

Checking the Alignment of Value-based Business Models and IT Functionality

Novica Zarvić*, Roel Wieringa, and Pascal van Eck

University of Twente, Department of Computer Science, Information Systems Group
P.O. Box 217, 7500 AE Enschede, The Netherlands
{n.zarvic,r.j.wieringa,p.a.t.vaneck}@ewi.utwente.nl

Abstract. Business–IT alignment is an ongoing activity of high importance for the success of a business. This is a hard task, especially in the context of value webs in which multiple businesses collaborate with each other to reach a common goal. Value models, as the outcome of the value web exploration phase, represent solutions for business people, but not for software engineers. The latter ones have to come up with a blueprint for the implementation at application level. This can have two faces: Either there are no systems at all and everything needs to be designed from scratch, or it is desired to use as many existing systems as possible. In the latter case, on which we focus in this paper, it must be checked which functionality existing systems should provide and whether they are usable for the given value web context. This affords several design activities, as well as checking consistency between different models. Our approach can be viewed as an *alignment checking method* and also as a *gap analysis*, in case existing systems miss a given functionality.

Keywords. Business–IT alignment, requirements engineering, value modeling, use cases, consistency checking.

1 Introduction

With the opening of the internet in the 1990s the number of companies that cooperated by means of computer networks increased rapidly, and the different kinds of networks that they formed also increased rapidly [1, 2]. We will call such networks “value webs”. The main purpose of a value web is to satisfy specific consumer needs, which one company cannot satisfy alone. Because of increasing significance of value webs, several modeling approaches have been introduced with the aim to help design such business constellations. We have for instance *e³-value* [3, 4], a method for exploring e-Business ideas. *e³-value* focuses on the trading paths of value objects (e.g. services, goods, money) between the business actors in a network. Apart from its graphical representation it also provides mechanisms for assessing the economic sustainability of the value web. Another approach is the *BMO* (Business Model Ontology), which was developed

* supported by the Netherlands Organisation for Scientific Research (NWO), project 638.003.407 (Value-based Business-IT Alignment)

by Osterwalder and Pigneur [5]. BMO takes the perspective of a single company by considering its environment that might be a value web, whereas *e³-value* is taking a network perspective and focuses on the value web itself. Yet another approach that could be used in the context of a value web is the *REA* (Resource–Event–Actor) ontology. REA takes an accounting perspective on the business relations between economic entities and was initially developed by McCarthy [6] and later extended to form an accountability infrastructure for enterprise information architectures [7, 8].

A method such as *e³-value* helps business analysts to explore different networked business ideas. Its outcome can be presented as a possible solution to business managers, but represents a business–IT alignment problem for software engineers, who have to implement the idea at the application level [9]. Currently there exists no systematic approach in providing a blueprint for addressing this challenge. Traditional business–IT alignment approaches, like e.g. Information Engineering are not optimally suited [10], because they were designed for single companies and therefore come short in the networked context [11]. Dietz described recently how to derive use cases from business process models [12], but there exists, to the best of our knowledge, no work that aligns use case diagrams (UCDs) and value models.

In this paper we propose an approach for checking the alignment between a value web and the IT that supports its realization, represented by an *e³-value* model and use case diagrams respectively. We thus consider the functional aspect of business–IT alignment, leaving the quality aspect for later. Section 2 provides brief background information on *e³-value* and the case study considered in this paper. Section 3 analyzes the role IT can play with respect to value webs and Section 4 describes the application of our method, which we extracted by means of case-study-oriented research. In Section 5 we discuss our results, propose future work, and conclude the paper.

