Quantitative Alignment of Enterprise Architectures with the Business Model

Wilco Engelsman[1,3], Jaap Gordijn[2,4], Timber Haaker[1], Marten van Sinderen[3], and Roel Wieringa[2,3]

[1] Saxion, University of Applied Sciences
[2] The Value Engineers
[3] University of Twente
[4] Vrije Universiteit Amsterdam

Abstract. For many companies, information and communication technology (ICT) is an essential part of the value proposition. Netflix and Spotify would not have been possible without internet technology.
Business model upscaling often requires a different ICT architecture, because an up-scaled business model imposes different performance requirements. This new architecture needs investments and has different operational expenses than the old architecture and requires recalculation of the business model. Investment decisions, in turn are guided by performance requirements.
There are currently no methods to align a quantified business value model of a company with performance requirements on the enterprise architecture. In this paper, we show how to derive performance requirements on an enterprise architecture (EA) specified in ArchiMate from a quantification of a business model specified in e^3*value*. Second, we show how we can aggregate investments and expenses from an ArchiMate model and insert these into an e^3*value* model.
We provide an initial evaluation of these quantitative alignment techniques in a real-world case study with an expert evaluation.

Keywords: e^3*value*, ArchiMate, Traceability, Business Value Model, Enterprise Architecture, Quantitative Alignment.

1 Introduction

Commercial services and physical products rely heavily on ICT. For example, Netflix and Spotify would not have been possible without the large scale deployment of content servers and networks. Physical products often have digital twins, which complement the product with additional features, allowing for simulation, training, etc. Since ICT is an intrinsic part of the value proposition of these organizations, it can no longer be considered as a cost-only factor. ICT should be part of value proposition design.

In an ecosystem products and services are exchanged between at least two, but often more enterprises. Each enterprise focuses on its core competences and jointly they satisfy a complex customer need. Following Moore [11], we define an ecosystem as a collection of companies that work cooperatively and competitively to satisfy customer needs.

To assess financial sustainability of an ecosystem, we need a *business value model* of the ecosystem (henceforth called "business model"), which we define as a conceptual model that represents the creation, distribution, and capture of value in a *network* of participants [5]. Valuable objects are services and products that satisfy customer needs, as well as payment for these. We use e^3*value* as a busines modeling language, because it allows quantification of ecosystem business models [5, 6].

A *quantified* business model of an ecosystem contains estimations of revenues and expenses of the ecosystem members. Revenues result from sales. Expenses are made to obtain e.g. raw materials, services or goods from others.

In ICT-intensive value propositions, expenses often relate to ICT components, both hard- and software. Therefore, in case of ICT-intensive services and products, the design of the provisioning *Enterprise Architecture* (EA) should be coordinated with business model design.

An EA is a high-level conceptual model of an enterprise designed to put the business strategy of an organization into operation [18]. In accordance with our networked view, EAs too should be extended to an ecosystem of enterprises [16]. We use ArchiMate [13] as the EA modeling language, where we focus on its capability to model business services and collaborations.

In previous work we created, validated and refined guidelines by which to design the business layer of an Archimate EA from an e^3*value* business model [3, 4]. In this paper we extend this with (1) guidelines to quantify workload requirements in an EA based on quantificiation of an e^3*value* model, (2) a technique by which to specify investments in and expenses on ICT in ArchiMate, and (3) a mechanism to import the specification of investments and expenses in e^3*value* models.

This paper is structured as follows. Section 2 describes related work, section 3 introduces our research methodology. In section 4 we introduce the design of our approach. In section 5 we apply our approach on a realistic example. We provide some lessons learned in section 6 and end with future work in section 7.

2 Related Work

In previous work we have created transformation guidelines between e^3*value* and ArchiMate [4]. This paper extends on this, the guidelines are required to realize the desired traceability. We build on this traceabilty to propagate economic transactions as workload requirements over the architecture and we aggregate investments and expenses of an IT architecture into an e^3*value* model.

