Modeling Decisions for Collaborative Enterprise Architecture Engineering

Dierk Jugel[1,2(✉)], Christian M. Schweda[1], and Alfred Zimmermann[1]

[1] Reutlingen University, Reutlingen, Germany
{dierk.jugel, christian.schweda, alfred.zimmermann}@reutlingen-university.de
[2] Rostock University, Rostock, Germany
dierk.jugel@uni-rostock.de

Abstract. New or adapted digital business models have huge impacts on Enterprise Architectures (EA) and require them to become more agile, flexible, and adaptable. All this changes are happening frequently and are currently not well documented. An EA consists of a lot of elements with manifold relationships between them. Thus changing the business model may have multiple impacts on other architectural elements. The EA engineering process deals with the development, change and optimization of architectural elements and their dependencies. Thus an EA provides a holistic view for both business and IT from the perspective of many stakeholders, which are involved in EA decision-making processes. Different stakeholders have specific concerns and are collaborating today in often-unclear decision-making processes. In our research we are investigating information from collaborative decision-making processes to support stakeholders in taking current decisions. In addition we provide all information necessary to understand how and why decisions were taken. We are collecting the decision-related information automatically to minimize manual time intensive work as much as possible. The core contribution of our research extends a decisional metamodel, which links basic decisions with architectural elements and extends them with an associated decisional case context. Our aim is to support a new integral method for multi-perspective and collaborative decision-making processes. We illustrate this by a practice-relevant decision-making scenario for Enterprise Architecture Engineering.

Keywords: Enterprise architecture · Decision modeling · Collaborative decision-making process

1 Introduction

The constant change in modern business models requires an enterprise to continuously adapt its business processes, its information systems, and the underlying IT infrastructure, which together form the enterprise architecture (EA). Different stakeholders, e.g. from senior management, but also from IT-operations take different perspectives on the EA and have different concerns in the evolution of the EA. Collaborative support by current developments in social and collaborative media and in adaptive case management has inspired the upcoming wave of collaborative enterprise architecture. The main

A. Persson and J. Stirna (Eds.): CAiSE 2015 Workshops, LNBIP 215, pp. 351–362, 2015.
DOI: 10.1007/978-3-319-19243-7_33

challenging recommendations from [1] for a successful collaborative EA are: establish a lean set of processes and rules instead of overloading stakeholders with bureaucratic processes and unsolicited artifacts, adopt evolutionary problem solving instead of extensively blueprinting the future rigidly, and foster and moderate open participation in decisions on the ground instead of relying on experts and top-down wisdom.

Decision processes in such complex environments have to balance these various concerns and stakeholder collaboration is a prerequisite therefore. It is nevertheless very difficult to reflect on the decision processes after the fact, especially to revisit on the perspectives that had been considered and the analysis techniques that had been applied.

These deficiencies are imminent also in the closely related field of software architecture. Tang et al. [2] elicited in a survey, that capturing decisions and rationale is very important. Without information on decisions and rationale, stakeholders find it difficult to understand the architecture. This is also already reflected in the definition of the term architecture provided in the broader context of the ISO Standard 42010 [3]: the architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and to the environment, and the **principles** guiding the decision and evolution. In particular these principles are often not documented explicitly, but implicit in the reasoning and rationale behind architecture decisions. In this light, another observation of Tang et al. [2] can be considered critically as well: stakeholders tend to forget their decisions over time. Combined with the fact that architecture decisions have a long-term impact, but often short-term contracted consultants are involved in decision-making, a significant part of the knowledge about an architecture gets lost over time.

EA Management is in its very nature a knowledge-intense activity and as such the decision-making processes are not formal, but driven by ad hoc information that is collected during the decision-making process and by assumptions made in the stakeholders' discussions. The stakeholder-specific views showing different perspectives on the EA are in this vain a starting point for decision making, but are enriched with additional information during the process. This yields a threefold challenge for EAM decision-making:

- Support different stakeholders in collaborative decision-making needed to balance different concerns.
- Capture the discussions and argumentation that lead to a distinct decision needed to facilitate organizational learning.
- Document the decision taken and its impact in various perspectives.