2 Background Information

Value webs are networked business constellations that as a whole deliver something of economic value, such as goods, services or data to a consumer. We consider the consumer to be part of the value web. In order to deliver the object of economic value to the consumer, the businesses in the web also exchange objects of economic value with each other. We consider value webs that are supported by information and communication technology (ICT). We follow in our research the *e³-value* ontology proposed by Gordijn and Akkermans [3, 4]. *e³-value* is based on the concept of *economic reciprocity* and it shows “who is offering and exchanging what with whom and expects what in return” [13, p. 86]. In the remainder of this subsection we explain shortly the core concepts of the *e³-value* ontology by means of a real-life example of Non-Governmental Organizations (NGOs) in the domain of international voluntary work [14]. Each NGO has the following administration tasks: It sends volunteers from its own country to projects in other countries and its own projects and it accepts volunteers

from other countries in its own projects. The projects are provided by so-called partner organizations, who pay a fee for the volunteers sent to their projects. The volunteers pay a handling fee for being placed in one of the projects. The overall aim of this collaboration is to learn from other cultures and to help in local social development. Contact between the NGOs is maintained by a supra-national umbrella organization that loosely coordinates the work of the NGOs. The following systems are present at each NGO: A general ledger system for the financial administration, a simple workflow management system (WFM) for placing each volunteer in a project, a project database of available projects, and a customer relationship management system (CRM) to manage information about the volunteers interested in voluntary work. Furthermore, each NGO has its own website. Since the NGOs are independent organizations, the implementations of these systems vary widely and do not provide compatible interfaces. In our example this represents the old business constellation, but one stakeholder in the network is studying the possibility to centralize the WFM and CRM systems and taking over the matching process from the NGOs. This represents the new business situation. The question to be solved is how this can be done such that the umbrella organization works profitable, while the NGOs perform better in terms of cost, quality, or both.

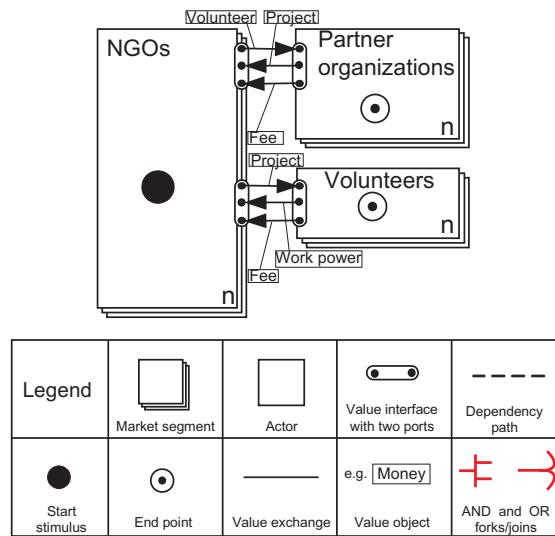


Fig. 1. NGO value model for old business constellation.

In Figure 1 the e^3 -value model for the old business constellation is shown. The old constellation consists of the market segments NGOs, partner organizations and volunteers, whereas the new business constellation (figure 2) contains

additionally an umbrella organization (represented by the dotted line) that is meant to be the new host and owner of the CRM and WFM. The new business constellation allows now for doing global matching.

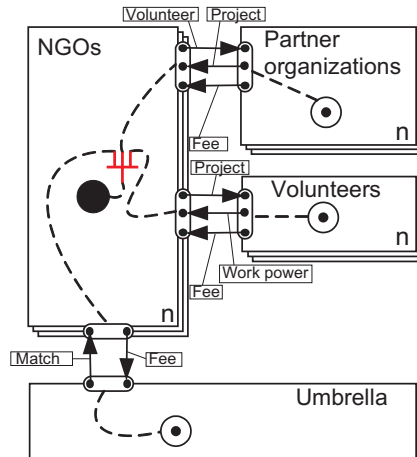


Fig. 2. NGO value model for new business constellation.

In e^3 -value a business actor is a profit-and-loss responsible business unit, like e.g. the umbrella organization. It is represented by a rectangle. A market segment is a group of actors that share the same needs and is represented as three stacked actors such as the NGOs. Actors and market segments exchange value objects (e.g. fee for matching functionality). This is realized through value ports, which are located at the value interfaces. A value interface consists of in and out ports that belong to the same actor or market segment. A value exchange is used to connect two value ports with each other and is shown as line between the value ports. A dependency path begins with a start stimulus, which in turn usually represent some kind of need. In our example the start stimulus is placed at the NGOs, because they want volunteer work to happen. Note that a dependency path is not the same as a process. For example, the AND-fork in Figure 2 merely specifies that to let voluntary work happen, both projects and volunteers are needed. It does not specify that these have to be acquired in parallel. In fact, the NGOs basically follow a sequential process in which projects are acquired first, which are then offered to potential volunteers.

