2015 IEEE 17th Conference on Business Informatics

A Conceptual Model for Compliance Checking Support of Enterprise Architecture Decisions

Georgios Plataniotis[*†‡], **Sybren de Kinderen**[*‡], **Qin Ma**[*‡], **Erik Proper**[*†‡]
[*] Luxembourg Institute of Science and Technology, Luxembourg
[†] Radboud University Nijmegen,Nijmegen, the Netherlands
[‡] EE-Team, Luxembourg, Luxembourg
{georgios.plataniotis, sybren.dekinderen, qin.ma, erik.proper}@list.lu

***Abstract*—Enterprise architecture modeling languages capture holistically the structure of an enterprise. They therefore represent how the services and business processes of an organization are supported by IT infrastructure and applications. However, the reasoning behind the selection of specific design decisions in the architecture remains usually implicit.**

In our earlier work we proposed the EA Anamnesis approach which captures design rationalization information in the solution space of the enterprise architecture. Its major contribution is a formal metamodel that captures the reasoning behind design decisions and the relationships between them. In this paper, we extend our approach with concepts from the problem space domain of the enterprise architecture, such as goals, principles, requirements. Furthermore, we provide a bridging with the existing concepts of EA Anamnesis which are part of the solution space. In doing so, we can represent the extent to which EA design decisions, which define the EA design, comply with given goals, principles and requirements. The extension is evaluated with a real world case study within a Research and Technology Organization.

***Keywords*—Enterprise Architecture, Design Rationale, Goals, Principles, Functional Requirements, Nonfunctional Requirements, Decision making process, Problem Space, Solution space**

I. Introduction

Enterprise Architecture (EA) models are considered as an instrument to represent an enterprise holistically [1], linking perspectives on an organization which are usually considered in isolation. In doing so, one can consider the organization-wide impact of a change [1], [2], expressing its complete business-to-IT-stack [3]. For example, for a newly introduced IT application, EA models can be used to consider implications on business processes, human resources, organizational goals, and more.

Although EA models can be used to capture the holistic design of an organization, they rarely specify the design decisions behind the resulting models. Even if we should be careful with the analogy, experience from the field of software architecture shows that leaving design rationales implicit leads to 'Architectural Knowledge vaporization' (cf. [4]). This means that without design rationale, design criteria, reasons and alternatives are left implicit.

As a result of this lack of rationalization, architects are unable to justify their designs [5]. Furthermore, new designs are constructed in an ad hoc manner without taking into consideration constraints imposed by past design decisions [5]. Here, constraints refer to boundaries implied by the design. These boundaries can be of business or technical nature, such as the choice for a programming language that comes with choosing a specific appilication development environment.

Moreover, a survey amongst enterprise architecture practitioners [6] provides indications of the usefulness of design rationalization for motivating design decisions and for architectural maintenance. At the same time, however, the survey shows that practitioners often forego the use of a structured template/approach when rationalizing an architecture, relying instead on ad hoc information capturing on tools such as Microsoft Office.

In our earlier work [7], [8], [9], 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 relies on a metamodel-based formal linkage between EA modeling languages (mainly ArchiMate) and the corresponding decision aspects to realize the connection between EA designs and EA rationale.

So far we have evaluated EA Anamnesis approach in the context of a real world enterprise architecture transformation. The evaluation showed that the approach was capable to sufficiently capture design rationales and enabled practitioners to structure their decision making problem. However, the evaluation also indicated that EA Anamnesis should provide traceability from the problem space down to the solution space. Currently, EA Anamnesis captures how the enterprise architect solved the given architectural problem but the role of critical concepts in enterprise architecture like goals, principles and requirements is not explicitly taken into consideration.

As a response, in this paper we provide a bridging between the concepts that belong to the problem space of enterprise architecture (goals, principles, requirements) and the existing concepts of EA Anamnesis that belong to the solution space of enterprise architecture. By doing so, we provide additional dimensions of design rationalization since we are able to identify the role of the architectural principles and requirements on the decision making process. We check the compliance of the EA decisions and design based on the given principles and requirements, we analyze how the principles are specialized in more refined functional and nonfunctional requirements, and

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

978-1-4673-7340-1/15 $31.00 © 2015 IEEE
DOI 10.1109/CBI.2015.46

we justify the elicitation of specific principles and requirements based on the given high level goals.