At the same time, in particular the capturing of discussions and argumentations should be automated as far as possible, as manual documentation after the fact is costly and time consuming.

In this paper, we present a method for documenting the decision-making processes in EAM. The method builds on the well-established work in the field of EAM, revisited in Sect. 2. In particular, we build on the EA Anamnesis method of Plataniotis et al. [4], which we extend in Sect. 3 to a method for multi-perspective and collaborative decision-making. A collaborative decision-making scenario in Sect. 4 exemplifies the method and its modeling concepts. Final Sect. 5 concludes our findings, reflects on limitations of the approach and outlines future streams of research.

2 Related Work

ArchiMate [5] is a modeling language that also defines a notation for visualizing Enterprise Architectures. The underlying metamodel comprises several concepts on different layers and relationships between them. In its core ArchiMate focuses on modeling a static state of the EA, being the current or an intended future state. Two extensions to ArchiMate address more transformation-oriented aspects of EAM.

The motivational extension [5] allows to model incentives and reasons for EA design. A *Stakeholder* is associated with a so-called *Motivational Element*. There are several characteristics of such an element. A *Motivational Element* e.g. can be a *Driver*, *Goal*, *Principle*, *Requirement* or *Constraint*. These concepts are related with each other and are useful to capture the reasons of designing EAs.

The implementation and migration extension to ArchiMate [5] describes EA transformations in more detail. Thereby a *Gap* describes the differences of two states. The states are defined as *Plateau*. A *Plateau* is realized by *Deliverables*. The detailed work that has to be done is modeled with *Work Packages*. *Work Packages* are similar to a project that has *Deliverables* as outcome.

While both extensions deal with decisions, either from the side of their motivations or the side of their impact on the architecture, an explicit decision concept focusing on the decision-making process, is missing.

In a similar manner, Buckl et al. discuss motivation and impact of EA transformations [6]. They combine the approach of Aier et al. described in [7] for modeling Work Packages and their impact on the EA with the i* modeling method [8] for describing motivations. While the employed modeling of transformations can be regarded as a refinement of implementation and migration extension, and i* modeling of motivation provides a refined perspective on e.g. soft goals, the approach does not provide an explicit concept for reflecting decisions.

Plataniotis et al. recognize the insufficiencies of above approaches and describe in [4] an approach called EA Anamnesis focusing on ex-post modeling Enterprise Architecture decisions and the decision-making process. They develop a metamodel for this purpose. An *EA Issue* represents the starting point of a decision-making process. It describes the design problem that has to be addressed. The *EA Decision* is described as a representation of a design decision that is taken. *EA Decisions* result in an *EA Artifact*. The *EA Artifact* is defined as result produced by an *EA Decision*. Furthermore it is described as representation of the result. The intention is to use this concept to relate an *EA Decision* with a visual representation of ArchiMate. In addition an *EA Decision* is associated with a *Layer* of ArchiMate.

Furthermore the approach provides four different relationships to relate *EA Decisions* with each other. *EA Decisions* can be translated into others using the *Translation Relationship*. This relationship enables deriving *EA Decisions*. For instance, a stakeholder takes a decision on Business Layer. This decision has impacts on the underlying Application Layer. Therefore the *EA Decision* on Business Layer has to be translated into an *EA Decision* on Application Layer and so on. Using *Decomposition Relationships* enables decomposing into more detailed ones. Before a decision is taken, the responsible usually has to choose between several alternatives. Such alternatives between *EA Decisions* can be modeled using the *Alternative Relationship*. Thus *EA Decisions* may have negative impacts on the Enterprise Architecture, the *Substitution Relationship* enables modeling how these negative impacts can be repaired. Furthermore the approach comprises concepts to model strategies for decision-making. These strategies e.g. include requirements that form a framework for decision-making. For instance, there are standards determined by governance processes that have to be involved in decision-making.