3 On Value Model Concepts and IT Functionality

3.1 Services and Value Object Transfers

We view a value web as a web of services, and define a service to be an interaction between two or more parties, which is of value for the client, and is in most

cases of intangible nature. An e-service is a service provided online. The visible interactions in an e^3 -value model are the value object transfers between the actors. Note that not all arrows represent services, or e-services respectively, because e^3 -value is based on the principle of economic reciprocity, meaning that an arrow is either a service, or the *obligation to refund* “somehow” the service provision. This does not necessarily imply value objects of monetary character (fee, payment, money, etc.), but could be also anything else of value. But the concept of a *value object* needs to be further elaborated, because it is a rather abstract entity that could be nearly everything an actor could consider valuable. According to the e^3 -value developers a value object can be a good, a service, money or even an experience [3]. We almost agree, but our view differs inasmuch that we put the concept of a service at the beginning of the equation, which results in the following view: *Every service in the value web is represented as a value object transfer, but not every transferred value object in the value web is a service.* For instance, a physical good is by definition not a service, but its transfer from one actor to another is a service. Therefore we replace service by data and activities and propose a value object taxonomy that consists of six categories. These are (A) monetary value objects, (B) data value objects, (C) physical value objects, (D) activity-based value objects, (E) experience value objects, and (F) other value objects, like shown in figure 3 below.

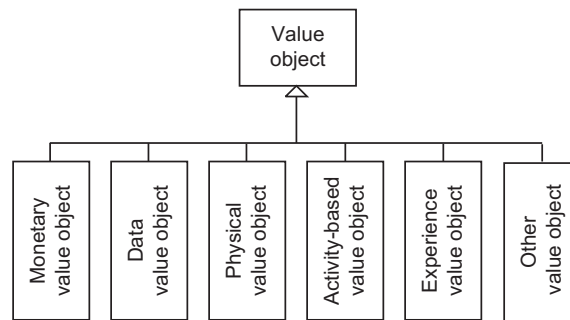


Fig. 3. Value object taxonomy.

A *monetary value object* refers to objects such as money, fee, or payment, which can be expressed in some currency. In most cases the transfer of a monetary value object will represent the obligation to refund a service (law of economic reciprocity). But this is not a strict rule, when thinking of financial services, e.g. a personal credit at a bank. A *data value object* may refer to data or information, like e.g. online newspaper or weather forecasts on the internet. *Physical value objects* (C) refer to tangible goods (e.g. book, house, chair). While selling and delivering them to a client, a service is provided. A further category in our taxonomy is represented by *activity-based value objects*. Take for instance security, where certain activities are needed to assure and maintain it, or reviewing a

paper for a conference. This category is most similar to the traditional view of a service (e.g. taxi ride, hair cut, or the like, where certain activities are needed to provide it). An *experience value object* refers to something a client experiences and finds valuable. Often such experiences cause satisfaction on the client's side. A popular example is to listen to a music stream on the internet. *Other value objects* is the last category of our taxonomy and encompasses all value objects that do not fall under one of the mentioned categories.

Now, considering the nature of such value objects, or more precisely the nature of their transfer from provider to client, helps us already to reason about IT functionality needed. Monetary value objects suggest that some kind of financial system (billing system, etc.) is needed to handle the money flow. Data objects suggest the presence of a database, containing all the data and information (e.g. in form of PDF files, mp3-files, or even XML data). Physical value objects might suggest that a provider needs an ordering system to manage the reorder process for the goods to be delivered (e.g. catalog company needs always to have a certain stock of goods in their warehouses in order to be able to deliver in time). For the remaining three categories this is often not so obvious. However, IT functionality must be planned in a more detailed way. The following subsection investigates the role of IT support during the entire service consumption cycle.