Derzi et al. [1] realize traceability between UML deployment diagrams and e^3*value*. They annotate UML diagrams with investments and expenses and create traceability between UML and e^3*value* to be able analyze the profitability of an organization with the proposed IT. Deployment diagrams are used because they indicate ownership of ICT components, and ownership comes with an investment and operational expenses. These financials are important for the e^3*value* business model. Our work shares some similarities, we take the basic idea, but extend on this. We realize bi-directional traceability. We import economic transactions into ArchiMate for scalability reasoning in conjunction with aggregating investments and expenses from ArchiMate into e^3*value*. Our solution also has more semantics, which can be used to create tool support.

Iacob et al. [8] propose a mapping from the Business Model Canvas (BMC) [12] to ArchiMate. Since the BMC is oriented towards the *single* enterprise, this work misses the networked ecosystem point of view that is crucial to most ecosystems. We claim that exploration of the ecosystem, e.g. all participating actors and the ICT systems, need to be included in business model analysis, rather than just a single enterprise and its direct customers and supplier. Moreover, the BMC does not have the capability to quantify the business model and simulate market scenarios, as e^3*value* has, nor does the BMC have the capability to quantify ArchiMate and bring this quantification to a business model expressed in e^3*value*.

Iacob and Jonkers introduce a generic quantification approach for ArchiMate [7]. They describe a generic approach of how to perform performance analysis using workload and response times on an ArchiMate model. Our work is based on the same principles, we derive our performance requirements from e^3*value* and we quantify ArchiMate with investments and expenses. Obviously, the work of Iacob and Jonkers is restricted to ArchiMate only and therefore does not include a networked business model point of view.

Miguens [10] proposes to introduce an additional viewpoint for ArchiMate where investment information can be assigned and calculated. We do not want to perform actual investment calculations in ArchiMate beyond aggregating the information. We do all the calculations in e^3*value* because they are part of business model analysis. Miguens also does not take the business ecosystem perspective as we do, nor do they have a way to identify performance requirements based on economic transactions.

Zhou et al. [19] developed and validated a method for assigning non-functional requirements to an ArchiMate model. They refine non-functional requirements in accordance with the layers ArchiMate. Our work takes a similar approach. We use e^3*value* as a source for non-functional requirements and propagate these over the ArchiMate model with increasing granularity.

De Kinderen, Gaaloul and Proper [9] propose to link ArchiMate to e^3*value* using an intermediary language. They do not propose a direct mapping between ArchiMate and e^3*value*. They wish to introduce transactionality in ArchiMate by using the DEMO language as an intermediary language [2]. They do not use the realized traceability for quantitative alignment. We use the economic transactions from e^3*value* as workload requirements for IT systems and quantify the ArchiMate model with investments and expenses to identify the investments and expenses and insert them into the e^3*value* model for Net Present Value (NPV) calculations.

Overall, the unique element of our approach is that we take a network approach, separate business model analysis from enterprise architecture design, define links between the business model and EA to synchronize the two models. We also provide an initial evaluation of this link.

3 Design goals, research questions and methodology

Our design goal is to design techniques by which to determine if a business model is feasible in terms of financial sustainability and technological feasibility.

e^3*value* models contain a *transaction table* that identifies and counts all commercial transactions among ecosystem actors. The transaction table contains crucial information to assess long-term financial sustainability of the ecosystem because it determines revenues and expenses of each actor. Our first subgoal is now to include information from the transaction table in an ArchiMate model. Our second subgoal is to find a way to use this information to identify workload requirements on the components of an EA. Our third subgoal is to find a way to specify investments and expenses on ICT in ArchiMate that will meet these workload requirements, and export these to the corresponding e^3*value* model. This gives us the following research questions.

- Q1: How can ArchiMate represent the economic transactions of e^3*value*?
- Q2: How can performance requirements in ArchiMate be identified from the transaction table?
- Q3: How can ArchiMate be quantified with investments and expenses?
- Q4: How can expenses and investments in an ArchiMate model be fed back into an e^3*value* model?
- Q5: Do these quantitative alignment techniques provide sufficient information for investment decisions?