While the approach of Plataniotis et al. provides a significant contribution in the field, it is limited with respect to the challenges outlined in Sect. 1. Firstly, the authors assume that a single stakeholder takes the decisions and the proceeding is straightforward. In practice many stakeholders are involved in decision-making. Secondly, the approach's intention is an ex-post documentation of decisions. While this is possible, we regard it to be time-consuming and not adequate in particular in a collaborative setting. Thirdly, an *EA Decision* is associated with exactly one layer of ArchiMate. The layers describe non-overlapping subsets of the EA. We assume that complex interdependencies within the EA make it impossible to confine a decision to a single layer.

3 Collaborative Decision Modeling

In this section we present an adapted metamodel based on Plataniotis et al. [4] targeting collaborative decision modeling. We further adopt relevant concepts from the ArchiMate implementation and migration extension [5] to more clearly elaborate on the impact of a decision. Especially in the field of EAM, where typically several stakeholders are involved, (1) collaboration support, (2) multiple viewpoints and (3) early documentation are necessary during the decision-making process. The adapted metamodel is shown by Fig. 1. Therein, we highlight added concepts, whereas the other concepts are adopted from [4].

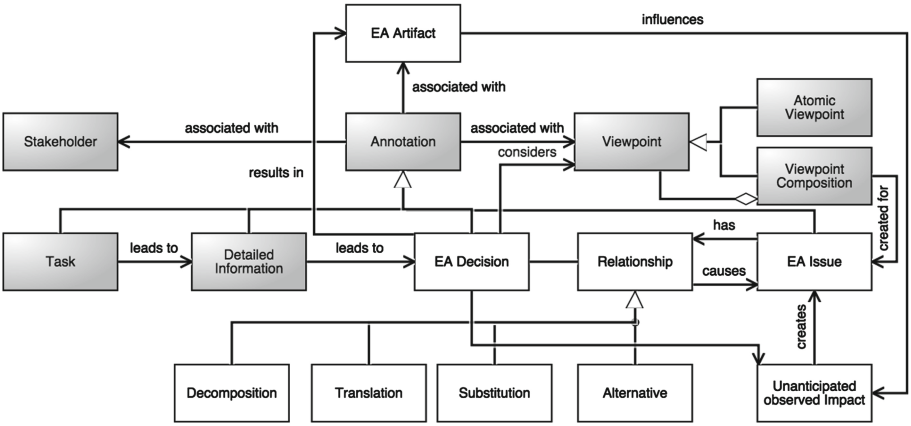

Fig. 1. Collaborative decision metamodel

A key differentiation to Plataniotis et al. [4] is the relationship from *EA Decision* to a *Viewpoint*. In their work, the authors relate an *EA Decision* to a particular *Layer* of ArchiMate [5]. Layers can be regarded as specific viewpoints in line with the ISO Std. 42010 [3], i.e. as "work product establishing the conventions for the construction, interpretation and use of architecture views". The layers are nevertheless specific as they target non-overlapping subsets of the EA. Like described in Sect. 2 complex interdependencies within the EA make it impossible to confine a decision to a single layer. Thus, we understand a decision to consider different, potentially overlapping viewpoints, instead of layers. In [9] we presented an interactive cockpit approach using a viewpoint composition consisting of several coherent viewpoints. Analyzing EAs using a viewpoint composition that can be considered in parallel supports stakeholders in the decision-making process and prohibits loosing the overall context. Thus we distinguish two types of viewpoints: Atomic Viewpoint and Viewpoint Composition.

Atomic Viewpoint: An *Atomic Viewpoint* is a single Viewpoint.
Example: "Application Usage Viewpoint" (see [5]).

