Towards Information Systems Design for Value Webs

Novica Zarvić*, Roel Wieringa and Maya Daneva**

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

Abstract. In this paper we discuss the alignment between a business model of a value web and the information systems of the participating companies needed to implement the business model. Traditional business-IT alignment approaches focus on one single company, but in a value web we are dealing with various independent businesses. Since a value web is actually a web of services, delivered by IT systems owned by different companies, to ensure alignment we need to specify the services and their properties and then map them on the available IT support in the different companies. Such mappings have to be evaluated in terms of their impact on the profitability of participating in the value web of the different companies. We propose techniques to map services to IT support and show how to do commercial trade-offs.

1 Introduction

In the past decade, the problem of business-IT alignment has become considerably more complex than it was before, because businesses now cooperate in value webs in which they must align their IT services to each other. By values webs we mean networks in which profit-and-loss responsible businesses provide e-services to each other or to consumers [1]. Traditional approaches to business-IT alignment, such as information systems (IS) planning methodologies [2,3] were designed for single companies, but not for value webs [4]. In our research we represent business models graphically by using the e^3*-value* methodology [5,6]. An e^3*-value* model is, thus, representing a value web that we will use to represent e-services. These e-services need to be analyzed and then mapped to functional and quality specifications of relevant information systems. Respectively, any change in the specification of these systems, e.g. because some implementations may be too expensive, may lead to an adjustment of the value model. Figure 1 gives an overview on the models involved in our research. We study the following design questions: How are e-services represented in e^3*-value* and mapped to IS functions? How to relate e-service quality properties to IS quality properties? In which situations to adjust the value model (or even drop its implementation)?

In Sec. 2 we describe the relationship between services and IS properties. In Sec. 3 we show how to use e^3*-value* to represent value webs by means of an working example.

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

** supported by the Netherlands Organisation for Scientific Research (NWO), project 632.000.000.05N01 (Collaborative Alignment of cRoss-organizational ERP Systems)

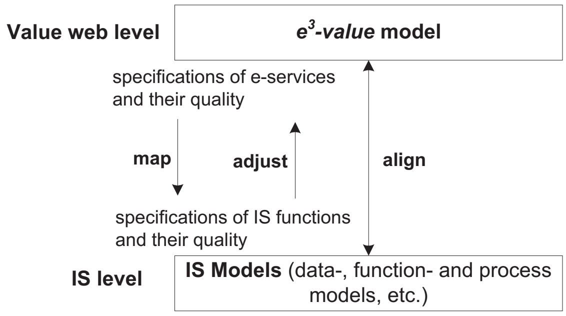

Fig. 1. Aligning value webs with IS models.

Section 4 discusses how to map e-services to IS functions, characterizes the relationship between service quality and software quality, and further describes possible impacts on the value web. In Sec. 5 we conclude the paper.

2 Services and IS properties

A service is defined to be an interaction between a service provider and a service client that has value for the client (who usually offers something of value, such as money, in return) [7,8]. Examples of services are cleaning services, a haircut or the provision of a taxi ride. We define an e-service to be a service delivered over an electronic network. An example is an internet radio service [9], where multiple businesses act together in order to satisfy a customer need. Web Services are often considered to be e-services [10], but they are implementation mechanisms of machine-to-machine interactions over a network. E-services are implemented by means of software systems. We define the *function* of a software system as the interaction between the software and its environment, triggered by some event and with an added value for some stakeholder in the environment. Examples are answering a query (trigger is a question by a user, the answer has added value to the same user), reordering an item when the stock is too low (trigger is a condition change, the reorder is of value for the shop owning the software product that does the reordering), or producing a periodic report (trigger is a tick of the clock, report is presumably useful for whomever reads it). We call all properties of a software system that are not functions and yet have an added value for some stakeholder *quality attrributes* (often called non-functional attributes). Typically, quality attributes are properties of functions. For instance, answering a query should follow a certain response and processing time (time behaviour), reordering an item should prevent unintended access and resist deliberate attacks (security), or producing a periodic report should happen according to conventions or regulations in law (compliance). *Time behaviour*, *security* and *compliance* are some of the quality attributes specified in the ISO 9126 standard [11]. In addition to software quality properties (at IS level) there are service quality properties (at value web level). We define *service quality* as being any service property that adds value to the service. The quality of a service is whatever the client perceives it to be [12]. For instance, a potential reader will not be interested in an online-article, if the download time would take hours. Service quality properties need to be realized by software quality properties. In Sec. 4 we show how they relate to each other.

