Rational Enterprise Architecture

Leendert van der Torre[1(✉)] and Marc van Zee[2]

[1] University of Luxembourg, Luxembourg City, Luxembourg
leon.vandertorre@uni.lu
[2] Google Research, Zurich, Switzerland
marcvanzee@google.com

Abstract. We are interested in formal foundations for enterprise decision support. In this perspective, enterprise architecture is characterised by highly uncertain plans in a changing environment, and translates strategic goals into an IT strategy. Typically there are a large number of stakeholders with conflicting views, communicating plans of action, and explaining decisions instead of making them. An enterprise architecture considers qualitative before quantitative data, has stronger business focus than other disciplines, and politics, emotions, and soft skills play a bigger role than in other areas. We view a plan abstractly as a sequence of commitments in time, and each commitment in the plan may come with a number of underlying assumptions. If these underlying assumptions change, then parts of the plan may require revision, which in turn may invalidate other parts of the plan, and so on. Therefore, assumptions have an inherently *non-monotonic* character: they are assumed to be true, unless it becomes clear they are false. This is related to the resource-boundedness of enterprise architecture: an enterprise architect cannot always know all of the assumptions, especially for long term plans.

1 Enterprise Architecture

We use the following definition of an enterprise architecture:

> "Those properties of an enterprise that are necessary and sufficient to meet its essential requirements" [6].

A commonly used metaphor for an enterprise architect is a city planner. City planners work on long-term visions, providing the roadmaps and regulations that a city uses to manage its growth and provide services to citizens. Using this analogy, we can differentiate the role of the system architect, who plans one or more buildings; software architects, who are responsible for the HVAC (Heating, Ventilation and Air Conditioning) within the building; network architects, who are responsible for the plumbing within the building, and the water and sewer infrastructure between buildings or parts of a city. The enterprise architect however, like a city planner, both frames the city-wide design and other activities into the larger plan.

There are a large number of responsibilities and skills that can potentially be associated with an enterprise architect. One way to frame these responsibilities and skills is to distinguish two main roles of an enterprise architect (Fig. 1).

S. Benferhat et al. (Eds.): IEA/AIE 2017, Part I, LNAI 10350, pp. 9–18, 2017.
DOI: 10.1007/978-3-319-60042-0_2

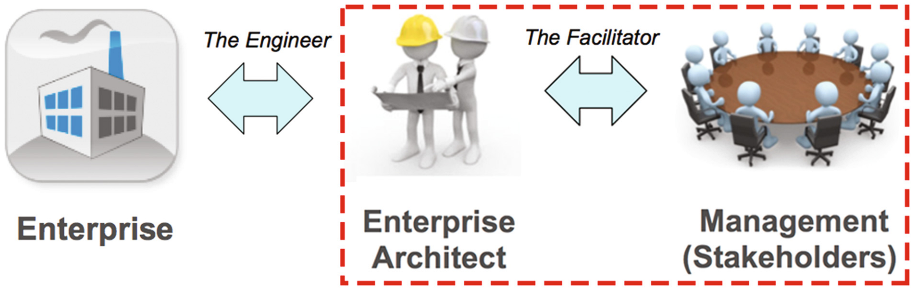

Fig. 1. Two roles of an enterprise architect, and our scope (red dashed line)

1. *The engineer.* The architect (usually a group of architects) develops models of business and IT. These models can be UML-like diagrams, specialised enterprise architecture diagrams, risk analysis tools, textual descriptions, or any other representation that the architect feels comfortable with.
2. *The facilitator.* The architect (usually the lead architect) is intermediator between IT and business. Often, the architect attends business meetings and serves as an IT expert, consulted by managers on what specific IT solutions to use.

Enterprise architects often work on large projects with a long duration, and try to steer the enterprise such that the long-term goals and visions of the enterprise are reached, with an emphasise on the IT part of the enterprise. Let us illustrate this through a simple example.