Q1-Q4 are design questions, Q5 is a knowledge question. The aspect of usefulness that we want to consider in Q5 is *scalability.* In other words, are these techniques useful to make decisions about scaling up a given EA? We present our answers to Q1-Q4 in section 4 by means of a toy example and provide a preliminary answer to Q5 by means of a real-world case study in section 5. This means that we follow a design science methodology [17]. In our previous work we created guidelines for designing an ArchiMate business layer model from an e^3*value* model, based on a conceptual analysis of the two languages [3]. These guidelines where then tested and refined in a lab test and subsequently in a field test [4]. The current paper is a further extension of the guidlines with quantification and a preliminary field test of this extension.

4 Design of Quantitative Alignment

Fig. 1 contains the value network and transaction table of an e^3*value* model on the left and bottom, and an ArchiMate model on the right. We will explain all parts of the figure in what follows.

e³value	Definition	ArchiMate	Definition
Actor	An entity that is economically independent.	Business Actor	Business entity capable of performing behavior.
Value Activity	Profitable task performed by an actor.	Business Service	Defined behavior that is exposed to the environment.
Value Port	Willingness to provide or request value objects.	Business Interface	Channel that exposes behavior.
Value Transfer	Willingness to transfer value objects between actors.	Flow	Transfer from one element to another.
		Serves	Provide functionality to other element.
Other *e³value* concepts			
Value Interface		Grouping of value ports .	
Value Object		An object that has economic value.	
Market Segment		A set of actors.	
Customer Need		Need to acquire something valueable.	
Boundary Element		Limit of value model.	
Other ArchiMate concepts			
Application component		Encapsulation of application functionality	
Flow		Transfer from one element to another .	
Assignment		Allocation of responsibility, performance of behavior.	
Application Service		Explicitly defined exposed application behavior.	
Realization		Realization of a more abstract entity.	
Node		A computational or physical resource.	

Table 1: Definitions of *e³value* and ArchiMate concepts. The first parts lists corresponding concepts. Using a business interface to represent a port is optional. Using a Serves relation to represent a value transfer is optional too.

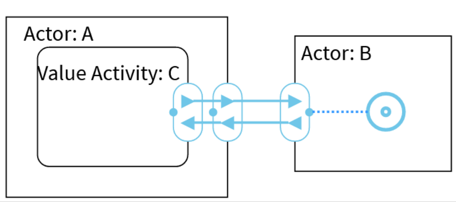
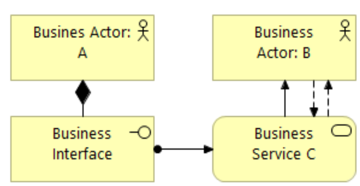
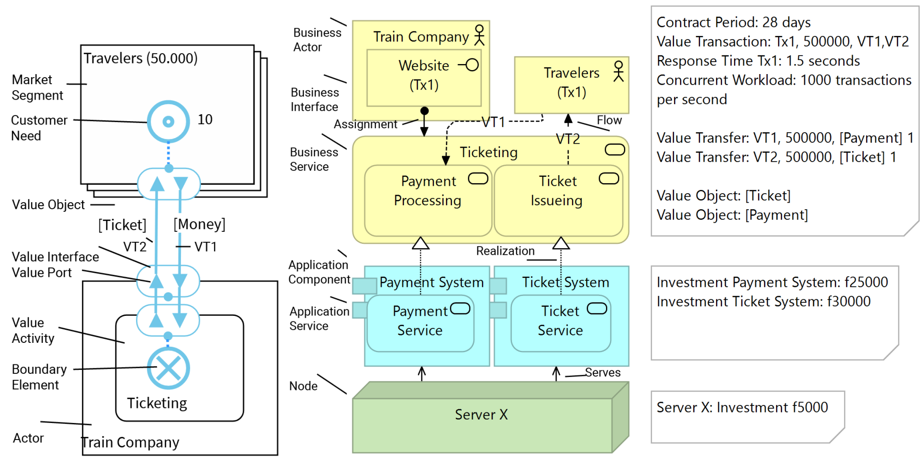
No	Guideline	Additional advice
G9	An *e³value* activity connected through a value exchange to a need of an actor is mapped to an ArchiMate business service serving the actor. 	If B contains a boundary element instead of a need, the direction of the serving relation would be reversed. 

