Capturing Design Rationales in Enterprise Architecture: A Case Study

Georgios Plataniotis[1,2,4], Sybren de Kinderen[3,4], and Henderik A. Proper[1,2,4]

[1] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
[2] Radboud University Nijmegen, Nijmegen, The Netherlands
[3] University of Luxembourg, Luxembourg
[4] EE-Team, Luxembourg, Luxembourg*
{georgios.plataniotis,erik.proper}@tudor.lu,sybren.dekinderen@uni.lu

Abstract. We aim for rationalizing Enterprise Architecture, supplementing models that express EA designs with models that express the decision making behind the designs. In our previous work we introduced the EA Anamnesis approach for architectural rationalization, and illustrated it with a fictitious case study.

In this paper we evaluate our approach in terms of its ability to capture design rationales in the context of a real life case study. Together with stakeholders from the business and IT domains of a Luxembourgish Research and Technology Organization, we captured the design rationales behind the introduction of a new budget forecast business process. Our case study shows that EA Anamnesis can reflect the design rationales of the stakeholders, also linking business and IT concerns. Furthermore our study shows that, for this particular case, the stakeholders often used heuristics (commonsensical "short cuts") to make their decision, or even made decisions without considering alternative choices. Finally, we discuss what the lessons learned from this case imply for further research.

Keywords: Enterprise Architecture, Design Rationale, Design Decisions, Case Study.

1 Introduction

As architects create blueprints for (re-)designing buildings, enterprise architects use EA modeling languages for (re-)designing organizations [1]. They do so by relating the business and IT concerns of an organization. For example, EA modeling languages can be used to design an IT application landscape suitable for a particular business process. Prominent examples of EA languages are the Open Group standard ArchiMate [2], and the recent OMG standard Unified Profile for DoDAF/MODAF (UPDM) [3], an UML profile for describing enterprise architecture in accordance with the enterprise architecture frameworks DoDAF/MODAF.

* The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, Radboud University Nijmegen and HAN University of Applied Sciences (www.ee-team.eu)

U. Frank et al. (Eds.): PoEM 2014, LNBIP 197, pp. 133–147, 2014.

Yet, EA modeling languages describe the EA designs, but not the reasoning behind these designs.This also holds for the recent motivation extension of the EA modeling language ArchiMate [2]. While the motivation extension allows for expression stakeholder intentions, it lacks well-established decision making concepts such as criterion, used decision making strategy and more. Moreover, in some cases practitioners have a different understanding even for the same EA model because they interpret the meta-conceptual constructs with different ways [4].

Experience from the field of software architecture shows that leaving design rationales implicit leads to 'Architectural Knowledge vaporization' (cf. [5]). This means that, without design rationale, design criteria and reasons that lead to a specific design are not clear. Also, alternatives that were considered during the design process are not captured.

Among others, a lack of transparency regarding design decisions can cause design integrity issues when architects want to maintain or change the current design [6]. This means that due to a lacking insight of the rationale, new designs are constructed in an ad-hoc manner, without taking into consideration constraints implied by past design decisions. Furthermore a survey on EA rationalization amongst EA practitioners [7] suggests the relevance of architectural rationalization for motivating design decisions, and for architectural maintenance. However, the same survey shows that practitioners often forego the use of a structured template/approach when rationalizing an architecture. Instead they capture decision characteristics in an ad hoc manner, and do so largely in plain text.

In our earlier work [8,9,10] we introduced the EA Anamnesis approach for architectural rationalization. EA Anamnesis captures decision characteristics such as decision criteria and used decision making strategy, and shows the relation between business-level and IT-level decisions. Furthermore, EA Anamnesis allows for a formal linkage to metamodel-based EA artifacts, thus allowing for a bridge between languages for EA design (basically ArchiMate) and the corresponding design rationale.

Thus far, EA Anamnesis has been developed with the aid of a fictitious case study, and with a survey amongst practitioners [7]. The fictitious case helped for idea development, while the survey provided us a first practical assessment of the EA Anamnesis's rationalization concepts. However, none offered us an *in-depth* assessment of the practical applicability of EA Anamnesis. In particular, we lack substantial insight into the extent to which EA Anamnesis can express real life decisions.

As a response, in this paper we apply our approach to a real world case in a Research and Technology Organization. Together with two stakeholders, from the financial and IT domain respectively, we extracted the design rationales behind the introduction of a new budget forecasting business process. This helps us identify how practitioners perceive the concepts of EA Anamnesis for capturing and understanding enterprise architectures. Moreover we observe that, for this particular case, practitioners select among alternatives by using simple decision making processes. Even more so, practitioners do not consider alternatives during their decision making process.