Example 1 (University of Luxembourg). The University of Luxembourg is in the process of merging three separate campuses into a single campus on a new location. This process takes several decades, and requires a complete re-design of the IT landscape. The university would like to ensure that their long-term strategy and vision is aligned with their overall IT strategy, so they hire a team of enterprise architects. The lead architect discusses and refines the strategic goals and vision of the university with the executive board. She then works together with the other architects to develop a long-term IT strategy. This plan involves modelling the current business-IT landscape in ArchiMate, a specialised enterprise architecture modelling language, performing risk analysis on various alternatives, etc. The board, not having expertise nor time to understand the technical details of this problem, are then presented a simplified version of these plans. The members of the board have different – and changing – concerns, discuss these plans with each other and with the architect, who then brings this input back to his team of architects. This process may be repeated any number of times.

Although the example above is (purposely) simplified, it does give an idea of the large number of varying tasks an enterprise architect should be able to carry out. The enterprise architect is responsible for ensuring the IT strategy and planning are aligned with the company's business goals, and must optimise information management through understanding evolving business needs (*facilitator*), but it also must ensure projects do not duplicate functionality or diverge from each other, and work with solution architects to provide a consensus based enterprise solution that is scalable and adaptable (*engineer*). As a result, an enterprise architect should have a large number of skills, from technical skills such as comprehensive knowledge of hardware, software, application, and systems engineering, to soft skills such as communication skills and the ability to explain complex technical issues in a way that non-technical people understand it, to managerial skills such as project and program management planning, time management and prioritisation.

2 ArchiMate

There are a large number of definitions of enterprise architecture in existence. The many existing definitions of enterprise architecture each have their own focus. For example, the definitions of enterprise architecture provided by IEEE [20], TOGAF [21], ArchiMate [8] and Giachetti [5], tend to focus on the elements of an enterprise architecture. The Zachman framework [31] and the GERAM framework [2] also have their focus more on the possible elements (in particular the relevant viewpoints) of an enterprise architecture. Dietz and Hoogervorst [4,7] define enterprise architecture primarily in terms of its meaning. The definition provided by Ross et al. [16] touches both on the purpose and elements aspects, while Op 't Land, et al. [11] put the focus more on its purpose as a means for informed governance for enterprise transformations.

In line with the different definitions, various languages and techniques for enterprise architecture have been developed in the last decades. In the ArchiMate project [10], a language for the representation of enterprise architectures was developed, together with visualisation and analysis techniques. The resulting ArchiMate language is an Open Group standard [8,21], and the TOGAF/ArchiMate combination of standards is playing an increasing role in the marketplace [30]. ArchiMate distinguishes itself from other languages such as Unified Modelling Language and Business Process Modelling and Notation (BPMN) by its scope on enterprise modelling as a whole. It is a language for describing the construction and operation of business processes, organisational structures, information flows, IT systems, and technical infrastructure. This is comparable to an architectural drawing in classical building where the architecture describes the various aspects of the construction and use of a building. This insight helps the different stakeholders to design, assess, and communicate the consequences of decisions and changes within and between these business domains.

The Archimate framework divides the enterprise architecture into a business, application and technology layer. In each layer, three aspects are considered: active elements that exhibit behaviour (e.g. Process and Function), an internal structure and elements that define use or communicate information. One of the main objectives of the ArchiMate language is to define the relationships between concepts in different architecture domains.

Initial research for the capturing of architectural design decisions was done during the ArchiMate project as well [10,29]. Following this preliminary work, Plataniotis et al. [12–14] develop a framework for the capturing and rationalisation of enterprise architecture decisions called EA Anamnesis. This framework formalises enterprise architecture decisions through meta models.

3 Rational Enterprise Architecture

Besides the preliminary attempts we described about, there is currently little research on enterprise architecture decision support. We provide an abstract overview of the evolution of decision support for enterprise architecture in Fig. 2. In the early days, that is, in the time before desktop computers were available to the common man, the tasks of enterprise architects were done on paper (left image, coaster). At the next stage of the evolution, the introduction of the desktop computer made available general tools such as Microsoft Powerpoint (second image from the left, Powerpoint). As the field of enterprise architecture matured, specialised languages and tool support were developed for enterprise architects, such as the ArchiMate language [9] (third image from the left, ArchiMate). In this thesis, we aim to lie the foundations for the next step (right image, question mark).