This paper is structured as follows. Section II introduces the Research and Technology Organization case study and discusses the case study protocol that we followed and the limitations. Section III introduces the enhanced conceptual model. For pedagogical reasons the problem space concepts of the metamodel are illustrated through the real world case study. In Section IV we evaluate our approach by capturing design rationales of the enterprise transformation of the research and technology organization. Section V presents related work from the domains of requirements engineering and design rationale and finally Section VI concludes.

II. Case study: Budget forecasting at LuxRTO

In this section we introduce a real world case study in a Luxembourgish Research and Technology organization, hereafter referred to as LuxRTO. Subsequently, we use it to illustrate our extended conceptual model in the next section.

A. Case study setup

1) 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 the 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 beginning of the transformation project, she had knowledge about the driving motives that initiated this transformation and how the business process design had evolved over time. On the other hand, the IT architect had significant IT expertise on the transformation project. Additionally, 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 the 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 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 [10]. For example: prior to the collaborative modeling we explained our approach to practitioners. This is in line with [10], that advices to prepare for data collection prior to the collection of evidence.

2) 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 almost 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 this information. 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 domain). We are aware of this restriction but in the current stage of our research we focus on how our approach captures design rationales and not on the support of multiple stakeholders decision making.

B. Budget forecasting at a Research and Technology Organization

During the last years, the Luxembourgish government introduced stricter rules on the budget spending of research institutions. This policy had to be incorporated by research institutions, meaning that the institutions should be able to establish long term financial projection plans. The idea is that these plans provide institutions with a better insight regarding the availability of resources, and in turn foster 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, a comparison between the forecasted and valuable budget, and in general robustness of the organization's financial data.

C. Enterprise Transformation

We now describe how the enterprise design was changed 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. Note 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.

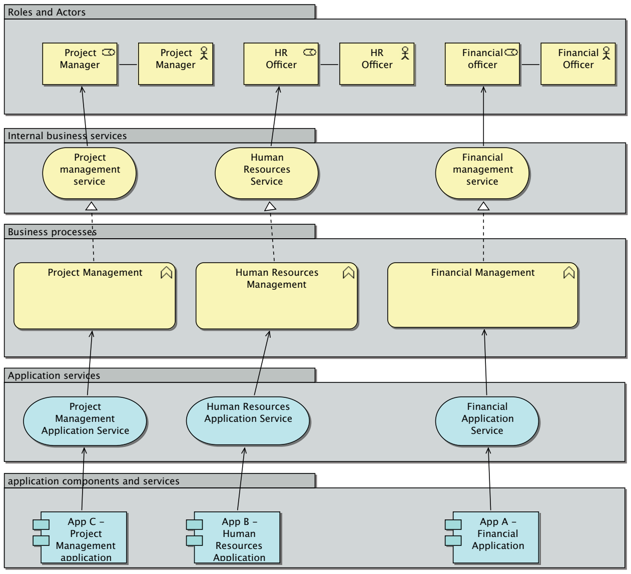
Roles and Actors
Project Manager
Project Manager
HR Officer
HR Officer
Financial officer
Financial Officer
Internal business services
Project management service
Human Resources Service
Financial management service
Business processes
Project Management
Human Resources Management
Financial Management
Application services
Project Management Application Service
Human Resources Application Service
Financial Application Service
application components and services
App C – Project Management application
App B – Human Resources Application
App A – Financial Application

