archimate
access	archimate
association	archimate
aggregation	archimate
triggering	archimate
composition	archimate
usedBy	archimate

Execute

Fig. 2. SPARQL queries exploring the structure of the EA in Protégé IDE

The further and comprehensive evaluation procedures can be conducted by using one or more of OWL language descriptions depending on the requirements of implementers or users. If our ontology is a classification hierarchy with simple constraints and it has a lower complexity, then OWL Lite can be used [15]. If the evaluation process is to be more conclusive, expressive and computationally complete, the use of OWL DL may be suggested. We may use the OWL Full language if we need a syntactic freedom of RDF, however, it cannot offer computational guarantees.

IV. Conclusion

Aforementioned, there have been various issues related to the evaluation of EAs. In this study, therefore, we propose an ontology-supported process model for the evaluation EAs, and present its implementation steps. The main argument is that a complete and holistic approach should be adopted rather than monolithic to ensure the consistency, interoperability and computational inferences during EA evaluation procedures. Furthermore, the rationale behind the use of an ontology-supported process model can be summarized as follows:

- To adopt a holistic approach rather than monolithic that usually concerns technical aspects of EAs,
- To have an automatic validation of EA models and their conformance to other reference EA models,
- To manage dependency and consistency between different information models (business, data, application, technology and conceptual) existed in EAs,
- To establish interoperability and communication between EAs and their components, and thus, allow computational inferences about EAs,
- To organize knowledge about EAs while enabling the reuse of this knowledge.

Although the time limitations in the research led to the use of a generic case, the first implications provided by the analysis tools and inferences indicate the feasibility of the proposed model. However, it still needs empirical evidences and also requires the application of this model in real case studies. Therefore, we direct our attention to the future work that would address these issues.

References

[1] Open-Group, TOGAF Version 9.1 Evaluation Form, 2011, published in the U.S. by The Open Group.

[2] M. Lankhorst, Enterprise Architecture at work: Modelling, communication, and analysis, 2009, Springer-Verlag Berlin Heidelberg.

[3] B.D. Rouhani, M.N. Mahrin, F. Nikpay, R.B. A.P. Nikfard. A systematic literature review on enterprise architecture implementation methodologies, Information and Software Technology, 2015, vol.62, pp. 1-20.

[4] M. Hoffmann, Analysis of the current state of enterprise architecture evaluation methods and practices, European Conference on Information Management and Evaluation, 2007, September 20-21, 2007, Montpellier, France.

[5] P. Andersen, A. Carugati, "Enterprise architecture evaluation: A systematic literature review" in Mola, L., Carugati, A,. Kokkinaki, A., Pouloudi, N., (eds), Proceedings of the 8th Mediterranean Conference on Information Systems, Verona, Italy, 2014, September 03-05. ISBN: 978-88-6787-273-2.

[6] K. D. Niemann, "From Enterprise Architecture to IT Governance, Elements of Effective IT Management", 1st Ed., 2006, Friedr. Vieweg & Sohn Verlag.

[7] J. Klein, M. Gagliardi, "A workshop on analysis and evaluation of enterprise architectures," 2010, CMU Software Engineering Institute Technical Report.

[8] A Hevner, S. March, J. Park, S. Ram, "Design science in information systems research", MIS Quarterly, 2004, 28(1), pp. 75-105.

[9] V.K Vaishnavi,. W.J. Kuechler, "Design science research methods and patterns: Innovating information and communication technology", 2008, Auerbach Publications, Taylor & Francis Group, NW, USA.

[10] S. Gregor, A.R. Hevner, "Positioning and presenting design science research for maximum impact", 2013, MIS Quarterly, 37(2), pp.337-355.