This paper is structured as follows. Section 2 presents the EA Anamnesis approach, Section 3 introduces the Research and Technology Organization Case Study and discusses the case study protocol we followed, the limitations and the capturing of design rationales with our approach. Section 4 presents lessons learned. Section 5 concludes.

2 EA Anamnesis Approach

Fig. 1 presents the EA Anamnesis metamodel as discussed in [8,9,11]. With this metamodel we allow for (1) contextualizing the decision making process of a single decision in terms of cross cutting/intertwining decision relationships, and (2) a comparison of decision outcomes to the original decision making process.

For comprehension purposes the concepts of our metamodel will be introduced in 3 subsections: decision properties (Subsect. 2.1), decision making process concepts (Subsect. 2.2) and decision relationships (Subsect. 2.3).

2.1 Decision Properties

EA Decision: We define decision as the choice made between alternative courses of action in a situation of uncertainty [12]. Moreover, an enterprise architecture (EA) decision names the decision that is made in the context of an Enterprise Transformation [13]. Regarding the distinction between made decision and alternative decision, see the decision relationship“alternative”.

EA Issue: Similar to the concept of an issue from [14], an EA issue represents the architectural design problem that enterprise architects have to address during the Enterprise transformation process.

EA Artifact: An EA artifact (similar to concept of an architecture element [6]) is either the direct result produced from a set of executed EA decisions, or a representation of this result. For now, we use an EA artifact to refer to architectural representations. Specifically, we use it as a bridging concept towards the EA modeling language ArchiMate, whereby an EA artifact allows us to link EA decisions to concepts from ArchiMate.

Layer: In line with the ArchiMate language [2], an enterprise is specified in three layers: *Business, Application and Technology.* Using these three layers, we express an enterprise *holistically*, showing not only applications and physical IT infrastructure (expressed through the application and technology layers), but also how an enterprise's IT impacts/is impacted by an enterprise's products and services and its business strategy and processes.

Observed Impact: The observed impact concept signifies an *unanticipated* consequence of an already made decision to an EA artifact. This opposes to anticipated consequences, as indicated by relationships such as translation or decomposition. Observed impacts can be positive or negative.

In current everyday practice, architects model *anticipated* consequences using what-if-scenarios [1]. Unfortunately, not every possible impact of made EA

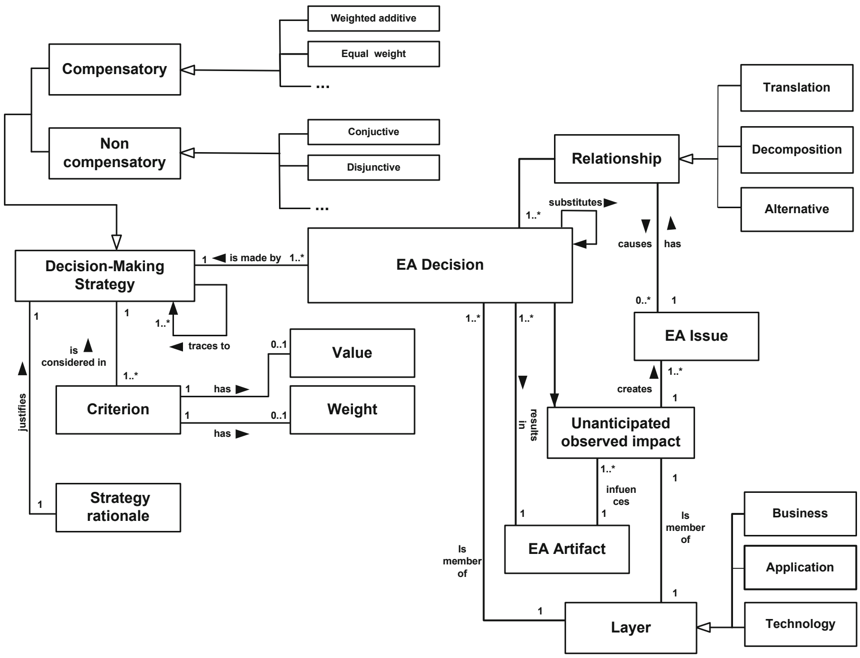

Fig. 1. EA Anamnesis metamodel

decisions can be predicted. This is especially true for enterprise architecture, where one considers impacts across the enterprise rather than in one specific (e.g. technical) part. The outcome of EA decisions can be observed during an ex-post analysis of the architecture [13]. Some of the consequences of EA decisions are revealed during the implementation phase, or during the maintenance of the existing architecture design. These unanticipated consequences are captured exactly by the concept of an observed impact.