Fig. 1. LuxRTO enterprise transformation - As is architecture

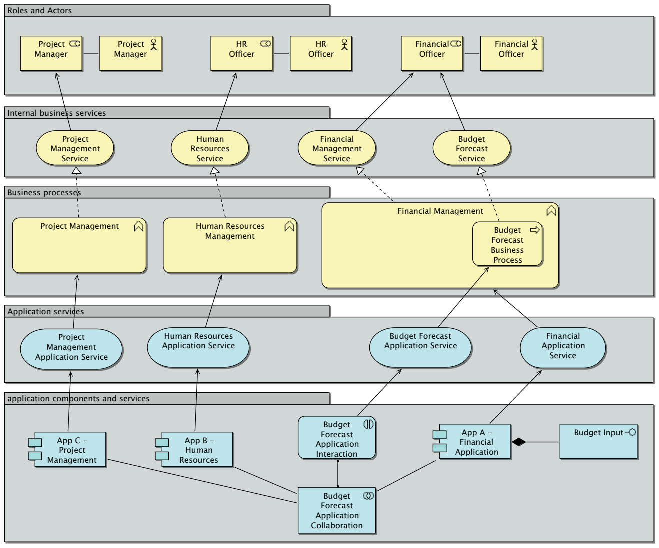
Roles and Actors
Project Manager
Project Manager
HR Officer
HR Officer
Financial Officer
Financial Officer
Internal business services
Project Management Service
Human Resources Service
Financial Management Service
Budget Forecast Service
Business processes
Project Management
Human Resources Management
Financial Management
Budget Forecast Business Process
Application services
Project Management Application Service
Human Resources Application Service
Budget Forecast Application Service
Financial Application Service
application components and services
App C – Project Management
App B – Human Resources
Budget Forecast Application Interaction
App A – Financial Application
Budget Input
Budget Forecast Application Collaboration

Fig. 2. LuxRTO enterprise transformation after the incorporation of the new business process

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

Application B is the human resources management application. It supports tasks like resource allocation, start/end dates of work contracts, a weekly calendar, and different types of leaves (sickness, vacation etcetera).

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

1) The as-is architecture.: Fig. 1 presents the enterprise architecture before the incorporation of the new budget forecast business process. The RTO had already implemented business processes that supported business stakeholders with the management of projects, finance and human resources. These business processes were supported by the corresponding information systems.

Fig. 2 depicts the EA model after the incorporation of the budget forecast business process. From this model we understand that the business process was supported by the interaction and collaboration among Applications A, B, C and a new application interface.

III. The EA anamnesis metamodel

For comprehension purposes the concepts of the metamodel are discussed according to its two viewpoints: problem space (Subsect. III-B) and solution space (Subsect. III-C). Figure 3 presents the enhanced EA Anamnesis metamodel.

A. Two viewpoints for EA Anamnesis

We provide a distinction between the problem space and the solution space in line with requirement engineering literature [11].

The problem oriented view in requirements engineering describes and analyzes the experienced problems and how they were interrelated. In this kind of approaches high level goals are refined to smaller problems that are more concrete and measurable. Subsequently, those refined problems can be addressed by specific decisions.

On the other hand, the solution oriented approaches provide the justification behind the selection of specific solutions. This justification should be based on specify quality criteria.

The idea behind the bridging between problem and solution space is that the solutions and the quality criteria which rationalize the solutions should be motivated and aligned with the requirements from the problem space. By doing so we maximize the compliance of our solution to the given requirements.

B. The Problem Space

1) Problem Space concepts:

Goal: In accordance with goal-oriented modeling approaches [12] a goal is defined as an end state that a stakeholder intends to achieve.

Goals can be classified as business, enterprise architecture or IT goals [13].

Example: The business goal of LuxRTO for this specific enterprise architecture transformation is 'Establish long term financial projection plans'.

Architecture principle: An architecture principle is a general rule and guideline that supports the realization of