Fig. 2. Evolution of enterprise architecture decision support. From coasters (left), to Powerpoint (second left), to ArchiMate (second right), to the future (right)

We contribute to the development of enterprise architecture decision support by focusing on the *facilitator* role (Fig. 1, red dashed line) of the enterprise architect. This means we focus on the interactions and dynamics between the enterprise architect (the facilitator) and management. Our aim is to formalise this using logical frameworks that store important commitments, or high-level decisions, made during discussions or meetings. Such decisions are based on underlying assumptions. Assumptions may pertain to the goals of stakeholders, strategic directions of the enterprise, architecture principles, requirements, arguments put forward in discussions, etc. In practice, enterprises are confronted with frequent changes and challenges to these assumptions. Even more, the assumptions, and their relative priority, also depend on the specific stakeholders that are involved in creating the architecture of the future enterprise, as well as the actual transformation.

4 Characterizing Decision Making in Enterprise Architecture

The state of the art research in enterprise architecture is diverse with many different definitions emphasising different parts of the field. Therefore, one of our first activities has been to understanding the field of enterprise architecture better by identifying important characteristics of enterprise architecture. We did this in various ways, and we briefly discuss two main approaches.

Empirical study. We performed an empirical study on how the practice of high-level decision making (i.e., decisions in the role of the *facilitator* of Fig. 1) in enterprise architecture is perceived by professional enterprise architects. We did so through a questionnaire incorporating qualitative and quantitative questions, targeting enterprise architects around the world, in order to determine what they consider to be the important characteristics of enterprise architecture decision making, and whether these characteristics differ considerably from those in closely related fields such as software architecture [22].

The most important characteristics of enterprise architecture we found are:

1. Translating strategic goals into an IT strategy
2. Communicating plans of action
3. Explaining decisions instead of making them
4. Qualitative before quantitative data
5. Stronger business focus than other disciplines
6. Politics, emotions, and soft skills play a bigger role than in other disciplines
7. Large number of stakeholders with conflicting views
8. Highly uncertain plans in a changing environment

We use these eight characteristics as yardsticks for a formal theory to support enterprise architects. We observe that approaches based on the idea of *classical rationality* may be less appropriate that those based on *bounded rationality*, which is motivated by the observation that our study shows architects often work with incomplete data and face many types of uncertainty. We propose to use logical theories based on practical reasoning, since such theories have rich concepts for motivational attitudes such as goals and intentions, which appear to be playing an important role.

Determining an ontology. We analyzed an existing framework for capturing enterprise architecture design decisions called *EA Anamnesis* and recognize various ambiguities and flaws in the specification. We proposed a more precise formalisation of EA Anamnesis using first-order logic, and used this first-order logic to develop an enterprise architecture ontology. Our main conclusion is that our formalism does not offer much support for the type of reasoning processes specific to enterprise architecture we found in the previous chapter. More notably, it is not directly possible to reason about the *dynamics* of decisions in a principled way.

5 Reasoning About Enterprise Dynamics

One of the earlier practitioners in system architecture Steven H. Spewak defined *enterprise architecture planning* as "the process of defining architectures for the use of information in support of the business and the plan for implementing those architectures" [19]. An important lesson from the ArchiMate project [10] was that it is inherently difficult to plan architectural design. TAFIM, an enterprise architecture model by and for the United States Department of Defence recommends that in a typical five-year plan, only the first year is detailed, and the other steps are described only in a very abstract way. At each step in the plan, not only must the future abstract plans be further detailed, but architectural designs also have to be reconsidered and possibly revised.

This is a complicated picture, and developing a logical framework for this type of reasoning may seem daunting. We view a plan abstractly as a sequence of commitments in time, and each commitment in the plan may come with a number of underlying assumptions. If these underlying assumptions change, then parts of the plan may require revision, which in turn may invalidate other parts of the plan, and so on. Therefore, assumptions have an inherently *non-monotonic* character: they are assumed to be true, unless it becomes clear they are false. This is related to the resource-boundedness of our problem domain: an enterprise architect cannot always know all of the assumptions, especially for long term plans.

6 Formal Methods for Rational Enterprise Architecture

Motivated by our empirical findings which indicated that the enterprise architecture domain is very complex, with many types of uncertainty, we choose to apply a *separation of concerns* approach when developing our formal theory. We make a distinction between the *enterprise architect* and an *intelligent database*.

- The *enterprise architect* and the *stakeholders* form plans, have discussions, change preferences, pursue goals, etc.
- The *intelligent database* stores temporal commitments made by the architect and the stakeholders, and reasons about the consistency of these commitments with underlying assumptions.