For us the main usefulness of capturing observed impacts is that they can be used by architects to avoid decisions with negative consequences in future designs of the architecture.

2.2 Decision Making Process Concepts

The decision making process concepts of our metamodel focus on capturing (1) decision making strategies that were used during the architectural design process for a specific EA decision, (2) the rationale behind this specific decision strategy choice, and (3) available alternatives and criteria that were taken into account. Below we provide the description of these concepts.

Decision-Making Strategy: This concept captures the decision making strategy used by the enterprise architect to (1) evaluate the alternatives, and make the actual EA decision. As we analyzed in our previous work [8], decision strategies are characterized as compensatory, noncompensatory, or as a hybrid of these two. A hybrid decision strategy is also supported by our metamodel. The relationship ‘trace to’ signifies the combination of two or more decision strategies during the decision making process.

Criterion: Criteria play an important role in our metamodel. Depending on the decision strategy that was used for the evaluation process, criteria can be compensatory or noncompensatory. For example, if a disjunctive strategy was used, the criteria that were used for the evaluation with this strategy are disjunctive. Furthermore, the concepts **value** and **weight** of criterion are included in our viewpoint. The value concept represents the value that the decision maker assigns to this criterion during the evaluation process. The weight concept represents the importance of this criterion, and is typically used in WADD strategies.

Strategy Rationale: In a decision making process, the architect not only has to choose amongst some alternatives (actual decision making process), but has also to select the decision strategy that satisfies his current evaluation needs. Actually, this concept represents the rationale for the decision strategy that was selected for the evaluation process. This is what is referred as metadecision making, decision making about the decision process itself [15].

2.3 EA Decision Relationships

The role of relationship concepts is to make the different types of relationships between EA decisions explicit. Based on ontologies for software architecture design decisions [16,17], we define four types of relationships:

Translation Relationship: Translation relationships illustrate relationships between decisions/EA issues that belong to different layers/EA artifacts. Architects translate the requirements that new EA artifacts impose (EA issue) to decisions that will support these requirements by means of another EA artifact [18].

Decomposition Relationship: The Decomposition relationship is in line with ‘Comprises (Is Made of, Decomposes into)’ of Kruchten’s ontology [17]. Decomposition relationships signify how generic EA decisions decompose into more detailed design decisions in the context of a specific EA artifact.

Alternative Relationship: This relationship type [17] illustrates the EA decisions that were rejected (alternatives) in order to address a specific EA issue.

Substitution Relationship: A substitution relationship explicates how one EA decision repairs the negative outcome of another EA Decision.

3 Research and Technology Organization Case Study

In this section we describe the application of our approach to a case study of a Research and Technology Organization in Luxembourg (LuxRTO).

3.1 Case Study Setup

Objectives and Setup: The main objective of this case is to review to what extent our approach is able to capture design rationales in the context of a *real life* enterprise transformation.

To this end, we study one particular transformation: the introduction of a new budget management business process at LuxRTO. We organized interviews with two key stakeholders that were involved in the transformation: The financial officer, and the IT architect. Both these stakeholders provided a good starting point for the domain knowledge that we had to capture. On the one hand, the financial officer possessed significant business expertise on this enterprise transformation project. Being involved from the start of the transformation project, she had knowledge about the drivers that initiated this transformation and how the business process design evolved over time. On the other hand the IT architect had significant IT expertise on the transformation project. Furthermore, the stakeholders provided us with the documentation of this transformation project (text documents, presentations, emails).

We started our case study by presenting the EA Anamnesis approach to the financial officer and the IT architect. We explained the goals and challenges of our case study, and we illustrated our approach using an example case. This example case helped the stakeholders to understand our approach.

After the presentation of EA Anamnesis, we conducted a collaborative modeling exercise with the two stakeholders. The goal of this exercise was to see to what extent our approach was able to capture the design rationales of this transformation. Furthermore we also identified the perception of stakeholders regarding the concepts of EA Anamnesis.

Note that the setup above is inspired by the main steps for doing case study research set out in [19]. For example: prior to the collaborative modeling we explained our approach to practitioners. This is in line with [19], who advices to prepare for data collection prior to the collection of evidence.

Limitations: In this subsection we discuss limitations that have potentially played a role in the application of our approach in LuxRTO and in the interpretation of the results of this study.

The first limitation is that the actual enterprise transformation was held around two years before the case study. This implies that stakeholders may had a bias in what information they captured during the case study (colored memory) or they may have forgotten certain things. Another limitation is the number of stakeholders that participated in the case study. Normally, multiple stakeholders participate in an enterprise architecture transformation. In our case we interviewed two stakeholders (one from business domain and one from IT). We are aware of this restriction but in the current stage of our research we focused on how our approach captures design rationales and not on the support of multiple stakeholders decision making.