Table 2: Transformation guideline G9.



(a) (b)

Actor	VA	Tx	#	VT	#	VO	VOo
Travelers	Ticketing	Tx1	500000	VT1	500000	Ticket	1
				VT2	500000	Money	1

Fig. 1: e^3*value* model and ArchiMate model of a toy example.

4.1 e^3*value*

Relevant definitions of e^3*value* and ArchiMate concepts are given in table 1. The e^3*value* model of Fig. 1 shows an *actor* train company exchanging *value objects* (tickets and money) with a *market segment* travelers. This is done through a *value activity* Ticketing. To quantify an e^3*value* model, we use a so-called *contract period*, which is the period in which actors perform the transactions represented in the e^3*value* value network. A quantification says how large a market segment is, how often consumer needs occur, what the monetary value of money flows is, etc. In Fig. 1, there are 50 0000 travelers with each on the average 10 ticket needs in the contract period.

The *transaction table* at the bottom of Fig. 1 contains a quantification of the single transaction present in the value network. It says that transaction Tx1 occurs 500 000 times and consists of two value transfers, VT1 and VT2, through which tickets and money pass hands. These numbers are computed by the e^3*value* tool based on cardinality information provided by the tool user.

4.2 ArchiMate

The right part of Fig. 1 contains an ArchiMate model. The *business layer* of this model has been designed following the guidelines of our previous work. The crucial guideline G9 is shown in Table 2. Fig. 1 shows two ArchiMate actors, *Train Company* and *Travelers*, and a *Ticketing* service decomposed into two subservices, *Payment Processing* and *Ticket Issuing*. In general, we define one (sub)service for each value transfer entering or leaving a business actor.

In this EA, these services are implemented in two applications that run on the same server.

We explain the remaining parts of Fig. 1 in the section that follows.

4.3 Representing the contract period in ArchiMate

To quantify an ArchiMate model, we need a contract period in ArchiMate too. Workloads, investments and expenses will refer to this contract period. We add the contract period to an ArchiMate model simply as a comment.

Just as in e^3*value*, we can define a sequence of consecutive contract periods, called a *time series*. This is useful for investment analysis, as we will see below. In Fig. 1 the duration of the contract period is 28 days.

4.4 Representing economic transactions in ArchiMate

In e^3*value* an economic transaction is created using *value ports*, *value interfaces* and *value transfers*. Except for value tranfers, ArchiMate does not contain equivalent concepts. Therefore it is impossible to represent economic transactions in ArchiMate without extensions.

To solve this, we add the information in an e^3*value* transaction table to ArchiMate models. In ArchiMate 3.1 one can define attributes for model components.[5]The collection of attributes defined for a component is called a *profile*.

- For each *value object*, we define an attribute of the ArchiMate model. The name of the attribute is the name of the value object. The EA in Fig 1 has two value objects, *Ticket* and *Payment*.
- Each e^3*value* transfer corresponds to a flow in the ArchiMate model. For this flow we define a profile consisting of the attributes name, number of occurrences, and a reference to the value object. The two flows in the EA of Fig. 1 have value transfer profiles with attributes VT1, 500 000 occurrences, and value objects *Payment* and *Tickets*.
- Each transaction in e^3*value* consists of two or more value transfers, where each value transfer is part of a value interface of the two e^3*value* actors connected by the transfer. The connection points are *ports* in e^3*value*. In the corresponding EA, ports may be explicitly represented by *business interfaces* or implicitly by the incidence of a flow relation on an actor. This is a design choice of the ArchiMate model designer.

  We define a transaction profile for these business interfaces and actors, consisting of the transaction name, the number of occurrences, and references

[5] See ArchiMate specification (`https://pubs.opengroup.org/architecture/archimate3-doc/chap15.html#_Toc10045465)/`

to the participating transfers. In Fig. 1. transaction Tx1 is defined for the business interface *Web Sit*e and for the business actor *Travelers*.

Fig. 2 illustrates how value objects, value transfers and value transactions are mapped to the ArchiMate model.