3 Using e^3-*value* to represent value models

We will illustrate the challenge of mapping e-services to IT support, and of adjusting e-services to available IT support, by means of the small example shown in Fig. 2.

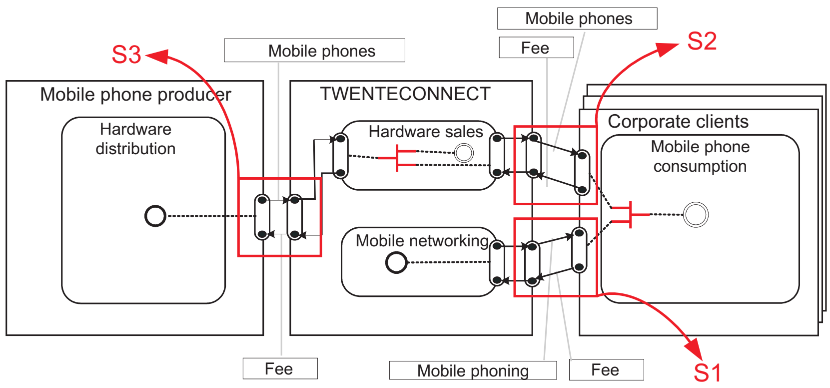

Fig. 2. The TwenteConnect case (regional mobile phoning).

Consider a small telephone company named TwenteConnect, that serves a regional market. The company has been providing so far only fixed land-line services and did not sell any hardware components such as cell phones to their customers. Now, TwenteConnect wants to expand to the area of mobile phone services, again in the same region. Their expansion plan says that before starting to target private clientele, they will run a test phase with corporate clients. The goal is to provide the local police and the staff of the local hospital with mobile phone connections, including mobile phones. As far as TwenteConnect does not produce the hardware (mobile phones) on its own, it relies on a collaboration with a mobile phone producer. In e^3-*value* a value model[1] shows "*who* exchanges *what* with *whom* and expects *what* in return" [13]. It focuses on the business actors and on the reciprocal transfer of value objects between the actors. Note that in Fig. 2 we have non-physical value objects like the mobile phoning provision, but also physical value objects like the mobile phones. This is not uncommon for a 'web of services' and many such constellations consist out of a combination of goods and services [14, p.140]. Following our service definition from Sec. 2, the visible provider/client interactions in e^3-*value* are the value object transfers. So, each value object transfer is a candidate for an e-service.

[1] See http://www.e3value.com/

4 Aligning e-services to IS properties

4.1 Mapping e-Services to IS properties

How are e-services represented in an e^3-value *model?* We call the service delivered to the consumer the *consumer service* of the value web. A scenario path shows how the consumer service is decomposed into services delivered by actors in the value web to each other. In the simple example of Fig. 2 we can identify six value transfers. As far as the *e^3-value* methodology is based on the principle of *economic reciprocity*, we need to consider this rule. This means that a service is represented by at least a value transfer from provider to client, and a value transfer back from client to provider. Such a combination is describing the reciprocal provider/client interaction and is usually grouped into one value interface at each actor. This way of reasoning allows us to separate between provider-specific activities and client-specific activities to be performed. By following the scenario path starting from the corporate clients we can identify three services:

- *mobile phoning*, labeled as S1 in Fig. 2
- *mobile phone delivery*, labeled as S2 in Fig. 2, and
- *mobile phone ordering*, labeled as S3 in Fig. 2.