Viewpoint Composition: A *Viewpoint Composition* forms a composite structure and consists of coherent Atomic Viewpoints or other *Viewpoint Compositions* needed by stakeholders to satisfy their information demands. Furthermore a *Viewpoint Composition* is assembled to consider an *EA Issue*.
Example: A *Viewpoint Composition* considering the *EA Issue* "Outdated Technologies" consisting of several *Atomic Viewpoints*, like "Application Usage Viewpoint" or "Infrastructure Usage Viewpoint" (see [5]).

We extend the metamodel of Plataniotis et al. [4] with the concept of the stakeholder – in line with the ISO Std. 42010 [3] – reflecting a person, role, or group having a concern in the EA and hence being involved in the decision-making process.
Example: Software Engineer, Business Process Owner, or Technology Expert.

Stakeholders provide different contributions during a decision-making process: they raise *EA Issues* to be addressed, provide relevant *Detailed Information* on the EA not covered by the model, assume responsibility for *Tasks*, or take *EA Decisions*. *EA Issues* and *EA Decisions*, as already present in the model of Plataniotis et al. [4], very much reflect the ex-post perspective of decision-making, whereas Detailed *Information* and *Tasks* are intermediary in the process. By making them explicit, we provide a means to document the rationale of a decision.

Detailed Information: Detailed Information enables modeling considerations, discussions and findings during the decision-making process that are necessary for taking a decision.
Example: Stakeholder A wants to highlight three architectural elements and assign them with a comment, because the stakeholders discuss about these elements.

Task: By using the *Task* concept open questions can be documented that have to be clarified by a stakeholder until the next meeting. A *Task* leads to *Detailed Information*.
Example: Stakeholder B has to identify the responsible for a particular part of the EA that they want to retire.

The different stakeholders' contributions in the decision-making process, ranging from *EA Issues*, over *Detailed Information* and *Tasks*, to *EA Decisions*, impact specific parts of the overall EA. In [4] these "parts" are alluded to as *EA Artifacts*. The contributions on the other hand are contextual, i.e. depend on the decision-making process in which they are created. Hence, we identify the contributions with the concept of the *Annotation* as introduced in [9]. This approach uses annotations to enrich the EA description with additional knowledge. We defined several interactive functions that support stakeholders in analyzing and planning an EA. An impact analysis is an example for such an interactive function. Each function enriches the description of the EA with an *Annotation* that explicitly represents analyzing and decision-making information. Accordingly, *Annotations* are associated with stakeholders, who performed the function and elements of the EA, i.e. the *EA Artifacts* of [4].

In the following we want to detail the concept of *EA Decisions*. Plataniotis et al. define an *EA Artifact* as a result of an *EA Decision*. We want to detail this definition. Aier et al. [7] describe three types of changing elements of an EA. Thereby elements can be introduced, retired or optimized. In other words an *EA Decision* is based on current valid *EA Artifacts* (as-is landscape) and transfers them into *EA Artifacts* in the future (to-be landscape). According to this approach we refine the "results in" relationship to be these three types. Moreover *EA Artifacts* need to have a period they are valid. Moreover *EA Artifacts* are affected by an *EA Change* and need to have a period they are valid. Thus this period is independent of an introduction or retirement, we added a particular date for both concepts. Through the relative independency of lifecycle planning information, which is manifested by the validity attributes, and the decision related controlling information of changes there could be three scenarios and their possible combinations: (1) consistent decisions of change in accordance with the specified lifecycle, (2) premature introduction of *EA Artifacts* before starting the lifecycle and (3) deferred retirement of *EA Artifacts* after the specified artifact" lifecycle. Figure 2 shows these metamodel changes in detail.