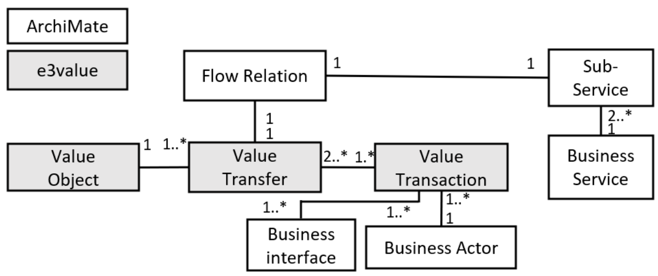


Fig. 2: Relations among e^3*value* concepts (grey) and ArchiMate concepts (white).

4.5 Identification of additional performance requirements

In our approach we identify *performance requirements* derived from the transaction profiles assigned to the business layer elements of the ArchiMate model and the *duration* of the contract period. The transaction profiles imply workload requirements.

The number of transactions in e^3*value* indicate the number of transactions that happen in a stated contract period, say one year. However, an e^3*value* model has no notion of time (except the contract period), so it does not model the distribution of transactions over the contract period. However, for technical scalability it is important to know this distribution, and more specifically the maximum number of transactions per second that can happen in the contract period. Therefore we define *peak economic transactions* requirements to represent this. These are additional to the requirements derived from the transaction table. Fig. 1 contains a Concurrent Workload requirement as illustration.

In certain cases economic transactions needs to be completed within a time frame where the exchange of value objects is useful for the customer. Therefore, we also define *response time requirements.* This is based on work from IT performance metrics. Service performance is measured with the time it takes to execute a single instance of the service [15]. These are additional too. In Fig. 1 we see that Tx1 has a Response Time requirement.

The workload requirements are propagated from the business layer of ArchiMate over the actors that are needed to realize the business services. The response time requirements are propagated down to the business processes, application services and technology services.

4.6 Introducing Investments and Expenses in ArchiMate

e^3*value* has three types of quantifications for investment analysis [5]. First, *investments*. Investments are often needed when a new business idea is implemented. They are done in the first contract period of a time series and are subtracted from the revenue generate in that period.

Second, *fixed expenses*. These are the expenses that do not change from period to period, for example maintenance costs of IT systems. Fixed expenses can be specified for value activities, market segments and actors.

Finally *variable expenses*. These are the expenses associated with the execution of a single economic transaction, for example power consumption of IT systems, license fees, or acquisition of new hardware. The more economic transactions there are, the higher the total variable expenses are. Variable expenses are associated with value transfers, through the value ports in e^3*value*.

Our main strategy is to collect investments and expenses from an ArchiMate model and insert these into the e^3*value* model. ArchiMate [13] has so-called *internal active structure* elements, which are actors that can be hired, bought or built. For these, we define a profile consisting of the attributes *investment* and *fixed expenses*.

For *behavioral* elements (e.g. business processes and application services), we define the attribute *variable expenses*. The amount of these expenses depends on the number of executions of the behavior.

Finally, we define a profile consisting of the attributes *aggregated investments*, *aggregated fixed expenses*, and *aggregated variable expenses* for the ArchiMate business actors and services that correspond to e^3*value* actors and value activities. The investments in and expenses of the application and technology layers can be aggregated in these profiles and then transferred to the e^3*value* model. The aggregation can be done using the scripting languages of the ArchiMate modeling tools. Transfer to the e^3*value* model requires an update of the e^3*value* tool with an import function.