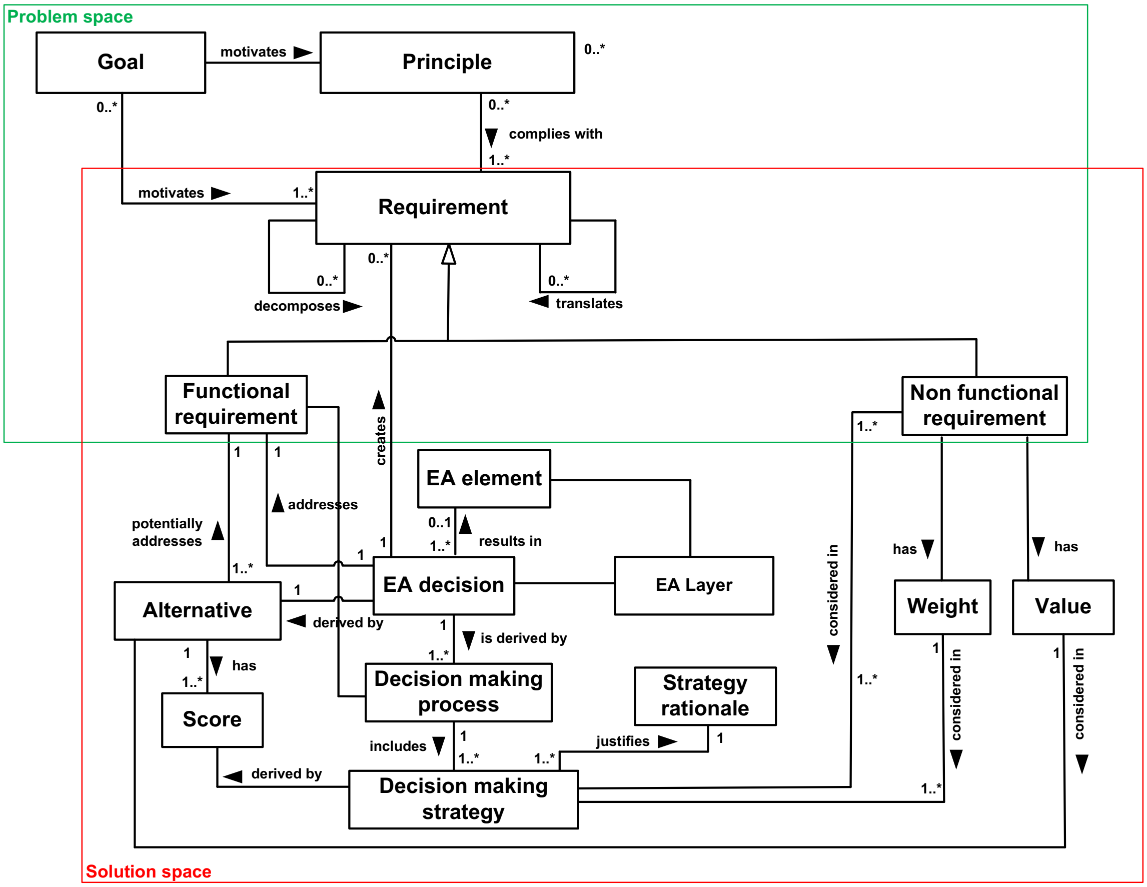

Fig. 3. Enhanced EA Anamnesis metamodel

the goals of the enterprise [14], [15]. Principles are usually abstract, high level propositions and should specialized in more refined entities which will guide the development and evolution of the enterprise. Principles are strongly related with goals and requirements. On the one hand, goals justify the selection of specific principles and on the other hand requirements should be compliant with the given principles [13].

Example: The principle 'enterprise architecture is built on reusable components'.

Requirement: A requirement is defined as a statement of need, condition or capability that should be realized by a system [11], [16]. They are usually derived from goals and they should be compliant with the given principles [16], [13].

This can range from a high, domain independent, level of abstraction to detailed, domain-dependent descriptions of system functions[11]. This means that they should be interpreted by the architect in different ways depending on the domain. For example, a generic requirement for security must be realized in different ways across the different domains of the enterprise. The enterprise architect should translate this generic need into specialized requirements per domain.

Requirement types:

a) Functional requirements:

Functional requirements specify **what** the system should do or in other words a specific behavior that a system must have [11].

Example: 'Find a solution for solution for storing and processing budget'.

b) Nonfunctional requirements:

Nonfunctional requirements specify the behavioral aspects of a system or in other words the quality criteria that determine **how** the system works.

Example: Application interoperability

Moreover, a special type of nonfunctional requirement is the constraint. Constraints restricts further the solution space of what is adequate for addressing a specific functional requirement [17]. A constraint does not prescribe a specific functionality on the system, but introduces restrictions on how the functionality will be realized. Examples of constraints can be financial or time issues or a specific technology that should be used.

Problem Space relations:

In this part we describe the different relationship types which interrelate the problem space concepts in our metamodel.

Motivates: In enterprise architecture goals should be translated into refined requirements [16], [13]. This is done in two ways, with or without the intervention of architectural principles [18].

• The direct relationship between goals and requirements is signified by the relationship 'motivates' between the concepts 'Goal' and 'Requirement'.

• Goals motivate principles. In our metamodel this is signified by the relationship 'motivates' between the concepts 'Goal' and 'Principle'. With these two relationships we support traceability from specific requirements and architectural principles to goals.

Complies with The relationship 'Complies with' provides a refinement of principles in both types of requirements: "functional" and "non functional". This is in accordance to [15], [13], which specifies that principles should inform requirements.