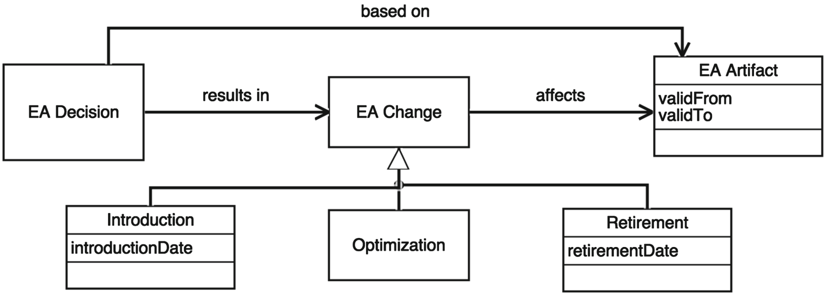

Fig. 2. EA Decision Consequences on EA Artifacts

The metamodel of [4] also includes a concept named *Unanticipated observed impact*. However such impacts can only be documented after taking a decision, because such impacts can only be found after implementation. Furthermore the authors describe decision strategies to document which criteria are used to take the decision.

To support stakeholders in decision-making, in line with Plataniotis et al. [10], we suggest a Decision Viewpoint that provides all collected information during the decision-making process and the current state. At the end the Decision Viewpoint can be used as documentation and stored in a central knowledge base. The knowledge base affords stakeholders to learn about decisions from the past to take better decisions in the future. Another advantage of such a knowledge base is that new employees get a fast insight about the architecture and what the reasons are how the architecture looks like. An example of such Decision Viewpoint is exemplarily described in Sect. 4.

4 Collaborative Decision-Making Scenario

In this section we adapt the example case of Plataniotis et al. [4] to illustrate our collaborative decision modeling approach. The example case is based on ArchiSurance that is a virtual insurance company introduced to demonstrate the capabilities of ArchiMate [5].

ArchiSurance hired an external consultant named John. He is a business consultant with experience in changing business models. His mandate is to change ArchiSurance's sales model towards an intermediate one. As changing business models has impact on the underlying applications and the IT infrastructure, he involve Mike – an application responsible – and Jack – an infrastructure and security expert. They both have particular concerns and need specific viewpoints to satisfy their information demands, i.e., to assess the impact of the business model change.

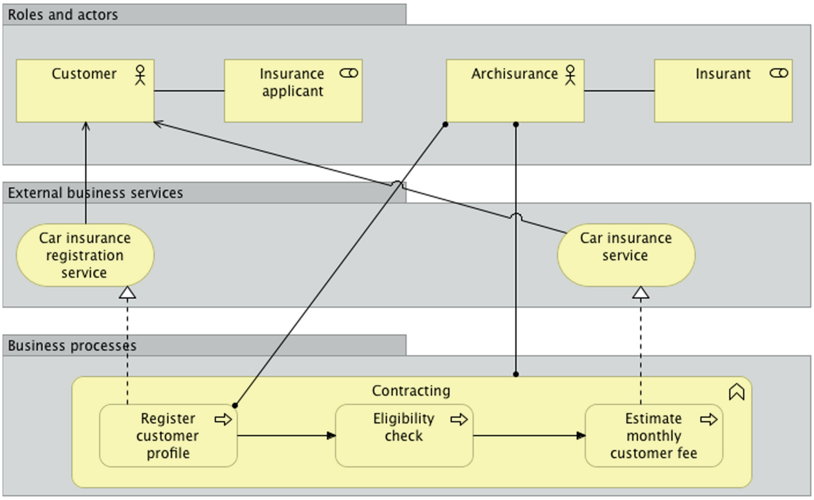

Fig. 3. Business Process Viewpoint

To analyze the current state of the EA and to generate implementation alternatives, they meet at an Architecture Cockpit like described by Jugel et al. [9]. The cockpit provides a *Viewpoint Composition* consisting of three different ArchiMate viewpoints