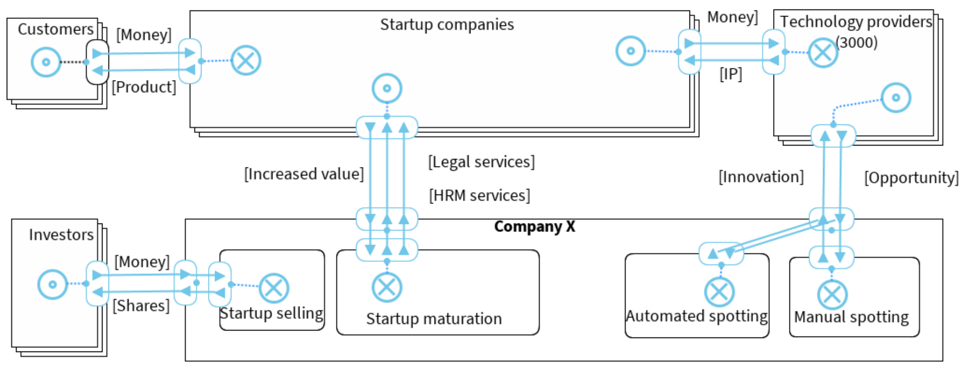
5 Case Study: Company X

5.1 The case company and its value model

Company X is responsible for building startups based on an acquisition of intellectual property, it has a portfolio of about 50 startups. The main goal of the organization is to increase the share value of the startup and finally sell the startup to other investors.

To validate the potential usefulness of our approach, we applied it to a company that is planning to upscale its business. We call the company X. Its business is to scout new technology, acquire the IP of that technology, create a startup for it, grow the startup and then sell it. Every year Company X identifies 3000 possible innovations based on pre-determined criteria per year. It uses a combination of automated technology spotting and manual technology spotting (Fig. 3).

For automated technology spotting they have developed bespoke technology. It accounts for 90 percent of all spotting activities. The split in spotting activities leads to two transactions. Automated spotting leads to transaction Tx1 and is executed 2700 times. Manual spotting leads to transaction Tx2, which is executed 300 times. These transactions are listed in Fig 3.



Actor	VA	Tx	#	VT	#	VO	VOo
Technology Providers	Automated Spotting	Tx1	2700	VT1	2700	Innovation	1
				VT2	2700	Opportunity	1
Technology Providers	Manual Spotting	Tx2	300	VT3	300	Innovation	1
				VT4	300	Opportunity	1

Fig. 3: A market scenario for company X. "f"is a generic currency symbol. Pronounce "florin".

5.2 Quantified Enterprise Architecture

Fig. 4 illustrates the part of the EA for spotting innovations, based on the e^3*value* model from Fig. 3.The top annotations contain the results from importing the transaction table from e^3*value*. These now serve as workload requirements.

The architecture of Company X needs to support 2700x automated spotting and 300x manual spotting per 365 days. The transactions are broken up into the value transfers and via the sub-services they are propagated through the EA. For example, the internal search engine needs to find at least 2700 new innovations per 365 days and perform 2700 outreaches. The application service needs to support 2700 automated patent identifications and 300 manual patent identifications.

We have also annotated the model with illustrative investments and expenses for the automated spotting service. These are illustrated in the left part of the

