Coping with Change – Specialization and Interoperability of ArchiMate Language

Martin Rod[1]

[1]Department of Information Engineering, Czech University of Life Sciences Prague, Czechia; e-mail: rodm@pef.czu.cz; https://orcid.org/0000-0003-3336-265X

Abstract. This paper discusses the usage of the ArchiMate language in the viewpoint of achieving consistency in the enterprise architecture, which in turn enables interoperability across the whole organization. ArchiMate language in continuous development. However, its popularity and adoption rate is miles ahead of the mentioned development in form of official releases. Thus, The ArchiMate language is also evolving in multiple shadowy ways parallel to the official one. The described situation presents threats for the effective maintenance of enterprise architecture and in consequence, hinders the decision-making process itself. This short paper presents a case study of Czech eGovernment and proposes a way how to achieve viable long-term conformity with the official release and its compliance.

Keywords: enterprise architecture; model consistency; model continuity; ArchiMate.

1 Introduction

With the increased scope of any of the information system, the complexity of this system also increases. The same applies to any organization. As the organization grows the complexity of it also and probably not linearly increases. As a parallel to managing the complexity of information systems a solution for managing the complexity of organization has emerged – the discipline of enterprise architecture.

This discipline contains of multiple rival frameworks and other tools which tackle this complexity from different sides (Lankhorst, 2017). One of approaches to cope with organization complexity is usage of diagrams, models which can be graphically created, maintained and interpreted. This paper is mainly focused on one of this type of tool, namely it is ArchiMate language which is one of a major modelling language concerning enterprise architecture. In this work the aspect to ArchiMate recent changes and further development and potential weaknesses which could lower value coming from enterprise architecture usage.

This paper is organized into the following parts:

Proceedings of the 9th International Conference on Information and Communication Technologies in Agriculture, Food & Environment (HAICTA 2020), Thessaloniki, Greece, September 24-27, 2020.

- This introduction is followed by chapter 2 with more detailed information about ArchiMate.
- Chapter 3 reflects on the recent changes to ArchiMate in correspondence to its implication for the modelling process. This chapter also points out the main relevant issues that are still unresolved and hinders ArchiMate usage due to the tailoring.
- Chapter 4 is a case study of Czech eGovernment accompanied by discussion and suggested course of action concerning ArchiMate users and ArchiMate development.
- The conclusion of this article.

2 ArchiMate Usage and Specification

The ArchiMate language is meant to ease the process of managing enterprise assets in order to increase effectiveness of an organization (Lankhorst, 2017). As a language ArchiMate could be viewed as a tool enabling model creation. This creation, due the ArchiMate specification, could be realized via modelling software – Computer Aided Systems Engineering (CASE). ArchiMate retains its higher-level abstraction and so UML, BPMN and other means are used to represent the realization of those concepts. From this point of view ArchiMate is modelling architecture, these other mentions concepts are used for designing the actual solution.