1) Decomposition:

As we can see in Figure 3 the requirement can be decomposed further until we arrive to a desired abstraction level for the domain architecture. For our approach this means that we can decompose the problem until we arrive to requirements that can be addressed by concrete EA design decisions. The decomposition relationship is in line with 'decomposes into' relationship of Kruchten's ontology [19].

2) Translation:

Translation relationship describes how requirements that belong to different EA layers of the enterprise are translated in requirements on a different layer [8]. This relationship is critical for the domain of enterprise architecture since it provides to the enterprise architect a holistic view of the cross layer dependencies of the requirements.

C. The Solution Space

Alternative: The concept of alternative represents the available choices that are under evaluation by using a specific decision making strategy [20].

EA decision: The concept of EA decision represents the final choice among the alternatives. Since the information for the alternatives already exist, the concept 'EA decision' is automatically derived from the alternative with the highest utility score.

Decision making process: A decision making process provides the summarization of the decision making for a specific design decision. As we explain below, one or more decision making strategies can be used for the making of a single decision. In our metamodel the decision making process is automatically derived by the set of the decision making strategies and it is linked to the concept of EA decision.

Decision-Making Strategy: This concept captures the decision making strategy used by the enterprise architect to evaluate the alternatives, and make the actual EA decision. Decision making strategies are characterized as compensatory, noncompensatory, or as a hybrid of these two. For a detailed description of the different types of decision making strategies in EA Anamnesis please refer to [9].

Strategy rationale: In the context of 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. Typical reasons for the adoption of different strategies by the architect are constraints that come from different domains of the enterprise. The capturing of this information justifies the decision strategy that was selected for the evaluation process. This is what is referred to as metadecision making, decision making about the decision process itself [21]. As stated in the metamodel, one strategy rationale can justify one ore more decision making strategies.

Value and weight: The value concept represents the value that the decision maker assigns to the nonfunctional requirement during the evaluation process and weight concept represents the importance of the non functional requirement. By capturing the value and weight of nonfunctional requirements, stakeholders that analyze in depth the architecture, can understand which criteria had a determinant role in the selection process and on which strategy they were based.

IV. Evaluation

In Section II we introduced the real world case study and the changes that were made in the enterprise architecture design. However, the rationale behind this design is not captured by the EA models.

The following questions are not easy to be answered by just examining an EA model. How do the EA design decisions comply with the given goal, principles and requirements? What was the role of the problem space concepts during the decision making process for the selection of the best alternatives?

By answering these questions we do not only provide traceability on the solution space of the architecture, but we also provide the bridging between the problem and the solution space. This is done by making explicit the role of the requirements and principles on the decision making process.

We use two different ways for representation of the captured information, visualization and summarization tables. The visualization of Figures 4,5 were constructed during the capturing of design rationales and Table I provides the summary of the decision making process for a specific design decision.

1) Compliance of EA decisions to Goals and Functional Requirements: As we mentioned before our approach provides a bridging between the problem and solution space of the enterprise architecture. In this part we capture and visualize in Fig. 4 how the business goal was analyzed into concrete functional requirements and how these requirements were addressed by specific design decisions. In this way we can trace from elements of the solution space back to the motivational aspects and vice versa.