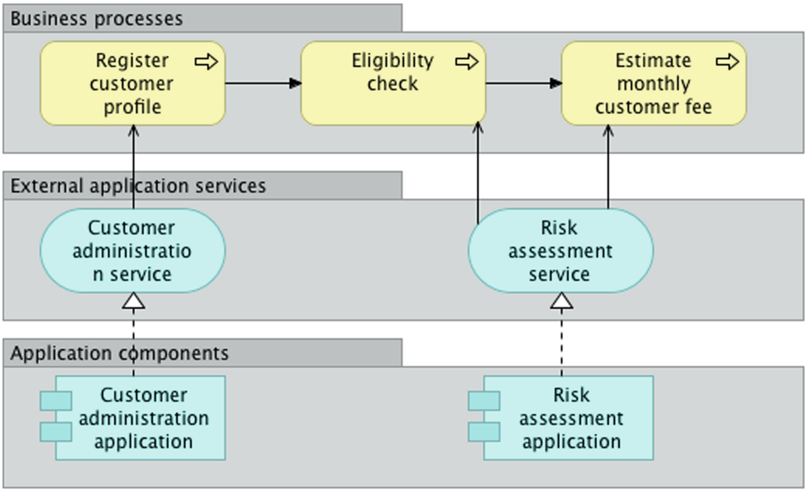
Business processes
Register customer profile
Eligibility check
Estimate monthly customer fee
External application services
Customer administratio n service
Risk assessment service
Application components
Customer administration application
Risk assessment application

Fig. 4. Application Usage Viewpoint

side-by-side, namely the Business Process Viewpoint, the Application Usage Viewpoint and the Infrastructure Usage Viewpoint. Exemplarily Fig. 3 shows the Business Process Viewpoint and Fig. 4 the Application Usage Viewpoint. The Business Process Viewpoint shows the overall context about ArchiSurance's sales model. By using this information, John wants to adapt the sales model. The Application Usage Viewpoint describes the applications, their application services and the business processes they support. Lastly the Infrastructure Usage Viewpoint required by Jack describes the dependencies between the infrastructure and the hosted applications.

In the example case, the starting point is an *EA Issue*, namely to change the sales model to enable intermediary. Thus John creates an *EA Issue* named "Intermediary sales model" and uses the Business Process Viewpoint to identify the *EA Artifact* that is affected by the *EA Issue*. This is the business function "Contracting", which he indicates by adding *Detailed Information* named "Main affected business function". As a starting point for the discussion, this business function is further highlighted in the business process viewpoint. As no information, on who is the responsible of this business function, is given, John creates a *Task* named "Clarify responsibilities" setting a schedule to the next meeting, because this is an additional stakeholder that has to be involved.

Afterwards John suggests his idea by creating the *EA Decision* "Changing the sales model" that provides a solution for previously created *EA Issue*. John envisions that the business process "Register customer profile" is separated from the business function "Contracting". Instead he recommends creating a new business function named "Create customized insurance package". The results of his suggestion can be modeled according to [7] by relating the impacted *EA Artifacts* by one of the defined relationship types. For example, a new *EA Artifact* named "Create customized insurance package" of the type business function is needed that has to be related to the *EA Decision* by an

Introduction concept. Furthermore, the business process “Register customer profile” has to be retired to realize the new demand by superseding it with a new business interaction named “Customer profile registration”. Mike recognizes that changing the business process “Register customer profile” is the only change in the business model that may have impacts on underlying applications and infrastructure. He documents this finding by adding *Detailed Information* named “May have impacts on applications and infrastructure” and relating it with the business process “Register customer profile”.

The state of the discussion after these initial considerations is represented in Fig. 5. The figure uses color-coding and comment-like symbols to indicate the different decisions. Thereby elements with a green fill color represent *EA Artifacts* that have to be introduced whereas elements that have to be retired are represented with a red fill color. Furthermore, the *EA Artifact* with the blue fill color represents the starting point and bears the *Detailed Information* “Main affected business function” described above.

Next they analyze the impacts of the business process change on application services and applications by using the impact analysis function. Thereby affected elements are assigned to new *Detailed Information* named “Impact Analysis of Register Customer profile” representing the analysis result and are highlighted on the views with a brown fill color. The result is, that the application service “customer administration service” has to be adapted to the new situation.