3.2 Budget Forecasting at a Research and Technology Organization

Here we present the introduction of a new budget management business process and how this process was supported by information systems in the context of an enterprise architecture transformation.

During the last years, the Luxembourgish government introduced stricter rules on the budget spending of research institutions. This policy had to be incorporated by the research institutions, meaning that the institutions should be able to establish long term financial projection plans. This would give to institutions a better awareness regarding the availability of resources and in turn the planning of future projects and personnel hiring.

LuxRTO did not have an established business process for the budget estimation. Stakeholders from the management side of LuxRTO had to design this new business process. Their initial objectives were that this business process should provide a clear view on human resources and projects coverage, an input for the future hiring plan, comparison between the forecasted and valuable budget, and in general robustness of the organization's financial data. Last but not least, a training for the users of this new business process should be organized.

3.3 Enterprise Transformation

In this part we describe how the enterprise design was changed in order to support the new budget estimation business process. For expressing the EA Design of the budget forecast project we used the ArchiMate EA modeling language. Not that LuxRTO had already established IT systems that were supporting other types financial, project and human resources business processes. Before we present the transformation we briefly describe the new business process and the already established IT systems.

Budget Forecast Business Process: The main objectives of this business process are the estimation and the planning of resources to ensure the planning activities, the assessment of the need for additional resources, the estimation of the associated budgets and the checking of the forecast in relation to the available budget in LuxRTO. The role of the business process is to provide annual budget estimates, which should be validated and approved by the finance department.

IT Systems: Application A is the main financial application of the organization. The main functionalities of this application is the management of procurements, traveling costs, personal costs, overhead costs calculation, salaries payment and project dashboard. The user access to this application in controlled and only allowed to financial officers.

Application B is the human resources management application. Tasks like resource allocation, start/end dates of work contracts, weekly calendar, different types of leaves (sickness, vacation etcetera) are executed by this application.

Application C is the project management application of the organization. The actual hours assigned per project in the organization are maintained in this application.

First Iteration of the Transformation: Fig. 2 depicts the EA model after the incorporation of the budget forecast business process. From this model we can realize that the business process was supported by the interaction and collaboration among Applications A, B, C and a spreadsheet application. However, due to some problems (which can not been described by the EA model), stakeholders had to do some additional changes in the EA design.

Second Iteration of the Transformation: Fig. 3 depicts the final iteration of the enterprise transformation. With this iteration stakeholders managed to address the aforementioned problem. Instead of using spreadsheets for entering the budget data, a new application interface was added in the financial application A.