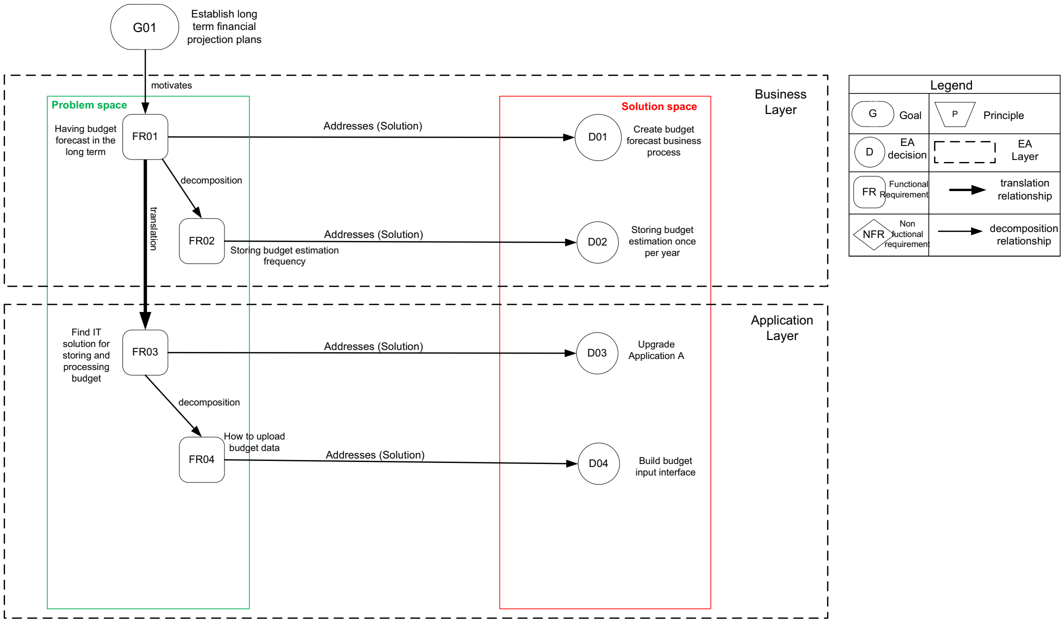
G01
Establish long term financial projection plans
motivates
Business Layer
Problem space
Having budget forecast in the long term
FR01
Addresses (Solution)
Solution space
D01
Create budget forecast business process
decomposition
translation
FR02
Addresses (Solution)
Storing budget estimation frequency
D02
Storing budget estimation once per year
Application Layer
Find IT solution for storing and processing budget
FR03
Addresses (Solution)
D03
Upgrade Application A
decomposition
How to upload budget data
FR04
Addresses (Solution)
D04
Build budget input interface
Legend
G
Goal
P
Principle
D
EA decision
EA Layer
FR
Functional Requirement
translation relationship
NFR
Non fuctional requirement
decomposition relationship

Fig. 4. Compliance of EA decisions to Goals and Functional Requirements

We started our evaluation by capturing the business goal G01 ‘Establish long term financial projection plans’, which initiated the enterprise transformation. Motivated by the business goal, the stakeholders defined the functional requirements and subsequently they made the appropriate decisions which would realize the goal. We captured functional requirements FR01-FR04 which describe the given architectural problem (problem space) and EA design decisions D01-D04 which describe how the stakeholders addressed the architectural problems (solution space). Moreover, the relationships translation and decomposition capture how from a given requirement we can go into a more refined one or to a requirement of another layer (cross-layer relationship).

Here we see how G01 motivates the functional requirement FR01 ‘Having budget forecast in the long term’ on the business layer of LuxRTO. Business stakeholder decided to establish the business process ‘Create budget forecast business process’ which is captured in D01. Moreover the functional requirement FR01 for the new business process is further decomposed (decomposition relationship) to FR02 ‘Storing budget estimation frequency’. Business stakeholders decided to provide estimation for the budget spending once per year. We captured their decision in D02.

After the definition of the new business process and its specificities, stakeholders had to identify an appropriate IT solution that would support the storing and processing of the budget information. We captured FR03 ‘Find IT solution for storing and processing budget’. FR01 and FR03 are related with a translation relationship in order to make explicit the cross-layer dependencies of requirements in enterprise architecture. Stakeholders of the IT domain after examining the alternative IT solutions (see Subsect. II-C) decided to upgrade the existing application A (D03). Furthermore, based on FR04 ‘How to upload budget data’, they decided to ‘Build budget input interface’ in order to upload data (D04).

2) Checking the Compliance of EA decision making with nonfunctional requirements and principles: In this part we elaborate further in a specific EA design decision of our case study and we show how our approach makes explicit the role of principles and nonfunctional requirements on the decision making process and subsequently how compliant our design decisions are with those factors.

Fig. 5 provides the visualization of the decision making process for EA decision ‘Upgrade Application A’ (D03). As we discussed above, D03 addresses FR03 ‘Find IT solution for storing and processing budget’. An alternative that potentially was addressing FR03 was AL01 ‘COTS Application’.