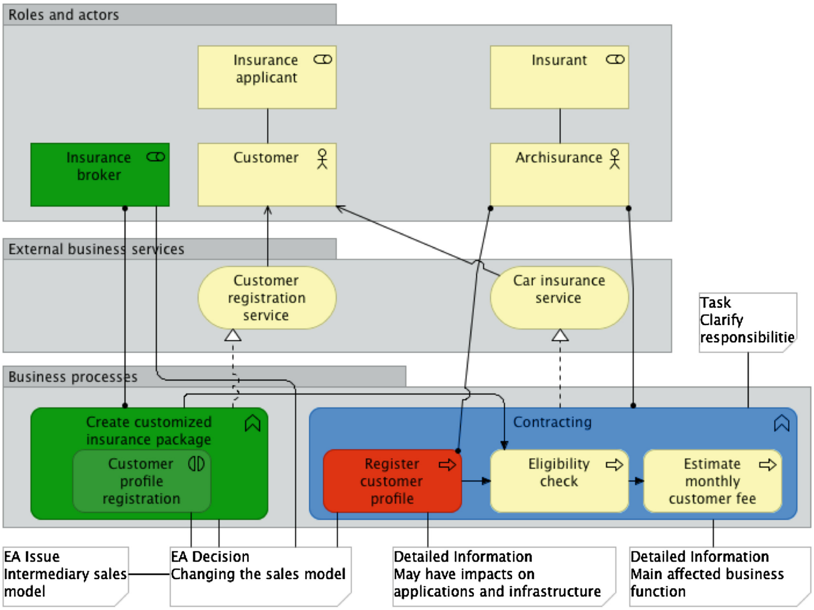

Fig. 5. Business Process Viewpoint (Sales model adaptions) (Color figure online)

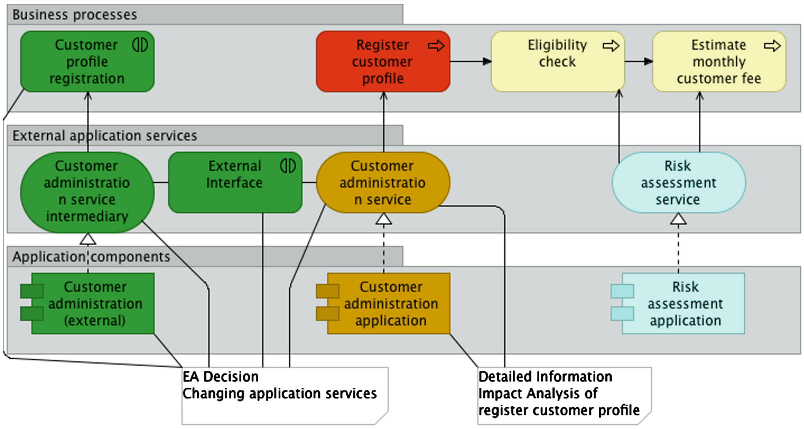
Business processes
Customer profile registration
Register customer profile
Eligibility check
Estimate monthly customer fee
External application services
Customer administratio n service intermediary
External Interface
Customer administratio n service
Risk assessment service
Application components
Customer administration (external)
Customer administration application
Risk assessment application
EA Decision
Changing application services
Detailed Information
Impact Analysis of
register customer profile

Fig. 6. Application Usage Viewpoint (with adaptations)

Mike recommends retaining the application service, but to use an external application service named “Customer administration service intermediary” that provides customer information from intermediary. An interface is created to link the two application services, enabling the external service to provide customer information to the internal one. Mike documents his idea by creating a correspondent *EA Decision* “Changing application services” according to the described procedure above. Thereby the *EA Decision* “Changing the sales model” is translated into the *EA Decision* “Changing application services”. Figure 6 illustrates the state of the discussion regarding the application landscape.