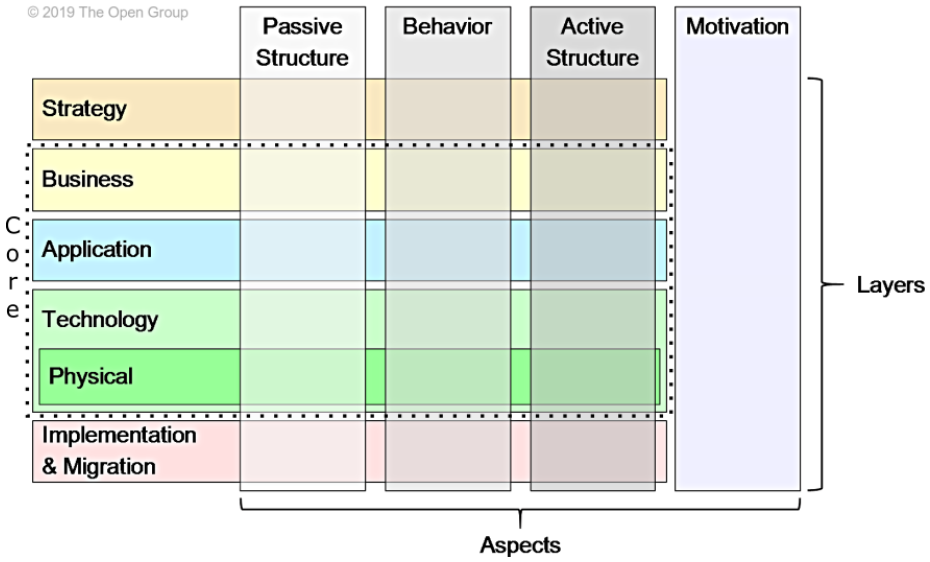

Fig. 1. ArchiMate full framework, core framework denoted by the dotted line. Source: https://pubs.opengroup.org/architecture/archimate3-doc/chap03.html, edited.

The ArchiMate language is quite a new language in comparison to the Unified Modelling language (UML) or the Entity-Relationship Diagram (ERD) which was in the modern form already published in Chen's work in 1976 (Chen, 1976). Due to this fact, ArchiMate incorporates the best practice form the languages that precede him. ArchiMate is influenced by UML from which incorporates and somewhat transforms structural relations (such as aggregation, composition, inheritance). On the other hand, it does not incorporate multiplicity or class instance notation due to its different abstraction levels (Object Management Group, 2017; The Open Group, 2019).

The full ArchiMate language currently breaks down into multiple layers and aspects (core version has only 3 layers and 3 aspects) fig. 1.

3 On Recent ArchiMate Standard Changes and Tailoring

The last addition for the ArchiMate language up to this date is release numbered as 3.1 in November 2019. This release replaces version denoted as 3.0.1 from August 2017.

The last release brought a new element, value stream, which brings Porter's value chain into ArchiMate. However, the concept of adding exactly this element, respectively coping with its absence was already proposed and discussed by Caetano et al. (Caetano et al., 2017). Other main changes were the addition of orientation of association relationship (the weakest relation in ArchiMate) and updated relationship derivation rules. These changes could be referred to as minor ones. The changes and elements introduced in 3.0.1 were bigger. However, the change to association relationship could bring issues with updating the old models and using the old not oriented association with the current one.

The ArchiMate language like TOGAF is built to be changed, further developed. This process is usually called tailoring (The Open Group, 2019, 2018). This enables to add new features to the language. On the other hand, these changes could be problematic when a new version of ArchiMate is released and is not complementary to the tailored one, then indeed the problem of consistency arises.

The expected addition of security aspect was not added and so the security is mainly tailored and is still causing active discussion (Hacks et al., 2019; The SABSA Institute, 2018).

4 Tailoring your ArchiMate - Case Study of Czech eGovernment

Although Czech eGovernment did not adopt TOGAF and ArchiMate fully, the framework that it uses is mainly based on those two. Czechia made its own framework and changed up the content of ArchiMate language, some features of ArchiMate has been restricted, e.g. elements but new features have been added for example stereotypes. The Czech architecture framework (National Architectural Framework, abbreviated NAR) is maintained by the Office of the Chief eGovernment Architect of the Ministry of the Interior of the Czech Republic (Czech Republic, 2020).

Overall those changes could be justified as a proposition to bring all those organizations participating in Czech eGovernment up to speed and as discussed by Al-Kharusi et al. (Al-Kharusi et al., 2018).

In order to cope with change a conceptual model of the proposed application for Czech eGovernment is proposed fig. 2. The proposal and solution are using the Model repository application which is currently being tested. The analytics application would be then built to use a graph database to perform analytics on tailored and base ArchiMate models. The mechanism of how to cope with possible changes and inconsistencies as the ArchiMate and tailored models mature is as follows:

- By access to model source code (which represents the model and is machine-readable), a stripped standardized model will be made.
- The previous step achieves consistency with future ArchiMate releases.
- The stripped model could be saved as a differential file, with indexed changes that mean the whole tailored model could be recreated.