3.2 Exploring the role of IT support during the service consumption process

Exploring in which ways IT can support services can give us guidelines about identifying required IT functionality in a fairly detailed manner. We distinguish the following tasks of IT support: *execution*, *registration*, *monitoring* and *control*. We map these against the consumption cycle of an service. The service consumption cycle contains following steps: *information provision about service*, *access to service*, *delivery of service*, *payment of service*, and *termination of service*. As result we get the framework in figure 4, which shows all possible IT functionalities for each phase in the service consumption cycle.

Note, this is an exhaustive list which does not require that all functions need to be present in a working implementation. A pure e-service will probably have most functionalities from the execution column and maybe some from the other columns. Consider for example that a service provider, after receiving an online order, sends an email with invoice number to the client asking him to make a bank transfer. After receiving the money the provider will deliver the service to the client. Clearly, in this example no online payment is provided, so that there is also no such IT functionality supported. It could even be that the registration that the service was payed is done by comparing account statements by hand. Nevertheless we would still talk about an e-service, if its delivery takes place online.

	Execution	Registration	Monitoring	Control
Information about service provision	Providing information online	Registering that info was provided	Comparing info provision with a norm	Taking action to change info provision
Access to service	Providing online access	Registering access has taken place	Comparing serv. access with a norm	Taking action to change access prov.
Delivery of service	Providing the service online	Registering that service was delivered	Comparing serv. delivery with a norm	Taking action to change serv. delivery
Payment of service	Providing online payment	Registering that service was payed	Comparing payment with a norm	Taking action to change payment
Termination of service	Providing termination online	Registering service termination	Comparing termination with a norm	Taking action to change termination

Fig. 4. Identifying IT functionality.

4 An Approach to Check Alignment

In this section we describe our approach to check the alignment of value-based business models and IT functionality. Our approach consists of several activities resulting in specific outcomes.

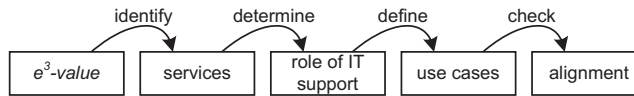


Fig. 5. The logic of our approach.

Figure 5 shows the logic of our approach. The rectangles represent the outcomes. Our approach starts with an e^3 -value model, which is the outcome of a previously performed e-business exploration phase. After identifying the services, which consist of value object transfers, we need to determine IT functions needed for the appropriate services. On this bases we define use cases, before we perform our alignment check. Our approach is explained by means of the NGO case in the following subsections.

4.1 Service Identification

As already mentioned, we view a value web as a web (or collection) of services, which as a whole satisfy a consumer need. A service, as a mostly intangible interaction between a provider and a client, consists of the provision of a value object from provider to client and of a reciprocal value object transfer in the opposite

direction, so that a service consists usually of at least two complementary value object transfers. In our NGO example we can identify three services, namely *project staffing*, *volunteer placement*, and the *matching* service. *Project staffing* consists of three value object transfers between two business participants. For getting volunteers, who do the voluntary work, placed in the projects (activity-based value object) the partner organizations must transfer fees (monetary value object) and offer a project (data value object) by means of information. *Volunteer placement* is a service that also consists of three value object transfers between two business participants, this time the NGOs and the volunteers themselves. The volunteers' main interest is to get placed in a project by the NGOs. The information about this project (data value object) is in turn the value object of their interest. They oblige themselves to pay fees (monetary value object) and to offer the provision of work power (activity-based value object) in return. The last service, *matching* consists of two value object transfers between the NGOs and the umbrella organization. For being able to do global matching (data value objects) and to retrieve relevant information the NGOs are paying fees (monetary value objects).

4.2 Determine role of desired IT support

The activity to determine IT functions is probably the most significant one in constructing a bridge between a business model and IT functionality. Building a blueprint for the implementation at application level heavily depends on performing this activity accurately. The circumstances in which this task has to be performed can be of two different natures: (i) Either there are no systems at all and everything needs to be designed from scratch, or (ii) it is desired to use as many existing systems as possible. In short, for both cases we can use Figure 4 to identify and specify IT functionality. In the first case (i), we can simply consider all cells as a check-list in designing the blueprint. This ensures that the designer does not forget to consider important IT support for the provision of a service. If we deal with the second case (ii), we can use it for analyzing available functionality of existing systems. Afterwards we determine whether this functionality suffices for enabling the services (and value object transfers respectively) from the *e³-value* model, or not (gap analysis). In our NGO case both situations exist. We deal with existing systems, because the ledger system, the project database and the website remain further on at the NGOs. Also the functionality of these existing systems does not change in the new business constellation. All CRM and WFM systems at the NGOs are replaced by one central CRM and one central WFM to be hosted by the umbrella organization. These systems currently do not exist and therefore need to be newly designed. The key requirement of the NGOs on the new CRM and WFM systems is to make matching more efficient and effective. Following constraint limits thereby the solution space:

C1: *The WFM and CRM system must support at least the same functionalities as they held previously in the old business constellation (cf. Section 2).*

The results are shown in Figure 6. Each cross indicates what kind of IT functionality was supported by the multiple WFM and CRM systems at each of the NGOs. From C1 we know that at least this IT support is also required in the centralized, new constellation. This means that the new CRM and WFM systems must have at least this functionality in order to enable the value object transfers that make up the matching service. Empty cells indicate that such IT support is not required.

	Execution	Registration	Monitoring	Control
Provide service info				
Access to service	X	X		
Delivery of service	X	X		
Payment of service		X		
Termination of service				

Fig. 6. Role of IT support according to C1.

4.3 Use Case Definition

Use cases are an integral part of the UML specification¹ and represent a simple but powerful technique for capturing functional requirements of a business system [15–17]. We assume that the reader is familiar with modeling use cases. What makes use cases interesting for the business-IT alignment task in general is that all stakeholders involved get a description of the system functionality that all understand. We are aware of limitations associated with use case modeling in the area of software development (e.g. [18]), which are based among others on the high level of functional abstraction. Our aim is to analyze and define the operational environment of a networked value constellation and thus provide a bridge between the business world and the software world. The concepts of the value-based business models are not data-driven, like in traditional systems planning approaches [19], and further also use very abstract concepts such as the previously discussed value object transfers. The decision for use cases for bridging early requirements with posterior implementation concerns seems appropriate to us and outweighs the concerns.

For the matching service, which we first identified from the *e³-value* model and for which we later set limits regarding the required IT functionality with respect to the service consumption cycle, the main functionality is provided by

¹ UML stands for Unified Modeling Language.
See also <http://www.uml.org/>

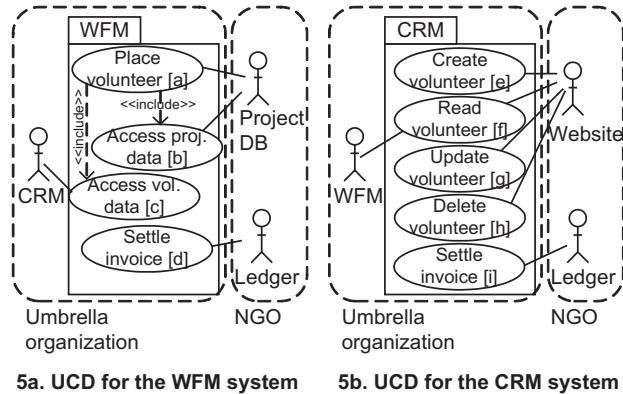


Fig. 7. UCDs for umbrella systems.

the WFM system, or more precisely by one of its use cases, [a] *place volunteer*. Because of C1, the definition of this use case is not difficult. But, for being able to perform the matching service and place a volunteer, the WFM needs access to external data. This is enabled by including following two use cases, [b] *access project data* from project database, and [c] *access volunteer data* from CRM system. [a], [b], and [c] are the use cases representing the four crosses in the access to service and delivery row respectively. The remaining cross in the cell 'payment of service/registration' is enabled by use case [d], *settle invoice*. This four use cases were supported in the old business constellation, and they must also be supported by the IT hosted by the umbrella organization, i.e. in order to satisfy C1 the centralized WFM must contain the mentioned four use cases. We considered so far only the WFM for which we defined the use cases. However, the CRM system is also outsourced to the umbrella organization. Its existence is essential, because the WFM obtains data about the volunteers from it. Without the CRM, among others, the umbrella organization would not be able to offer the matching service. The main functionality of the CRM system is to capture information done by volunteers via the website by letting them [e] *create*, [f] *read*, [g] *update*, and [h] *delete* volunteer data. The CRM, like the WFM has a use case called *settle invoice* [i]. Figure 7a. shows the UCD for the WFM system and Figure 7b. the UCD for the CRM system. Both UCDs capture the required functions for executing and registering the access to the matching service, for executing and registering the its delivery, and for sending invoices to the NGOs. In both figures organizational boundaries were added by a dashed line. This helps pointing out, whether the associations between actors and use cases are in-house or across organizational boundaries.