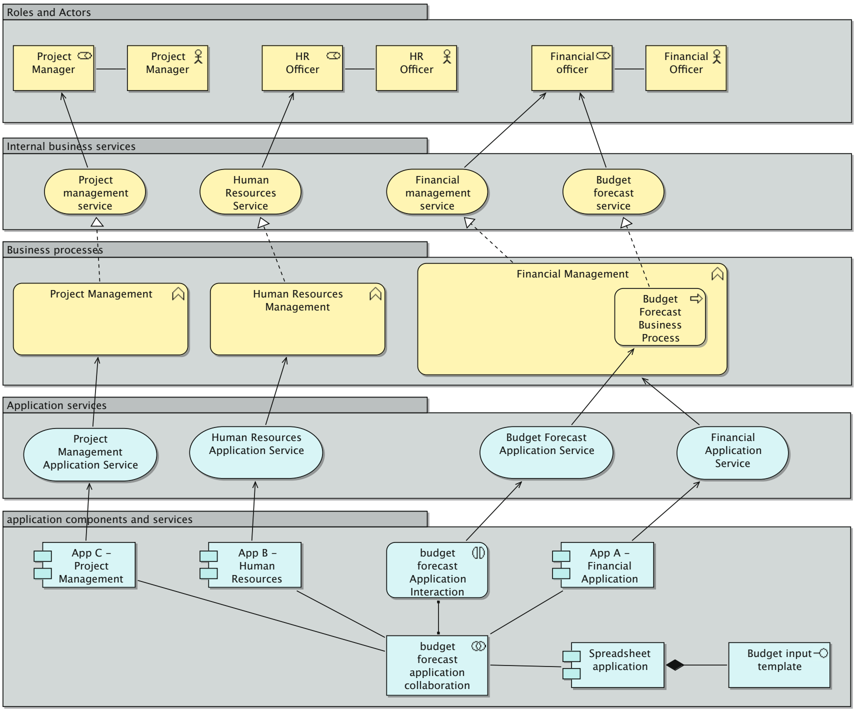

Fig. 2. LuxRTO enterprise transformation - First iteration

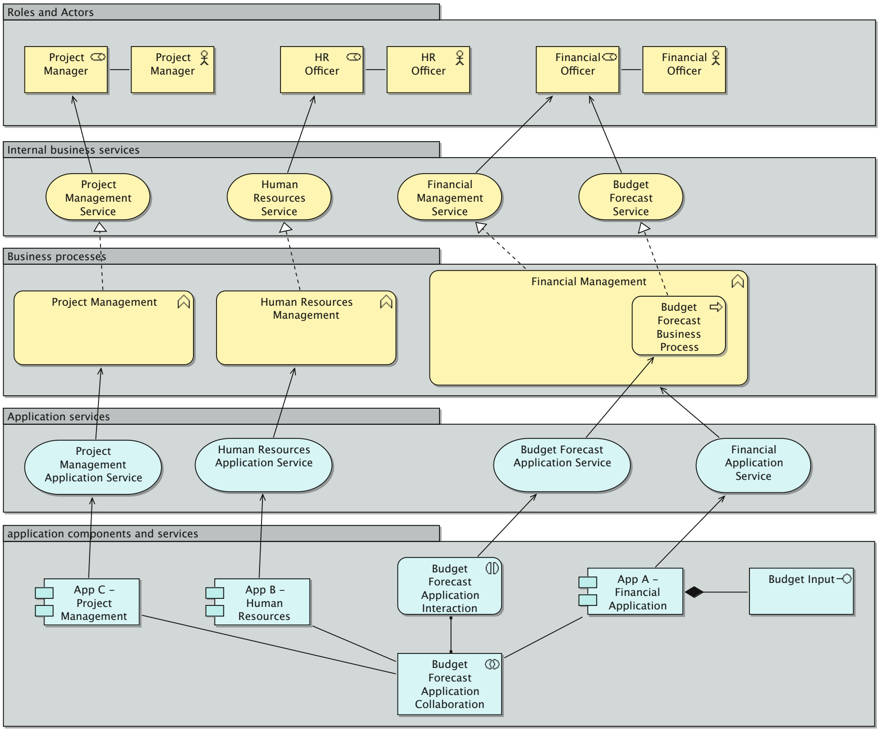

Fig. 3. LuxRTO enterprise transformation - Second iteration

3.4 Capturing the Rationale behind the Budget Forecast Design

In the previous subsection we described the changes happened in the enterprise architecture design in order to support the new budget forecast business process. However, the rationale behind this design is not captured by the EA models. Based on the case study we could potentially ask these questions:

Why these IT systems were selected for the realization of the business process? Were there any other alternatives? What were the unanticipated consequences of these decisions in the enterprise architecture?

The answers to these questions provide a useful insight in the understanding of the EA design and can not be answered just by examining EA models. This is exactly the point where EA Anamnesis approach intervenes.

Our approach uses two elements for capturing and representing design rationales. The visualization of Fig. 4 is a design decision graph which is constructed while design rationale is captured. The graph represents design decisions and how they are interrelated with other design rationale concepts (issue, observed impact etc) of the EA Anamnesis approach. The graph is accompanied with Table 1 which provides a summarization of the design rationale information.