Consider as an example e-service S1 that consists of two value object transfers which need to be supported by the IS. Now, we can further decompose the activities of S1 into (a) the *provider-specific activities* of the value transfer from TwenteConnect to the corporate clients and (b) into the *client-specific activities* of the value transfer from the corporate clients to TwenteConnect. To identify the IS functions/activities needed to realize the services S1, S2 and S3, we propose to use Porter's value chain as a reference point [15]. It is a generic enough description of activities in any business, and thus, allows us to derive the first list of IS functions/activities needed in support of the services. For instance, provider-specific activities are mainly to be found in the sets of operation and outbound activities in Porter's reference model. Client-specific activities are mainly placed in the set of inbound activities. For further refining these activities we may use for instance function refinement trees [16] or even state diagrams "for finding missing or obscure functions" like suggested by Lauesen [17]. Note that note all identified services in our example are pure e-services. For instance, S3 implies next to the activities taking place over the electronic network also the delivery of mobile phones from the mobile phone producer to TwenteConnect. Note further that S1 and S2 are two independent services offered and provided to the corporate clients. In case we would bundle S1 and S2, we would have to consider the service bundle as one service instead of two.

How to relate service quality attributes to software quality attributes? Clearly, the services in a value web, are realized by the IT systems of the actors. The attainment of service quality in services became an active research area in service marketing [18], which resulted in the definition of different service quality models [12]. The quality of a service in the value web depends on the quality of its enabling IT systems, which explains the relationship between service quality and software quality. In our research we considered at IS level the ISO 9126 standard on software quality [11]. We assume that the reader is familiar with ISO 9126. At service level we considered (i) early results of work by Parasuraman et al. [18], which resulted later in the so-called SERVQUAL model [19], and (ii) recent work by O'Sullivan et al. on service properties [20].

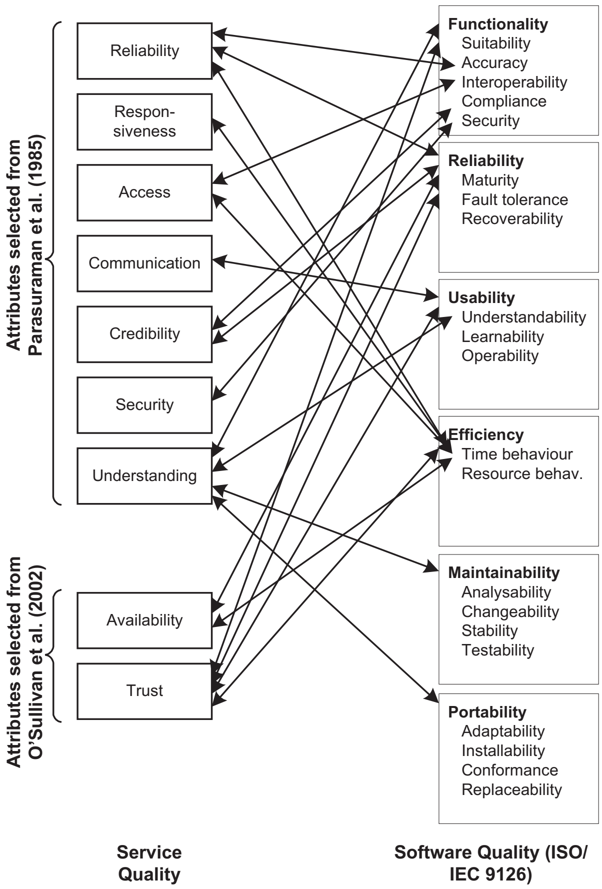

Fig. 3. Relating Service Quality to Software Quality.

This led to the selection of service quality attributes presented in Fig. 3 on the left. We used our service quality definition as a criterion for the selection. We noticed that O'Sullivan discusses many properties, e.g. obligation for payment, penalties, etc. These, however are rather business rules and do not add value to a service. Furthermore, the relationships between service quality attributes and software quality attributes are based on the attribute definitions of the quality models. We analyzed each definition in [18] and [20] and mapped it to those elements of [11], which had same or very similar meaning. The arrows in Fig. 3 represent those mappings. Note that a box on the left can be mapped either to a box on the right (representing a set of software quality attributes), or directly to a single software quality attribute inside a box. In the following we describe shortly the service quality attributes and how they relate to ISO software quality attributes. *Reliability* involves consistency of performance and dependability and some of these aspects can be found at software level in the set of reliability attributes, but also at accuracy and time behaviour. *Responsiveness* concerns the willingness or readiness to provide services. It involves timeliness of service. *Access* involves ease of contact and can be mapped to interoperability and time behaviour. *Communication* means keeping customers informed in language they can understand and can directly be mapped to understandability in ISO 9126. *Credibility* involves trustworthiness, believability and honesty and relates to certain aspects of compliance, but also to the set of reliability.