4.4 Alignment Checking

Checking alignment between a value model and IT functionality requires us to investigate several consistency relationships. Discussing and constructing a

system from different points of view is often done in order to deal with the complexity of a system. Each viewpoint contains a partial specification of the system from a particular perspective. Recently, consistency between value models and UML activity diagrams [20] and Petri nets [21] respectively (business process viewpoint) have been investigated, but there exists, to the best of our knowledge, no work that checks functional alignment at the systems level.

The problem is that each viewpoint, and therefore each model specifying such a viewpoint, has a different focus and is built up of different concepts. As a consequence the models of different viewpoints are not directly comparable. An *e³-value* model depicts value exchanges in a multi-actor network from a business perspective, whereas UCDs capture functionality from an application viewpoint. For checking, whether multiple models are consistent, it is important to concentrate on common constructs present in the involved models. We use communication diagrams [22] to represent IT systems (and functional requirements respectively) in the context of the value web. A communication diagram consists simply of business actors, information systems, and communication channels between the systems. Business actors are simply derived from an *e³-value* model itself, by mapping each actor (and market segment respectively) to an organization boundary in the communication diagram. We follow the system-subsystem argument from systems engineering and view the applications and information systems as the subsystems of the system business actor. For populating the organization/system boundary with subsystems we need to define the latter ones. Sometimes information about the subsystems is given in the case description, like in our case, or this can be done as described in Section 4.3. Communication channels represent data flows between the subsystems. This allows us to check consistency between the business perspective and application perspective. For doing so, we perform three main consistency checks, namely we are checking *business actors*, *use cases*, and *communication channels*.

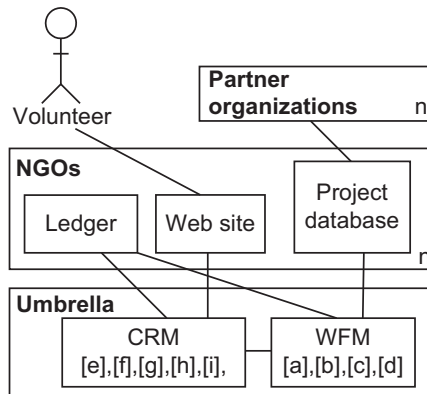


Fig. 8. Communication diagram for (a) checking business actors and (b) checking use cases .

Checking business actors. An e^3 -value model and a communication diagram are consistent, if each actor (and market segment) in the value model are represented in the communication diagram of the value web under consideration and vice versa. In Figure 2 we have four actors (including market segments), which can also be found in the communication diagram shown in Figure 8. Note that we have drawn the volunteer as a stick person, because s/he will not need a specific system implemented, but only internet access to enter data on the website applications of the NGOs.

Checking use cases. UCDs, as the models that we use to capture functional requirements of value models, are consistent with communication diagrams, if use cases are allocated to the same systems in both diagrams. Comparing Figure 7a and 5b with Figure 8, we can see that all use cases [a–i] are placed in the correct and appropriate systems.

Checking communication channels. UCDs, as the models that we use to capture functional requirements of value models, are consistent with communication diagrams, if the defined associations are represented as communication channels in the communication diagram of the value web. We see in Figure 7 that the uses cases that make up the WFM are associated to a number of actors, namely [A] with the CRM, [B] with the Project database, and [C] with the ledger system. The use cases that make up the CRM are associated to following actors, namely [D] website, [E] ledger system. As already mentioned there is already a association with the WFM [A]. We can identify these in the annotated communication diagram shown in Figure 9.