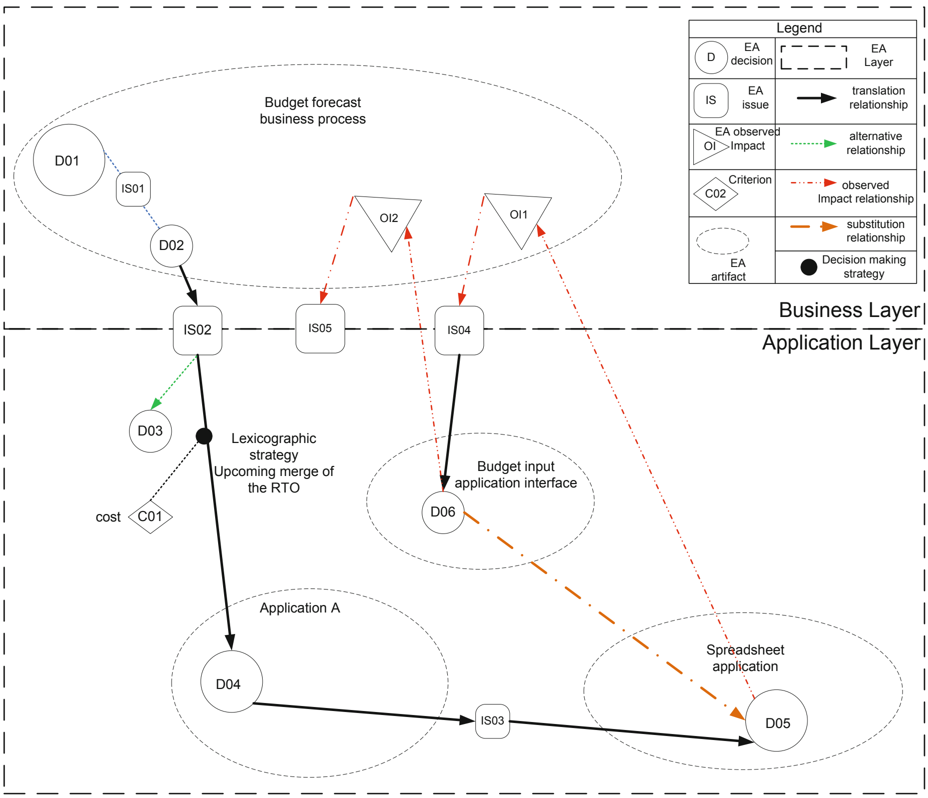

Fig. 4. Budget forecast design decision graph

We start by capturing the design decisions of the EA artifact "Budget forecast business process". EA artifacts are depicted in the decision graph as circles with dashed lines. At the same time the EA artifact "Budget forecast business process" is depicted in the EA models of Figures 2 and 3. This helps us trace design decisions, since we can start examining EA artifacts and then zooming (using the graph) in the design rationale behind the specific EA artifacts.

Decision 01 (D01) "Create budget forecast business process" is the decision that initiated the transformation. Reasons behind this decision is the business goal of having budget forecast in the long term. The execution of D01 triggered amongst others a new enterprise architecture architecture issue (IS01) "Storing budget estimation frequency". This means that the business stakeholder should define a certain frequency of storing the budget estimation per year. This issue was addressed by a newer decision (D02) "Storing budget estimation once per year". Since these two design decisions belong to the same EA design artifact they are interrelated with a *decomposition* relationship.

Table 1. Design rationale summarization table

D01	Create budget forecast business process
IS01	Storing budget estimation frequency
D02	Storing budget estimation once per year
IS02	Find solution for storing and processing budget
D03	COTS Application A
D04	Upgrade application A
IS03	How to upload budget data
D05	Create budget spreadsheet
OI1	Each department created its own excel form, resulting in incompatible information
IS04	How to upload budget data
D06	Build budget input interface
OI2	Errors in the calculation of the budget forecast. The application does not detect mistakes
IS05	Extend the application with business logic rules

The new design decision (D02) created the EA issue (IS02) "Find solution for storing and processing budget". This means that stakeholders should find a way to store the budget information. Stakeholders decided to support this need in the application level. More specifically with D04 "Upgrade application A" they decided to upgrade the existing financial application A in order to store and process the financial information. A *translation* relationship between D02 and D04 signifies how the design issue in the business layer was addressed by a design decision in a different artifact/layer of the enterprise. The alternative was the acquisition of a COTS application which is depicted by D03 "COTS Application A" and is represented with an *alternative relationship* from IS02 to D03. This signifies that D03 is a rejected design decision.

So, what was the reason that stakeholders chose the upgrade of the existing financial application? By interviewing the stakeholders we understood the context which influenced their decision making: during the execution of the enterprise transformation another high level decision from the Luxembourgish government had to be applied in the organization. The government decided that LuxRTO had to be merged with another national Research and Technology Organization. This implied the need for serious changes in the organizational structure since some departments of LuxRTO had overlapping roles with departments of the other organization. Moreover new business models should be defined based on the exchange of research expertise of research groups.

The upcoming merge of the organization posed some serious design challenges on the involved stakeholders of the budget estimation business process. On the one hand they had ambitious design goals considering the realization of this business need, while on the other hand they had to compromise because of the merge. It was not clear how the financial departments and business processes would be merged, therefore the risk of wasting budget for significant business and IT development was high.

Consequently, despite the fact that the initial plan of the stakeholders was the acquisition of a new COTS application, budget restrictions led stakeholders to the decision of upgrading the in house applications. We captured and depicted the budget restriction by using criterion C01 "cost" on our decision graph. Furthermore, by explaining to stakeholders the different types of decision making strategies we identified that they used a "lexicographic strategy", which means that they rejected alternatives by just using the "cost" as the most important criterion without examining other quality characteristics. The rationale behind this strategy was, as we mentioned before, the "upcoming merge of LuxRTO".