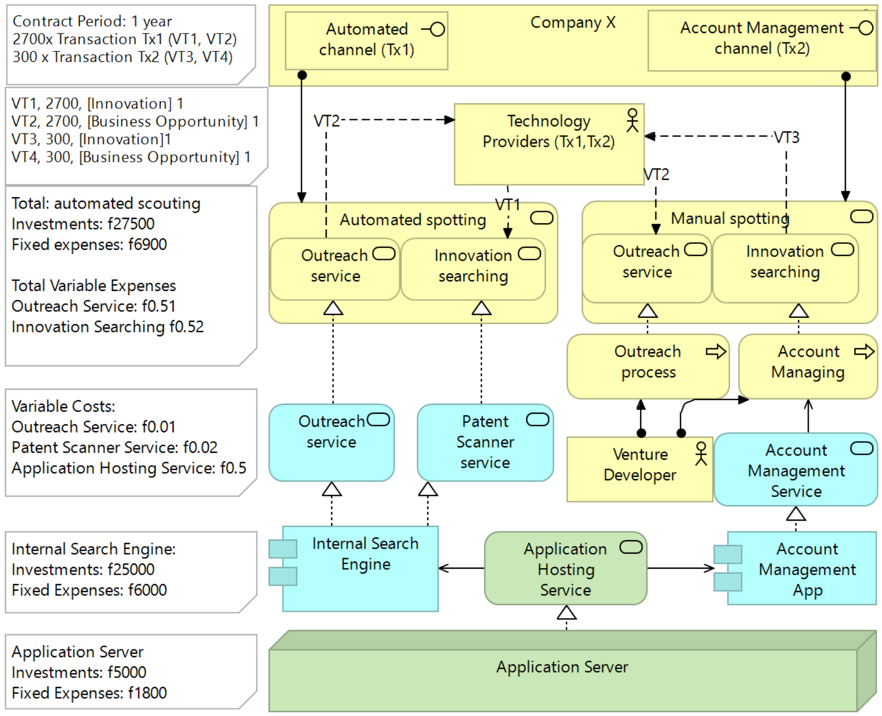


Fig. 4: Architecture for innovation spotting

figure. The investments and fixed expenses are aggregated to the automated spotting service and from this inserted in the e^3value model through the automated spotting value activity. The variable expenses are aggregated to the services and this way linked to the value transfers associated with the flow relations. From this they are inserted into the e^3value model at the value transfer level.

5.3 Initial expert evaluation

We presented the quantified EA model to the enterprise architect of Company X and asked his opinion about the usefulness for their investment decisions. In his opinion, the quantitative alignment techniques are useful. He explained that Company X desires an approach where they can simulate the effects of different increases in economic transactions in the business model on the IT architecture. Using workload requirements based on economic transactions is a suitable way to reason about what the IT must support in that scenario according to the enterprise architect. It will help them identify the possible bottlenecks and provides a good overview of how far the economic transactions propagate down into the architecture. It will help them design the EA better.

The relation of investments and expenses with different scaling scenarios is particularly useful. It will help them get a breakdown of the investments and expenses for different scaling scenarios and improve decision making in investments decisions. The approach answers the questions where to invest and how expensive scaling would be related to the possible added benefits.

The enterprise architect also mentioned that he saw the most benefit in purely digital business models where the business model is completely realized in ICT. However, also in cases where people and business processes are used to implement a business model he saw an added benefit. The workload requirements could be propagated to the business process designers. Since Company X has a business model where parts are directly realized by IT and some by humans he was also interested to see the difference between the two variants in terms of expenses and scaling capabilities. Using operational data in the future is a nice to have. But using workload requirements is sufficient.

6 Discussion

6.1 Answers to research questions

ArchiMate can represent the economic transactions of e^3*value* (Q1) by exporting the transaction table into ArchiMate. The transactions are stored in custom profiles and assigned to ArchiMate elements. Performance requirements are identified (Q2) from the economic transactions and propagated down the ArchiMate model. Additionally peak concurrent transactions and response time requirements are identified as well. ArchiMate model elements can be extended (Q3) with a custom profile for investments and expenses and fed back into e^3*value*(Q4) because we have a custom profile aggregated expenses at the business services. These are mapped to the value activities and value transfers. Our initial evaluation suggested that these quantitative alignment techniques (Q5) provide sufficient information for investment decisions. It provided a breakdown of the investments and expenses needed per scaling scenario. It answers the question how expensive scaling is related to the possible additional benefits.

6.2 Validity

Internal validity is the extent to which the outcome of an experiment has been produced by the treatment. We followed the design guidelines that are our answers to Q1 - Q4 in our case study, so that at least we can say that this application of the guidelines produced the required outcome. The validation of the result by the enterprise architect (Q5) is encouraging but might also have been affected by the experimenter effect: Because the enterprise architect knew we had produced the models, he evaluated the models positively. We tried to reduce this threat to validity by asking specific questions about how the enterprise architect would use the results. Also, because we offered X to be a beta user of our tool implementation of these techniques, the enterprise architect has an interest in making this tool as useful as possible for them. This reduces the risk of an experimenter effect.

Another open issue is the *external validity* of this treatment. Can other people use these our method and come up with similar results? Does our method work for all companies? Are the resulting EAs useful for other companies too? To answer these questions we need to do more case studies and experiments, in which we ask other people to use these guidelines for other companies.

A higher-level external validity question is whether guidelines like these can be used with other business modeling and EA languages. Achieving that level of generality is not our goal. Since our results are derived from an analysis of the metamodels of e^3*value* and ArchiMate and refined in experiments and case studies using these languages, we do not expect generalizability beyond these languages.