As such, the intelligent database is a tool to assist the enterprise architect in dealing with high cognitive load of the enterprise architecture domain. Indeed, our intelligent database is very similar to an *intelligent calendar* (See [17] for conceptual underpinnings). Such a *database perspective* has proven itself useful in the consumer domain already: Yoav Shoham developed these ideas with Jacob Banks, one of his PhD students, and behavioral economist Dan Ariely into the intelligent calendar application Timeful, which attracted over $6.8 million in funding and was acquired by Google in 2015[1], who aim to integrate it into their

[1] http://venturebeat.com/2015/05/04/google-acquires-scheduling-app-timeful-and-plans-to-integrate-it-into-google-apps/.

Fig. 3. We view consistency of commitments and beliefs as a database management problem.

Calendar applications. As Shoham [18] says himself: "The point of the story is there is a direct link between the original journal paper and the ultimate success of the company (Fig. 3)." (p. 47)

In our system, an enterprise architect is in the process of making plans, possibly with a group of stakeholders, and stores commitments and beliefs in two database. We focus on two main sources for the databases to change:

1. The enterprise architect forms a new beliefs, e.g. from discussions with stakeholders, or from a piece of data. If the new belief is inconsistent with the existing beliefs, these beliefs will have to be revised to accommodate it. We give general conditions on a single revision with new information that the database has already committed to incorporating using ideas from the classical AGM postulates [1] approach.
2. The enterprise adds a commitment. We formalise these tasks as *future directed atomic intentions*, understood as time-labeled actions pairs (a, t) that might make up a plan. It is assumed the enterprise architect has already committed to the intention, so it must be accommodated by any means short of revising beliefs. The force of the theory is in restricting how this can be accomplished. The job of the database is to maintain consistency and coherence between intentions and beliefs.

In order to formalise this, we develop a logic for beliefs about actions in time. We associate pre-and postconditions with actions. A key element in our approach is the asymmetry we put on assumptions about preconditions and postconditions of actions. First of all, we assume that

If an enterprise architect intends to do an action, she assumes the consequences of this action hold.

However, for preconditions we add a weaker requirement:

If an enterprise architect intends to do an action she cannot believe that its preconditions do not hold.

The result of this weakened requirement is that preconditions of actions are treated as *assumptions*: An enterprise architect makes plans under the assumption that these preconditions will be made true somewhere in the future.

This computationally motivated view on decision support leads to a very interesting take on intention revision, which is in stark contrast with existing approaches which are based on philosophical logic (e.g., [3,15]). In our work, we develop a temporal logic comparable to CTL*, but slightly less complex, in order to describe our belief database. We axiomatise our logic and prove it is sound and strongly complete with respect to our semantics [25,28]. In order to specify the dynamics of our databases, we develop a set of revision postulates comparable to the well-known AGM postulates for belief revision [1]. We prove that our revision postulates correspond to a preorder over semantical models, such that revising beliefs and intentions corresponds to selecting minimal models in some preorder [23,26,27].

7 Conclusion

While decision support systems have found their way in many domains such as software architecture and information architecture, they have not been adopted in enterprise architecture. We report on the past four years of our effort in analysing the domain of enterprise architecture, and developing logical foundations that can be used as a starting point for such enterprise decision support systems. The results and insights in this paper are further elaborated in the PhD thesis of Marc van Zee [24].

References

1. Alchourrön, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: partial meet contractions and revision functions. J. Symbolic Logic **50**, 510–530 (1985)
2. Bernus, P., Nemes, L., Schmidt, G. (eds.): Handbook on Enterprise Architecture. International Handbooks on Information Systems. Springer, Heidelberg (2003)
3. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artif. Intell. **42**(2–3), 213–261 (1990)
4. Dietz, J.: Architecture - Building Strategy into Design. Netherlands Architecture Forum. Academic Service - SDU, The Hague (2008)
5. Giachetti, R.: Design of Enterprise Systems: Theory, Architecture, and Methods. CRC Press, Boca Raton (2010)
6. Greefhorst, D., Proper, E.: Architecture Principles: The Cornerstones of Enterprise Architecture, 1st edn. Springer Publishing Company Incorporated, Heidelberg (2011)
7. Hoogervorst, J.: Enterprise Governance and Enterprise Engineering. Springer, Berlin (2009)