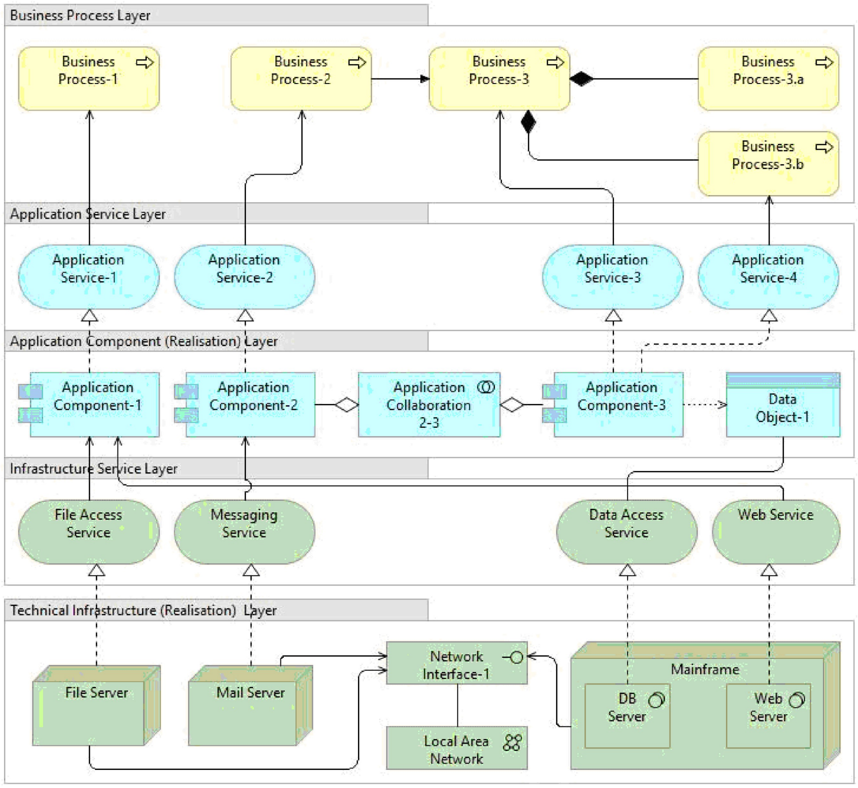
[11] G. Antunes, M. Bakhshandeh, R. Mayer, J. Borbinha, A. Caetano, "Using ontologies for enterprise architecture analysis", 17th IEEE International Enterprise Distributed Object Computing Conference Workshop, 2013, DOI: 10.1109/EDOCW.2013.47.

[12] P. Szwed, "Plugin for transforming Archimate files into OWL", Retrieved from http://home.agh.edu.pl/ ~pszwed/en/ doku.php?id=archi_to_owl, on December 12, 2016.

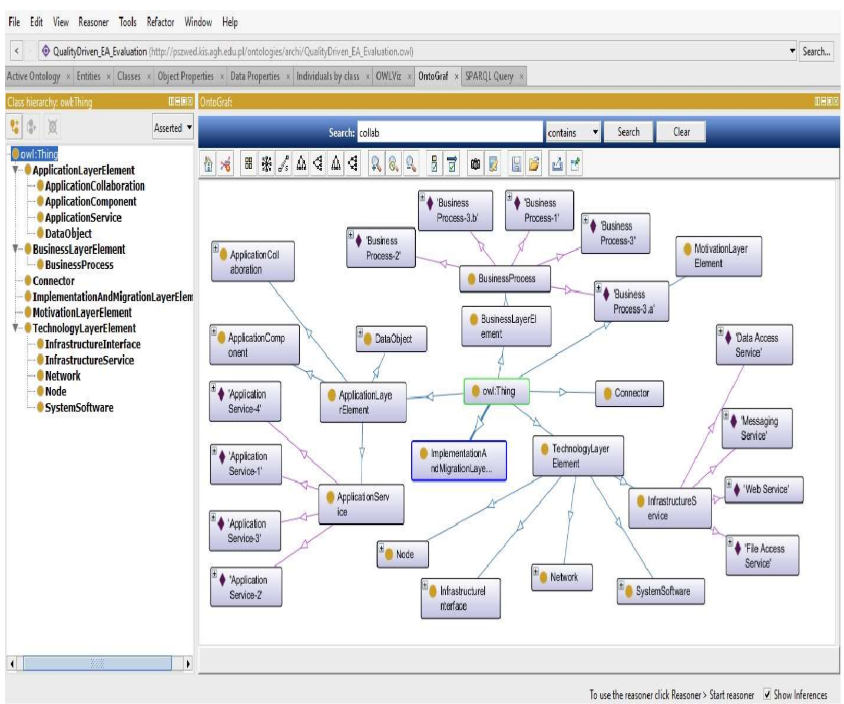
[13] P. Green, M. Rosemann, "Business systems analysis with ontologies", 2005, Idea Group Publishing, PA, USA.

[14] M.A. Musen, "The Protégé project: A look back and a look forward. AI Matters", Association of Computing Machinery Specific Interest Group in Artificial Intelligence, 1(4), 2015. DOI: 10.1145/2557001.25757003.

[15] W3C OWL Web Ontology Language Guide, Retrieved from "https://www.w3.org/" on December 10, 2016.



Appendix 1. A Generic Enterprise Architecture Model



Appendix 2. A generic enterprise architecture transformed to an ontology in Protégé IDE