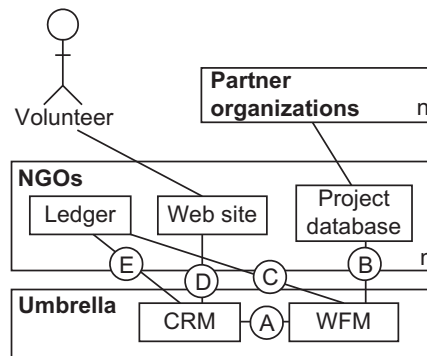


Fig. 9. Communication diagram for (a) checking internal and (b) external communication channels.

Note that sometimes, depending on the case on hand, we are able to be more detailed while checking consistency with respect to communication channels. We can check for (a) *internal* and (b) *external* communication channels. In Figure 7

we can see that the association between a use case of the WFM and the actor CRM is located inside the system boundaries (dashed lines). We referred to this association previously as [A]. All the other associations ([B], [C], [D], and [E]) are of inter-organizational nature. Again, we can identify [A] being inside the umbrella organization, whereas the others are connected with systems at the NGOs.

5 Discussion and Further Research

Summary. In this paper, we proposed an approach for planning IT functionality for a given value model. Our approach considers both extrema in systems design, i.e. planning completely from scratch and considering existing systems. The resulting models are validated by performing an alignment check. Our verification of alignment consists of three distinct consistency checking rules. We have developed this design and alignment approach to address an increased need of business-IT alignment in a value web context.

Difficulty of approach. Checking alignment for this specific case was not very difficult, because the existing systems at the NGOs must have the same functionality in the new business constellation as before. The centralized new systems require also the same functionality as in the old business constellation (which worked already for many years). We captured this functionality for each system by means of use cases and mapped these to the systems to be build in the communication diagram in order to ensure an aligned architecture. Note that, even though this is not treated explicitly in this paper, changes are also required at the existing systems. Although the functionality of these remains the same, new interfaces to the WFM and CRM systems at the umbrella organization are needed. However, this is not necessarily a disadvantage, when compared to the old business constellation. Successful application of our approach requires that the designer understands the problem area well enough, in order to be able to carry out the functionality identification (cf. Section 3.2) and design (cf. Section 4.3) activities.

Benefits for NGOs. The NGOs will profit enormously from the new business constellation. In the new business constellation each NGO has to adapt its interfaces just to one CRM and one WFM system at the umbrella organization, and not anymore to each individual NGO. Furthermore, in the old way NGOs were searching only locally for volunteers, i.e. by using the WFM of another NGO. Now the WFM manages all information, so that global searching and matching is achieved. In the old business constellation the communication caused problems, because it was not fully automated and matching did not span all WFMs at one time.

Gap Analysis. Our alignment checking approach is also very useful in identifying gaps. Assume a simple case, where the comparison between the defined use

cases and the use cases present reveal that not all of the defined use cases are present in an existing system. The non-existent use cases (accompanied by use case documentation) represent already a useful specification of the missing functionality. We know exactly which functionality is missing, and has either to be added to an existing system, or instead a COTS (commercial of the shelf) component or package needs to be bought for this. Note that these are cost-related issues that need to be considered in the *e³-value* profitability calculations. Negative numbers can lead to another value model, e.g. it is not profitable to buy a COTS component and therefore the IT functionality is outsourced. In this case an additional actor would appear. Other scenarios are also conceivable.

Future Research. The NGO case is a valuable study platform. We know the case from previous research and have access to all data. In future research we plan to further specify and formalize the logic of our approach and to investigate how quality is related to the alignment process. More precisely, we aim to investigate the impact of changes at the business level on quality issues at the IT level. For doing so we will define metrics to evaluate the structural properties that result from changes. Further, we intend to suggest ex-post design guidelines for achieving high qualitative value web implementations.