D04 created some additional issues in the application layer. The financial application was able to support the storage of the financial data but this information should somehow entered in the system (IS03). Stakeholders, by having in mind again the budget restiction, decided to use a spreadsheet standardized template (D05). This template was distributed by the financial department to different departments of LuxRTO. The users of the other departments had to fill the spreadsheet template and send it back to the financial department for further processing. This flow of EA design decisions and issues comprises the underlying rationale of the EA model of Fig. 2.

However, several unanticipated consequences occurred after the execution of these decisions. The use of spreadsheet templates for the insertion of budget data was problematic. More specifically, the users of each department started modifying the template and the order of the data fields. The financial officer who was receiving the input budget data had serious problems on the processing of this information and in turn on the calculation of the budget forecast. The usability of the budget forecast business process was deteriorated. The observed impact (OI1) "Each department created its own excel form, resulting in incompatible information" captures and represents this problem.

In order to solve the problem the stakeholders decided to upgrade further the financial application A with a budget input application interface (D06). EA Decision D06 "Build budget input interface" solves the unanticipated consequences of D05. This is also represented by a *substitution* relationship between D06 and D05. The resulting EA model after these modifications is depicted in Fig. 2. Despite the fact that this EA model represent the final outcome of this enterprise transformation, other EA issues were still open. After the incorporation of the budget input application interface another problem arose in the budget forecast business process and it was not addressed. This application module lacks business logic error checking functionality during the data entry of the budget input. The problem is depicted in OI2 "Errors in the calculation of the budget forecast. The application does not detect mistakes" and users of this application who are not familiar with financial parameters can create serious mistakes on the calculation of budget forecast. A new EA issue was created (IS05) "Extend the application with business logic rules". Despite the fact that stakeholders were aware of the problem, they were not able to take additional decisions because of the upcoming merging. The EA issue remained unresolved.

Without rationalization the above reasons behind the architecture designs of Figures 2, 3 remain implicit. Yet clearly such rationalization is useful. For example: by using rationalization one explicates the negative observed impact of diverging spreadsheets as a result of the introduction of the new business process. As a result this negative observed impact can be anticipated on for future similar decisions.

4 Lessons Learned

This section presents the lessons learned of applying EA Anamnesis to a real life case study.

Lesson 1: EA Anamnesis can reflect the decisions made by the budget forecasting practitioners. As was stated in Sect. 3, the main objective of this study was the evaluation of our approach in terms of its ability to capture and represent design rationales of Enterprise Architecture designs.

As stated in Sect. 3, the design rationale was created together with the involved stakeholders. Their perception was that the approach was adequate in terms of expressivity of reasoning and decision relationships. They were able to trace their design decisions and to realize what were the cross cutting implications of their decisions. For example: using our approach, the stakeholders could express that the IT-application layer decision to create a budget spreadsheet has the business process layer impact of having different, and incompatible, spreadsheets from each department.

Lesson 2: Stakeholders use simple selection processes, or decide without examining alternatives. This reduces overall capturing effort Our approach is designed to cover a variety of decision making strategies, compensatory or non compensatory. Our findings, at least for this case, show that actually designers use simple techniques to eliminate alternatives from their choice set. For example, in Sect. 3 we have seen that "cost" is the only criterion for the decision "Upgrade Application A". Even more, sometimes stakeholders solved an EA issue without examining alternative choices. The main reason for not considering alternatives is that experienced stakeholders make decisions based on previous experiences from similar cases.

Advanced techniques like multiple-criteria decision analysis (MCDA) were not used for any of the captured design decisions.

We argue that this finding actually supports the applicability of our approach in practice since it is easier in terms of capturing effort for the designer to capture the underlying decision making strategies.

Lesson 3: By modeling decisions in EA Anamnesis, stakeholders became aware of decision making strategies. We had to educate the stakeholders and make them understand how they actually decide. Implicitly, they were using decision making processes. However, the stakeholders did so *without being aware of this.* The awareness of different types of decision making strategies enabled them to better structure and analyze the decision problem. This means that they were able to explicitly describe how they decided (decision making strategy) for a certain decision problem and what evaluation criteria they used.

Lesson 4: EA Anamnesis insufficiently reflects that decision making can be ongoing, with open issues. As can be observed from EA issue 05 "Extend the application with business logic rules" (Fig. 4) some EA issues were not resolved. Reasons such as lack of resources (budget, time) sometimes prevent designers from addressing open issues.