So, what was the reason that stakeholders chose the upgrade of the existing financial application (D03) instead of AL01 ‘COTS Application’? By interviewing the stakeholders we understood and captured 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 imposed 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. Their initial plan was to acquire ‘COTS Application’ that would be possible to also support the future needs of the organization. This plan was in accordance with the architecture principle 01 of LuxRTO

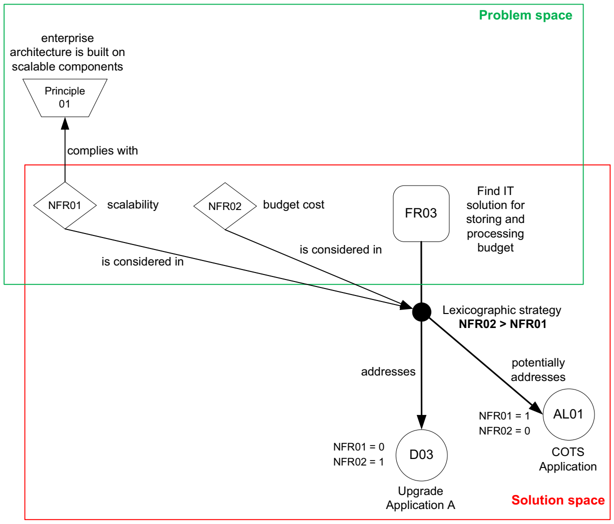

Fig. 5. Checking the Compliance of EA decision making with nonfunctional requirements, principles and constraints

'enterprise architecture is built on scalable components'. At the same time 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 'COTS Application', the upcoming merge led to minimization of budget spending which led stakeholders to the decision of upgrading the in house applications. By using our approach we captured the aforementioned situation: Two nonfunctional requirements were considered in the decision making process for the application selection, NFR01 'scalability' which complied with Principle 01 and NFR02 'budget cost'. On the one hand, 'COTS Application' was satisfying NFR01 but not NFR02 and on the other 'Upgrade application A' was not satisfying NFR01, but was satisfying NFR02. In the specific decision making context stakeholders decided that they should prioritize and decide based on NFR02 'budget cost', therefore they rejected AL01 'COTS Application'. The decision making strategy for this kind of prioritization is a 'lexicographic' strategy.

Without rationalization the above reasons behind the architecture designs of Figures 1, 2 remain implicit. Stakeholders or even newcomers to the enterprise that want to analyze or provide justification of past decisions to management or other stakeholders, can use EA Anamnesis approach in order to understand the role of principles and requirements to the decision making process. For example, if a newcomer is asked about the alignment of the 'Upgrade application A' with the principle 'enterprise architecture is built on scalable components', he will be able to justify that this application is not aligned with the given principle since his predecessors had to compromise with a low budget solution.

V. Related Work

The motivation extension of the ArchiMate EA modeling language models reasons behind EA designs. It offers the possibility to explicitly link architecture designs to problem space concepts like requirement, goal, driver, assessment, or constraint. Thereby, the motivation extension provides a bridging between EA designs and corresponding problem space concepts. However, according to [22] the large amount of concepts in combination with their sometimes ambiguous definition, introduces difficulties in the usability of the motivational extension by enterprise architecture practitioners. The authors identify that three concepts are well understood: stakeholder, requirement and goal. The remainder is considered ambiguous. Interestingly, the large amount of the ambiguously defined concepts also hints that the motivational extension is at odds with one of the key design principles behind the ArchiMate language: 'the language should be as compact is possible' [23]. Differently, EA Anamnesis uses a limited set of problem space concepts, i.e., no more than strictly necessary to rationalize an architectural design.

Furthermore, while the ArchiMate motivation extension can be used to link EA designs to motivational concepts, it cannot express the underlying decision making process. As such, it leaves implicit prioritization and conflicts amongst requirements.

Moreover, Tang et al [24] proposed a metamodel that provides a bridging between the problem space and solution space for the domain of software architecture. However, this metamodel is tailored to the software architecture. As a result it lacks typical enterprise architecture concepts, such as 'Principle'.

TABLE I. EA decision 03 summary

Title:	Upgrade Application A
Functional requirement:	Find IT solution for storing and processing budget
Layer:	Application
Alternatives:	COTS application A
Rationale:	Minimize budget cost prioritized over scalability
Decision making strategy:	Lexicographic

VI. Conclusions and future work

In this paper we presented an enhancement of the EA Anamnesis approach with problem space concepts. This enhanced conceptual model provides a bridging between the problem and solution space of enterprise architecture and makes explicit the role of problem space concepts on the decision making processes. We evaluated through a real world enterprise architecture transformation the ability of our approach to check the compliance of EA design decisions with the given problem space concepts.