8. Iacob, M.-E., Jonkers, H., Lankhorst, M., Proper, H.: ArchiMate 1.0 Specification. The Open Group (2009)
9. Iacob, M.-E., Jonkers, H., Lankhorst, M., Proper, H.: ArchiMate 2.0 Specification. The Open Group (2012)
10. Lankhorst, M. (ed.): Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin (2005)
11. Op 't Land, M., Proper, H., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture - Creating Value by Informed Governance. Enterprise Engineering Series, Germany (2008)
12. Plataniotis, G., Kinderen, S., Proper, H.A.: Capturing decision making strategies in enterprise architecture – a viewpoint. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) BPMDS/EMMSAD -2013. LNBIP, vol. 147, pp. 339–353. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38484-4_24
13. Plataniotis, G., de Kinderen, S., Proper, H.A.: Relating decisions in enterprise architecture using decision design graphs. In: 2013 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC), pp. 139–146. IEEE (2013)
14. Plataniotis, G., De Kinderen, S., Proper, H.A.: Ea anamnesis: an approach for decision making analysis in enterprise architecture. Int. J. Inf. Syst. Model. Des. (IJISMD) **5**(3), 75–95 (2014)
15. Rao, A., Georgeff, M.: Modeling rational agents within a BDI-architecture. In: KR (1991)
16. Ross, J., Weill, P., Robertson, D.: Enterprise Architecture as Strategy: Creating a Foundation for Business Execution. Harvard Business School Press, Boston (2006)
17. Shoham, Y.: Logical theories of intention and the database perspective. J. Philos. Logic **38**, 633–647 (2009)
18. Shoham, Y.: Why knowledge representation matters. Commun. ACM **59**(1), 47–49 (2016)
19. Spewak, S.H., Hill, S.C.: Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology. QED Information Sciences Inc. (1993)
20. The Architecture Working Group of the Software Engineering Committee. Recommended Practice for Architectural Description of Software Intensive Systems. Technical report IEEE P1471: 2000, ISO/IEC 42010: 2007, Standards Department, IEEE, Piscataway, New Jersey, September 2000
21. The Open Group. TOGAF Version 9. Van Haren Publishing, Zaltbommel (2009)
22. van der Linden, D., van Zee, M.: Insights from a study on decision making in enterprise architecture. In: PoEM (Short Papers), CEUR Workshop Proceedings, vol. 1497, pp. 21–30 (2015)
23. van Zee, M.: Rational architecture = architecture from a recommender perspective. In: Proceedings of the International Joint Conference on Artificial Intelligence (2015)
24. van Zee, M.: Rational Architecture: Reasoning about Enterprise Dynamics. Ph.D. thesis, University of Luxembourg (2017, to appear). 5
25. van Zee, M., Dastani, M., Doder, D., van der Torre, L.: Consistency conditions for beliefs and intentions. In: Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning (2015)
26. van Zee, M., Dastani, M., Shoham, Y., van der Torre, L.: Collective intention revision from a database perspective. In: Collective Intentionality Conference, July 2014

27. van Zee, M., Doder, D.: AGM-style revision of beliefs and intentions. In: Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016), September 2016
28. van Zee, M., Doder, D., Dastani, M., van der Torre, L.: AGM revision of beliefs about action and time. In: Proceedings of the International Joint Conference on Artificial Intelligence (2015)
29. Veldhuijzen van Zanten, G., Hoppenbrouwers, S., Proper, H.: System development as a rational communicative process. J. Systemics Cybern. Inform. **2**(4), 47–51 (2004)
30. Wilson, C., Short, J.: Magic Quadrant for Enterprise Architecture Tools. Technical Report ID Number: G00207406, Gartner, October 2010
31. Zachman, J.: A framework for information systems architecture. IBM Syst. J. **26**(3), 276–292 (1987)

http://www.springer.com/978-3-319-60041-3

Advances in Artificial Intelligence: From Theory to Practice
30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2017, Arras, France, June 27-30, 2017, Proceedings, Part I
Benferhat, S.; Tabia, K.; Ali, M. (Eds.)
2017, XXIX, 642 p. 183 illus., Softcover
ISBN: 978-3-319-60041-3