References

1. D. Tapscott, D. Ticoll, and A. Lowy: Digital Capital - Harnessing the Power of Business Webs. Nicholas Brealy Publishing, London (2000)
2. P. Weill and M. R. Vitale: Place to Space: Migrating to eBusiness Models. Harvard Business School Press (2001)
3. J. Gordijn and H. Akkermans: Designing and Evaluating E-Business Models. IEEE Intelligent Systems **16**(4) (2001) 11–17
4. J. Gordijn and H. Akkermans: Value-based requirements engineering: exploring innovative e-commerce ideas. Requirements Engineering Journal **8**(2) (2003) 114–134
5. A. Osterwalder: The Business Model Ontology: A Proposition in A Design Science Approach. PhD thesis, University of Lausanne, Lausanne, Switzerland (2004)
6. W. E. McCarthy: The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review **LVII**(3) (July 1982) 554–578
7. G. L. Geerts and W. E. McCarthy: An Accounting Object Infrastructure for Knowledge-Based Enterprise Models. IEEE Intelligent Systems **14**(4) (July/August 1999) 89–94
8. P. Hraby: Model-Driven Design Using Business Patterns. Springer (2006)
9. R. Wieringa, J. Gordijn, and P. van Eck: Value Framing: A Prelude to Software Problem Framing. In: Proceedings of the 1st International Workshop on Advances and Applications of Problem Frames (IWAAPF) at ICSE. (2004) 75–84
10. N. Zarvić and M. Daneva: Challenges and Solutions in Planning Information Systems for Networked Value Constellations. In M. Weske and M. Nüttgens, ed.: Proceedings of the EMISA 2006 workshop. Volume P-95 of LNI - Lecture Notes in Informatics., Hamburg, GI - Gesellschaft für Informatik (2006) 119–131

11. N. Zarvić, M. Daneva, and R. Wieringa: Value-based Requirements Engineering for Value Webs. In P. Sawyer, B. Paech and P. Heymans, ed.: Proceedings of the 13th Working Conference on Requirements Engineering: Foundations for Software Quality (REFSQ 2007). Volume 4542 of LNCS., Berlin Heidelberg, Springer Verlag (2007) 116–128
12. J. Dietz: Deriving Use Cases from Business Process Models. In I.-Y. Song et al., ed.: Proceedings of the 22nd International Conference on Conceptual Modeling (ER 2003). Volume 2813 of LNCS., Berlin Heidelberg, Springer Verlag (2003) 131–143
13. J. Gordijn: Value-based Requirements Engineering: Exploring innovative e-Commerce ideas. PhD thesis, Free University of Amsterdam (2002)
14. P. van Eck, R. Wieringa, and J. Gordijn: Risk-Driven Conceptual Modeling of Outsourcing Decisions. In P. Atzeni et al., ed.: Proceedings of the 23rd International Conference on Conceptual Modeling (ER 2004). Volume 3288 of LNCS., Berlin Heidelberg, Springer Verlag (2004) 709–723
15. A. Cockburn: Writing Effective Use Cases. Addison Wesley (2001)
16. I. Alexander and T. Zink: Introduction to systems engineering with use cases. Computing & Control Engineering Journal **13**(6) (December 2002) 289–297
17. I. F. Alexander and N. Maiden: Scenarios, Stories, Use Cases. John Wiley and Sons (2004)
18. D. G. Firesmith: Use Cases: The Pros and Cons. In C. F. Bowman, ed.: Wisdom of the Gurus: A Vision for Object Technology, New York, SIGS Books Inc. (1996) 171–180
19. J. Martin: Strategic Data Planning Methodologies. Prentice Hall, New Jersey (1982)
20. Z. Zlatev and A. Wombacher: Consistency Between e^3 -value Models and Activity Diagrams in a Multi-perspective Development Method. In R. Meersman and Z. Tari, ed.: Proceedings of CoopIS/DOA/ODBASE 2005. Volume 3760 of LNCS., Berlin Heidelberg, Springer (2005) 520–538
21. L. Bodenstaff, A. Wombacher, M. Reichert, and R. Wieringa: Monitoring Collaboration from a Value Perspective. In E. Chang and F. Hussain, ed.: Proceedings of the IEEE-DEST 2007, Cairns, Australia. IEEE-DEST 2007, Los Alamitos, IEEE CS Press (2007) 134–140
22. R. Wieringa: Design Methods for Reactive Systems. Morgan Kaufmann, San Francisco (2003)