This is currently not reflected in EA Anamnesis, which assumes that decision making is a past rather than an ongoing activity. We feel that, in a future iteration, ongoing issues should be captured explicitly by our approach. This is because awareness of unresolved issues gives the ability of better justification of EA designs. For example, by capturing open issues a stakeholder of the RTO organization can justify a lacking usability of the budget forecast business process due to a lack of business logic control mechanisms in the application layer.

5 Conclusions and Future Work

In this paper we presented the application of EA Anamnesis approach on a real world enterprise architecture transformation. By conducting case study research we testified the capability of our approach to capture and represent adequately design rationales. The approach captures sufficiently design rationales for EA. Furthermore, during the application of our approach, some important lessons derived from this case. The decision making strategies used by the stockholders of this case were much more simpler than initially perceived. This can reduce further the capturing effort of our approach and in turn improve its usability in practice.

For future research, we intend to confront decision models of our approach to enterprise architecture practitioners. An example of such an evaluation is to divide participating practitioners in two groups, whereby one group receives an architectural design and the other group receives an architectural design and an EA Anamnesis rationalization thereof. Subsequently, we could ask both groups the same questions about the architectural design, and observe to what extent and how EA Anamnesis aids the architects on the understanding of the EA design.

Acknowledgments. The authors would like to thank Nathalie Bonnel and Frédéric Klein for their valuable contributions to this case study.

This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg* (`www.fnr.lu`), via the PEARL programme.

References

1. Lankhorst, M.: Enterprise architecture at work: Modelling, communication and analysis. Springer (2009)

2. The Open Group: ArchiMate 2.0 Specification. Van Haren Publishing (2012)
3. OMG: Unified profile for DoDAF and MoDAF (UPDM), version 2.1 (2013)
4. van der Linden, D., Hoppenbrouwers, S., Lartseva, A., Molnar, W.: Beyond terminologies: Using psychometrics to validate shared ontologies. Applied Ontology 7, 471–487 (2012)
5. Jansen, A., Bosch, J.: Software architecture as a set of architectural design decisions. In: 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005, pp. 109–120. IEEE (2005)
6. Tang, A., Jin, Y., Han, J.: A rationale-based architecture model for design traceability and reasoning. Journal of Systems and Software 80, 918–934 (2007)
7. Plataniotis, G., de Kinderen, S., van der Linden, D., Greefhorst, D., Proper, H.A.: An empirical evaluation of design decision concepts in enterprise architecture. In: Grabis, J., Kirikova, M., Zdravkovic, J., Stirna, J. (eds.) PoEM 2013. LNBIP, vol. 165, pp. 24–38. Springer, Heidelberg (2013)
8. Plataniotis, G., de 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 2013 and EMMSAD 2013. LNBIP, vol. 147, pp. 339–353. Springer, Heidelberg (2013)
9. Plataniotis, G., der Kinderen, S., Proper, H.A.: Relating decisions in enterprise architecture using decision design graphs. In: Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC)(2013)
10. Plataniotis, G., de Kinderen, S., Proper, H.A.: Ea anamnesis: An approach for decision making analysis in enterprise architecture. In: International Journal of Information System Modeling and Design (IJISMD) (2014)
11. Plataniotis, G., de Kinderen, S., Proper, H.A.: Ea anamnesis: towards an approach for enterprise architecture rationalization. In: Proceedings of the 2012 workshop on Domain-specific modeling, DSM 2012, pp. 27–32. ACM, New York (2012)
12. Eilon, S.: What is a decision? Management Science 16, B–172-B–189 (1969)
13. Harmsen, F., Proper, E., Schalkwijk, F., Barjis, J., Overbeek, S. (eds.): PRET 2010. LNBIP, vol. 69. Springer, Heidelberg (2010)
14. Tyree, J., Akerman, A.: Architecture decisions: Demystifying architecture. IEEE Software 22, 19–27 (2005)
15. Mintzberg, H., Raisinghani, D., Theoret, A.: The structure of unstructured decision processes. Administrative Science Quarterly, 246–275 (1976)
16. Kruchten, P.: An ontology of architectural design decisions in software intensive systems. In: 2nd Groningen Workshop on Software Variability, pp. 54–61 (2004)
17. Kruchten, P., Lago, P., van Vliet, H.: Building up and reasoning about architectural knowledge. In: Hofmeister, C., Crnković, I., Reussner, R. (eds.) QoSA 2006. LNCS, vol. 4214, pp. 43–58. Springer, Heidelberg (2006)
18. Op't Land, M., Proper, H.A.: Impact of principles on enterprise engineering. In: ECIS 2007 Proceedings (2007)
19. Runeson, P., Host, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14, 131–164 (2009)