6.3 Lessons learned

Realizing complete equivalence between ArchiMate and e^3*value* is not possible. Nor is it desirable. Combining all this information in one model would result in unmanageable models that are hard to understand. Separating commercial business models from EA models allows us to communicate with management and technical personnel separately. Keeping these models quantitatively aligned improves decision-making about ICT investments.

ArchiMate lacks constructs for generating economic transactions. However, using the profiling mechanism of ArchiMate the calculated transaction can be imported into ArchiMate and mapped to similar concepts. This way we do realize the required traceability between e^3*value* and ArchiMate.

7 Future work

In order to further test the internal and external validity of our conclusions, we must first implement these techniques in the e^3*value* tool[6] and an ArchiMate tool. The tools should allow export of quantified transactions from e^3*value* to ArchiMate, and export of investments and expenses from ArchiMate to e^3*value*. Using this, we can test our ideas in more cases with different stakeholders and tool users.

Finally, methodological integration in enterprise architecture development methods (e.g TOGAF [14]) is also planned.

References

[1] Z. Derzsi et al. “Assessing feasibility of IT-enabled networked value constellations: A case study in the electricity sector”. In: *International Conference on Advanced Information Systems Engineering*. Springer. 2007, pp. 66–80.

[6] https://e3web.thevalueengineers.nl/

[2] J. L. Dietz. “Understanding and modelling business processes with DEMO”. In: *International Conference on Conceptual Modeling*. Springer. 1999, pp. 188–202.

[3] W. Engelsman et al. “Realizing Traceability from the Business Model to Enterprise Architecture”. In: *International Conference on Conceptual Modeling*. Springer. 2019, pp. 37–46.

[4] W. Engelsman et al. “Transforming e3value models in ArchiMate diagrams”. In: *2020 IEEE 24th International Enterprise Distributed Object Computing Conference*. Springer. 2020, pp. 11–20.

[5] J. Gordijn and R. Wieringa. e^{3}value*User guide*. https://e3value-user-manual.thevalueengineers.nl/. The Value Engineers, 2021.

[6] J. Gordijn and J. M. Akkermans. *Value Webs: Understanding E-business Innovation*. www.thevalueengineers.nl. The Value Engineers, 2018.

[7] M.-E. Iacob and H. Jonkers. “Quantitative analysis of enterprise architectures”. In: *Interoperability of Enterprise Software and Applications*. Springer, 2006, pp. 239–252.

[8] M.-E. Iacob et al. “From enterprise architecture to business models and back”. In: *Software & Systems Modeling* 13.3 (2014), pp. 1059–1083.

[9] S. de Kinderen et al. “Bridging value modelling to ArchiMate via transaction modelling”. In: *Software & Systems Modeling* 13.3 (2014), pp. 1043–1057.

[10] J. Miguens et al. “A Viewpoint for Integrating Costs in Enterprise Architecture”. In: *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*. Springer. 2018, pp. 481–497.

[11] J. F. Moore. *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*. Harper, 1996.

[12] A. Osterwalder and Y. Pigneur. *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons, 2010.

[13] The Open Group. *ArchiMate 3.01 Specification*. Van Haren Publishing, 2017.

[14] The Open Group. *TOGAF Version 9*. Van Haren Publishing, 2009.

[15] I. Traore et al. “Software Performance Modeling using the UML: a Case Study”. In: *Journal of Networks* 7.1 (2012), p. 4.

[16] R. Wieringa et al. “A business ecosystem architecture modeling framework”. In: *2019 IEEE 21st Conference on Business Informatics (CBI)*. Vol. 1. IEEE. 2019, pp. 147–156.

[17] R. J. Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.

[18] J. A. Zachman. *The concise definition of the Zachman framework*. https://www.zachman.com/about-the-zachman-framework. 2017.

[19] Z. Zhou et al. “An evaluation of quantitative non-functional requirements assurance using ArchiMate”. In: *IEEE Access* 8 (2020), pp. 72395–72410.