Jack, the infrastructure and security expert, is concerned about security issues. He documents his concerns by adding *Detailed Information* named “Security Issues”. He demands to consider all impacts on the underlying infrastructure and hence they perform an impact analysis choosing the application “Customer administration application” as starting point. Based on the results, Jack suggests some changes at the infrastructure. He recommends adding a new firewall that protects the application service “Customer administration service” within the enterprise’s LAN from undesired external access. The group agrees and they take an *EA Decision* named “Adding a Firewall” that is translated from the *EA Decision* “Changing application services”.

Concluding they discuss whether the proposed and taken *EA Decisions* should be realized. They decide to discuss the implications with other employees from their respective departments, ask them for opinions and possible alternatives that should be taken into account. To facilitate these discussions, they apply the Decision Viewpoint that shows all relevant information during the process. In this example case the Decision Viewpoint (see Table 1) is illustrated using a table. Alternatively decision graphs like described in [10] can be used.

Table 1. Decision Viewpoint

Decision Viewpoint		
Stakeholder	Annotation	EA Artifacts & Viewpoints
John	EA Issue “Intermediary sales model”	Business Process Viewpoint
John	Information “Main affected business function”	Contracting
John	Task “Clarify responsibilities”	Contracting
John	EA Decision “Changing the sales model”	**Introduce:** Create customized insurance package, Customer profile registration, Insurance broker **Retire:** Register customer profile
…	…	…

5 Conclusion and Outlook

Our research raised the question how a framework for supporting collaborative decision-making in the EA context looks like. We have analyzed and extended related work about fundamental decisional support for enterprise architectures by introducing our generic decisional metamodel for collaborative enterprise architecture engineering. To close the connection between the *EA Decision* and the *EA Artifact* we have extended related work results and mapped our decisional metamodel to the viewpoint-supported Enterprise Architecture approach form ArchiMate and other standards. The illustration of a working decision-making scenario aims to show the usage of our decisional metamodel into a real-life EA scenario. We are planning to validate our decisional framework in our current and future research with selected industrial partners. Currently we are performing case studies with enterprise architects and students in the architecture cockpit at Reutlingen University. We have to investigate in our future research and work out the limitations of the approach. Future work should also respect extended decisional contexts, like the decisional case and inputs from adaptive case management and new elements from knowledge management. Another important idea we are currently researching is about connecting architectural decisions with rationales and in a planned extension with explanations and semantic-supported inferences for architectural impacts.

References

1. Bente, S., Bombosch, U., Langade, S.: Collaborative Enterprise Architecture. Morgan Kaufmann Elsevier, Waltham (2012)
2. Tang, A., Babar, M.A., Gorton, I., Han, J.: A survey of architecture design rationale. J. Syst. Softw. **79**(12), 1792–1804 (2006)

3. International Organization Of Standardization: ISO/IEC/IEEE 42010:2011 - Systems and software engineering - Architecture description (2011)
4. 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. **4**(1), 75–95 (2014)
5. The Open Group: ArchiMate 2.0 Specification (2012)
6. Buckl, S., Matthes, F., Schweda, C.M.: Socio-technic dependency and rationale models for the enterprise architecture management function. In: 5th International Workshop on Ontology, Models, Conceptualization and Epistemology in Social, Artificial and Natural Systems (ONTOSE 2011), London (2011)
7. Aier, S., Buckl, S., Gleichauf, B., Matthes, F., Schweda, C.M., Winter, R.: Towards a more integrated EA planning: linking transformation planning with evolutionary change. In: 5th International Workshop on Enterprise Modelling and Information Systems Architectures, Hamburg (2011)
8. Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, Annapolis, pp. 226−235 (1997)
9. Jugel, D., Schweda, C.M.: Interactive functions of a cockpit for enterprise architecture planning. In: 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Ulm, pp. 33−40 (2014)
10. Plataniotis, G., De Kinderen, S, Proper, H.A.: Relating decisions in enterprise architecture using decision design graphs. In: 17th IEEE International Enterprise Distributed Object Computing (EDOC), Vancouver, pp. 139−146 (2013)