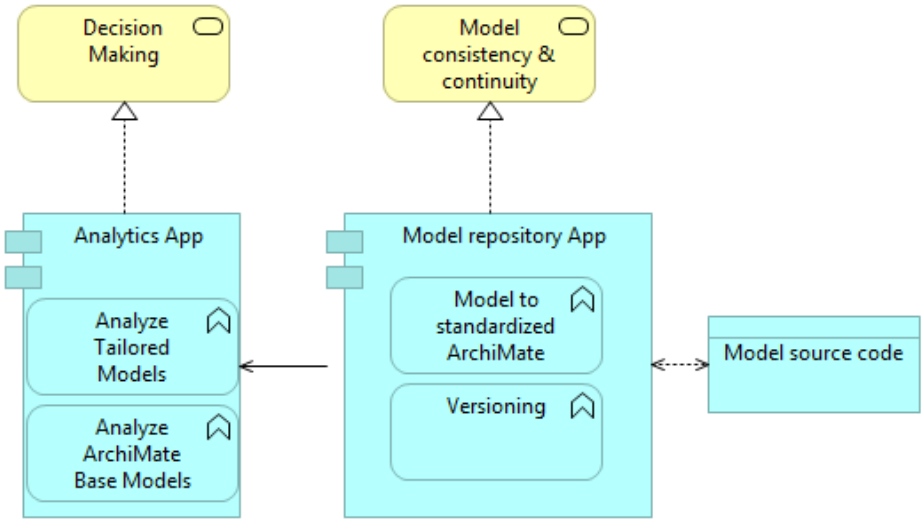

Fig. 2. Conceptual architecture model for continuality and consistency. Source: author.

5 Conclusion

This paper presented the ArchiMate language in view of recent changes and issues which are still present in utilizing the ArchiMate language. The principle of tailoring for the perspective of ArchiMate was discussed. Then a case study of Czech eGovernment is presented. Due to the tailoring process, the approach of versioning and architecture model source code manipulation is presented via the conceptual model. Given the fact that ArchiMate models could be exported and used in the form of a machine-readable file, this function could be fully automated. However, this

approach has drawbacks as it is generally "doubling" the number of models – one tailored and one ArchiMate base specification ready (the data usage is not doubled as only the difference between the two models must be kept). On the other hand, adding new information to the ArchiMate base model could be then cross-referenced to the original one and so it accelerates the modeling process.

References

1. Al-Kharusi, H., Miskon, S., Bahari, M., 2018. Enterprise Architecture Development Approach in the Public Sector: International Journal of Enterprise Information Systems 14, 124–141. https://doi.org/10.4018/IJEIS.2018100109
2. Caetano, A., Antunes, G., Pombinho, J., Bakhshandeh, M., Granjo, J., Borbinha, J., da Silva, M.M., 2017. Representation and analysis of enterprise models with semantic techniques: an application to ArchiMate, e3value and business model canvas. Knowl Inf Syst 50, 315–346. https://doi.org/10.1007/s10115-016-0933-0
3. Chen, P.P.-S., 1976. The entity-relationship model---toward a unified view of data. ACM Trans. Database Syst. 1, 9–36. https://doi.org/10.1145/320434.320440
4. Czech Republic, 2020. National eGovernment Architecture of Public Administration [Architektura eGovernmentu ČR] [WWW Document]. URL https://archi.gov.cz/en:start (accessed 3.31.20).
5. Hacks, S., Hacks, A., Katsikeas, S., Klaer, B., Lagerstrom, R., 2019. Creating Meta Attack Language Instances using ArchiMate: Applied to Electric Power and Energy System Cases, in: 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC). Presented at the 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC), IEEE, Paris, France, pp. 88–97. https://doi.org/10.1109/EDOC.2019.00020
6. Lankhorst, M., 2017. Enterprise Architecture at Work: Modelling, Communication and Analysis.
7. Object Management Group, 2017. About the Unified Modeling Language Specification Version 2.5.1 [WWW Document]. URL https://www.omg.org/spec/UML/About-UML/ (accessed 3.31.20).
8. The Open Group, 2019. ArchiMate® 3.1 Specification [WWW Document]. URL https://pubs.opengroup.org/architecture/archimate3-doc/ (accessed 3.31.20).
9. The Open Group, 2018. The TOGAF® Standard, Version 9.2 [WWW Document]. URL https://pubs.opengroup.org/architecture/togaf9-doc/arch/ (accessed 3.31.20).
10. The SABSA Institute, 2018. SABSA Resources [WWW Document]. The SABSA Institute. URL https://sabsa.org/white-paper-requests/ (accessed 3.31.20).