Security can directly be mapped to the software security quality attribute. *Understanding* involves making the efforts to understand the client's needs and analogies can be found in several sets of ISO 9126. The *availability* of a service is the times when and places where the service is available. This affects time behaviour and several reliability issues. *Trust* is an attribute that deals with trusting the competence and intentions of a service provider. It does not have a direct counterpart in ISO 9126, but relates to several software attributes as can be seen from the figure.

For instance, TwenteConnect might have implemented a direct debiting system for collecting the fees for service S1. To add value to the functionality TwenteConnect decided to collect the fees via a secure communication channel. So the quality attribute *security* for the payment of the fees adds value to service S1. This property can directly be related to *security* at software level and should be considered in the IS design process. Now, consider for instance *fault tolerance* from the set of *reliability* properties. Fault tolerance are those properties that influence the ability to maintain a specified level of performance in cases of software faults. The point is that we cannot have a high performance (e.g. quick responses) if there are too many security checks, each of which takes its time to be completed. In no way security should be comprised to performance. This example makes clear that quality properties should not be considered solely and that it should always be determined which impact they have among each other.

Note that Fig. 3 is actually a set of hypotheses: Each bidirectional arrow is a hypothesis and states the service attribute has impact on the IS attribute. So, each arrow should be elaborated to give guidelines about what exactly this impact might be. Note that there might also be relationships between service attributes and software attributes that are not assigned in the figure, because the similarities are based solely on the attribute definitions. In future work we plan to research these relationships.

4.2 Adjusting Value Models to IS properties

IS design for value webs implies the identification of IS functions. It is desired to reuse available systems. It may turn out that there does not exist available systems capable for realizing functions, which in turn results in the need to design or buy new systems. Such an investment needs to be evaluated financially. Currently two techniques are supported by the e^3*-value* tool for assessing economic sustainability of a value web, namely *net value flow* and *discounted net present cash flow* technique (DNPC), which is based on the well-know net present value (NPV) technique. In the context of our working example, we want to evaluate whether the test phase promisses a positive net value flow for TwenteConnect. We consider that the local hospital needs 20 and the police needs 80 mobile phone connections and mobile phones, so in total 100. For each mobile phone TwenteConnect has to pay 40 Euros to the phone producer (100*40€=4.000€), but sells it for 1 Euro to its corporate clients (100*1€=100€). TwenteConnect sells connectivity as a monthly flatrate for 15 Euros (100*15€=1.500€/month). If we consider the time-period of one year we can assume to get a net income of 14.100€(-4.000€+18.100€=14.100€). Note that the second year will differ in such a way that the income will be 18.000€, because the corporate clients already have mobile phones and we assume two years of average usage of such hardware. So far we did not address the time value of money, but for doing so we can use the DNPC. Take the first time-period were we calculated an undiscounted net value flow of 14.100€. By discounting it, let's say with an interest rate of 5%, we have a value at the start of the first period of just 13.428,57€. If we discount the net value flow for the second year ($18.000/1.05^2$), the value will at the start of the first time-period be just 16.326.53€. The DNPC approach allows to include expenses for *investments*. We might find out that the functionality of available systems does not suffice to realize participation in the value web. In our case we would need to make an investment for a software piece amounting to 3525€, for realizing the business case. In terms of the DNPC this is called an upfront investment, where a special time-period 0 has to be introduced.