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 into 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] M. Lankhorst, *Enterprise architecture at work: Modelling, communication and analysis*. Springer, 2009.

[2] M. Op't Land, E. Proper, M. Waage, J. Cloo, and C. Steghuis, *Enterprise architecture: creating value by informed governance*. Springer, 2008.

[3] R. Winter and R. Fischer, "Essential layers, artifacts, and dependencies of enterprise architecture," *Journal of Enterprise Architecture–May*, pp. 1–12, 2007.

[4] A. Jansen and J. Bosch, "Software architecture as a set of architectural design decisions," in *Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on*. IEEE, 2005, pp. 109–120.

[5] A. Tang, Y. Jin, and J. Han, "A rationale-based architecture model for design traceability and reasoning," *Journal of Systems and Software*, vol. 80, no. 6, pp. 918 – 934, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121206002287

[6] G. Plataniotis, S. de Kinderen, D. van der Linden, D. Greefhorst, and H. A. Proper, "An empirical evaluation of design decision concepts in enterprise architecture," in *Proceedings of the 6th IFIP WG 8.1 working conference on the Practice of Enterprise Modeling (PoEM 2013)*, 2013.

[7] G. Plataniotis, S. de Kinderen, and H. A. Proper, "Ea anamnesis: An approach for decision making analysis in enterprise architecture," *International Journal of Information System Modeling and Design (IJISMD)*, 2014.

[8] G. Plataniotis, S. d. Kinderen, and H. A. Proper, "Relating decisions in enterprise architecture using decision design graphs," in *Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC)*, 2013.

[9] G. Plataniotis, S. de Kinderen, and H. A. Proper, "Capturing decision making strategies in enterprise architecture – a viewpoint," in *Enterprise, Business-Process and Information Systems Modeling*, ser. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2013, vol. 147, pp. 339–353. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38484-4_24

[10] P. Runeson and M. Host, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10664-008-9102-8

[11] K. Pohl, *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.

[12] A. Van Lamsweerde, "Goal-oriented requirements engineering: A guided tour," in *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*. IEEE, 2001, pp. 249–262.

[13] D. Stelzer, "Enterprise architecture principles: literature review and research directions," in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*. Springer, 2010, pp. 12–21.

[14] D. Greefhorst and E. Proper, *Architecture Principles: The Cornerstones of Enterprise Architecture*. Springer Science & Business Media, 2011, vol. 4.

[15] V. Haren, *TOGAF Version 9.1*. Van Haren Publishing, 2011.

[16] The Open Group, *ArchiMate 2.0 Specification*. Van Haren Publishing, 2012.

[17] M. Glinz, "On non-functional requirements," in *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*. IEEE, 2007, pp. 21–26.

[18] D. Quartel, W. Engelsman, H. Jonkers, and M. Van Sinderen, "A goal-oriented requirements modelling language for enterprise architecture," in *Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International*. IEEE, 2009, pp. 3–13.

[19] P. Kruchten, P. Lago, and H. Vliet, "Building up and reasoning about architectural knowledge," in *Quality of Software Architectures*, ser. Lecture Notes in Computer Science, C. Hofmeister, I. Crnkovic, and R. Reussner, Eds. Springer Berlin Heidelberg, 2006, vol. 4214, pp. 43–58.

[20] J. Orasanu and T. Connolly, "The reinvention of decision making." *Decision making in action: Models and methods*, pp. 3–20, 1993.

[21] H. Mintzberg, D. Raisinghani, and A. Theoret, "The structure of unstructured decision processes," *Administrative science quarterly*, pp. 246–275, 1976.

[22] W. Engelsman and R. Wieringa, "Understandability of goal-oriented requirements engineering concepts for enterprise architects," in *Advanced Information Systems Engineering*. Springer, 2014, pp. 105–119.

[23] M. M. Lankhorst, H. A. Proper, and H. Jonkers, "The anatomy of the archimate language," *International Journal of Information System Modeling and Design (IJISMD)*, vol. 1, no. 1, pp. 1–32, 2010.

[24] A. Tang, P. Liang, V. Clerc, and H. van Vliet, "Supporting co-evolving architectural requirements and design through traceability and reasoning," *Relating Software Requiremens and Software Architecture*, 2011.