Period	Revenues	Expenses	Investments	Net value flow	DNPC
0			3.525	-3.525	-3.525
1	18.100	4.000		14.100	13.428,57
2	18.000			18.000	16.326,53
Total				**28.575**	**26.230,10**

Table 1. Comparing evaluation approaches: net value flow vs. DNPC

Table 1 compares the (undiscounted) net value flow calculations with the DNPC for the two mentioned years (period 1 and 2) with an upfront investment period 0 to include the investment. We recommend the usage of DNPC in order to get a more realistic picture of the economic situation, because it discounts future profit. The investment in this case was so small that it had no impact on the business constellation, but there are other examples conceivable, where the investment exceeds the profit. Now, suppose again that the functionality of available systems does not suffice for realizing the business idea of Fig. 2. TwenteConnect would have following possibilities: (i) developing or buying a new system, (ii) adjusting the value model, and (iii) revoking participation. In case TwenteConnect would need an investment of 35.000€, it would almost three years run negative numbers. As a result TwenteConnect would probably not commit the huge investment of buying or developing a new system. As a result, TwenteConnect would need to analyze whether the value model can be adjusted to supported IS functions of its available systems, and how this would differ from the initial *e^3-value* model. The worst case would appear, if (i) and (ii) are both not feasible. Then possibility (iii) would step in, which means that TwenteConnect would drop the idea of participating in the value web.

5 Conclusions and Future Work

In this paper we addressed how to perform IS design for value webs, by identifying services and functions from the *e^3-value* model and mapping them to the IS level. We also considered quality attributes at both levels and indicated how these could be related. In future research we will investigate some relations in Fig. 3 in more detail, to research the impact, and what guidelines we can derive from that. We will investigate these issues by means of performing case study-oriented research with our industrial partners.

References

1. R. Wieringa, J. Gordijn, and P. van Eck: Value-based Business IT Alignment in Networked Constellations of Enterprises. In: Proceedings of the 1st Internation Workshop on Requirements Engineering for Business Needs and IT Alignment (REBNITA 2005). (2005)
2. J. Martin: Strategic Data Planning Methodologies. Prentice Hall, New Jersey (1982)
3. C. Finkelstein: An Introduction to Information Engineering - From Strategic Planning to Information Systems. Addison Wesley, Sydney (1989)
4. 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
5. J. Gordijn and H. Akkermans: Designing and Evaluating E-Business Models. IEEE Intelligent Systems **16**(4) (2001) 11–17
6. J. Gordijn and H. Akkermans: Value-based requirements engineering: exploring innovative e-commerce ideas. Requirements Engineering Journal **8**(2) (2003) 114–134
7. W. J. Regan: The Service Revolution. Journal of Marketing **27**(3) (1963) 57–62
8. IBM: Services Sciences, Management and Engineering. `http://www.research.ibm.com/ssme/services.shtml` (2004)
9. J. Gordijn, E. Yu, and B. van der Raadt: e-Service Design Using i* and e3value Modeling. IEEE Software **23**(3) (2006) 26–33
10. T. Pilioura and A. Tsalgatidou: E-Services: Current Technology and Open Issues. In: Proceedings of the Second International Workshop Technologies for E-Services (TES 2001), Springer (2001) 1–15
11. ISO 9126-1: Information Technology - Software quality characteristics and metrics (1995)
12. C. Grönross: Service Management and Marketing - A Customer Relationship Management Approach. 2nd edn. John Wiley, Chichester (2000)
13. J. Gordijn: Value-based Requirements Engineering: Exploring innovative e-Commerce ideas. PhD thesis, Free University of Amsterdam (2002)
14. N. Slack, S. Chambers, C. Harland, A. Harrison, and R. Johnston: Operations Management. Pitman Publishing (1998)
15. M. E. Porter: Strategy and the Internet. In: Harvard Business Review. (2001)
16. R. Wieringa: Design Methods for Reactive Systems. Morgan Kaufmann, San Francisco (2003)
17. S. Lauesen: Software Requirements: Styles and Techniques. Pearson Education Limited, Harlow, England (2002)
18. A. Parasuraman, V. A. Zeithaml, and L. L. Berry: A Conceptual Model of Service Quality and Its Implications for Future Research. Journal of Marketing **49** (1985) 41–50
19. V. Zeithaml, A. Parasuraman, and L. Berry: Delivering Quality Service - Balancing Customer Perceptions and Expectations. The Free Press, New York (1990)
20. J. O'Sullivan, D. Edmond, and A. ter Hofstede: What's in a Service? Towards Accurate Descriptions of Non-Functional Service Properties. Distributed and Parallel Databases **12** (2002) 117–133