MANAGING CHANGES IN COLLABORATIVE NETWORKS: A CONCEPTUAL APPROACH

Completed Research Paper

Novica Zarvić
Information Management and Information Systems
University of Osnabrück
Katharinenstr. 3, 49074 Osnabrück, Germany
novica.zarvic@uni-osnabrueck.de

Michael Fellmann
Information Management and Information Systems
University of Osnabrück
Katharinenstr. 3, 49074 Osnabrück, Germany
michael.fellmann@uni-osnabrueck.de

Oliver Thomas
Information Management and Information Systems
University of Osnabrück
Katharinenstr. 3, 49074 Osnabrück, Germany
oliver.thomas@uni-osnabrueck.de

Abstract

Collaborative Networks represent organizational forms that became omnipresent in today's way of making business. Such organizational forms are often established in order to satisfy a complex customer need, which one company could not satisfy on its own. This means that the participating companies are to a certain degree dependent on each other. Managing inter-firm relationships by means of inter-organizational interdependencies represents an important Business-IT Alignment issue. In this paper, we present the Dependency-based Alignment Framework, which represents a conceptual approach for managing changes in Collaborative Networks from a holistic perspective. A detailed and methodologically well-founded approach in the definition and design of our framework is accompanied by a detailed investigation of relevant properties of this design artifact. To demonstrate the applicability of our framework in practice, we introduce a case study, which uses Semantic Media Wiki and the SPARQL query language. Finally, we evaluate our results in an argumentative and deductively descriptive way.

Keywords: Operational Business-IT Alignment, Collaborative Networks, Design Science

Introduction

The advent of the internet as well as advances in related concepts and technologies represent the main driving force for Collaborative Networks (Tapscott et al. 2000). The most apparent trait of Collaborative Networks is that two or more independent companies are working together towards a common goal, which one company could not achieve on its own (Gordijn and Akkermans 2003; Kanter 1994). The emerging field of Collaborative Network research (Camarinha-Matos and Afsarmanesh 2004) has the aim to advance theory and practice that is needed to deal with collaborative organizational forms and is viewed as a completely new multidisciplinary research discipline. Collaborative Networks do nowadays not represent exceptions anymore, but became omnipresent business constellations (Camarinha-Matos and Afsarmanesh 2005; Kanter 1994). Because of their high appearance in reality and the resulting practical relevance, Collaborative Networks require specific attention from the scientific community. The operational practices required to manage Collaborative Networks differ from those used in traditional organizational forms (Zarvić et al. 2007), which in turn means that the management of geographically distributed IT (information technology) resources is a challenging task requiring adjusted forms of IT organizational design (Sia et al. 2010b). These facts enormously influence the way of how to perform Business-IT Alignment in a Collaborative Network context.

Chan and Reich (2007) state that Business-IT Alignment is a highly relevant topic and that it presents now for more than two decades a top IS (information systems) research issue, which is important to both academia and practice. Business-IT Alignment generally refers to the continuous and dynamic process of assuring a fit between business needs and IT (Tiwana and Konsynski 2010). A recent Gartner report claimed the need for improved alignment practices and stated that an integrated IT and business process is a critical business competence (Mahoney and Kitzis 2009). This is confirmed by a recent study, where Business-IT Alignment represented in 2010 a key issue for IT executives and was ranked 3rd after it was given persistent high ranks in the previous years as well (Luftman and Ben-Zvi 2010). Other new research results state that Business-IT Alignment is “a major concern for business executives”, but “there is little published research that attempts to give methodological support” (Bartenschlager and Goeken 2009). This is also in line with Aggarwal’s claim that “there is little agreement on the way of how to achieve alignment as well as how this topic should be researched” (Aggarwal 2009). Summing up, Business-IT Alignment in the context of single companies is still an issue that is not completely solved, but in the context of Collaborative Networks it is considered to be far more problematic and difficult than in single companies (Wieringa 2008).

The problem of performing Business-IT Alignment in Collaborative Networks is a phenomenon that is due to intercompany relationships, where multiple network members work together to achieve a joint outcome. Companies deciding to participate in Collaborative Networks consequently depend on each other. The inter-organizational bundling of resources and activities – often initiated by a complex customer need (Gordijn and Akkermans 2003) – is from its very nature based on mutual dependencies, and for this reason deserves special attention. This statement is supporting Kanter’s (1994) claim that interdependence is the main characteristics of intercompany relationships, where business partners need each other, because of possessing complementary assets and skills (Kanter 1994). Hence, a sound understanding of dependencies and interdependencies in a networked context is needed for being able to manage them properly.

Managing dependency relations is a crucial and complex task for the success of any Collaborative Network (Zarvić et al. 2010). Such dependency relations are representing the core object of investigation in this paper. However, dependency management is not restricted to the business level only, but may also influence the underlying inter-organizational processes or IT architecture, and vice versa. The result is a mutual dependency relation between these perspectives and levels respectively. Dependency relations can be found at every describable enterprise architecture level. Such a dependency relation (e.g. included in a specific model) at one enterprise architecture level might have some impact on other dependency relations at other levels. Consequently, a change in a model at one certain level might cause changes in other models at other levels, which clearly makes the management of dependency relations an important Business-IT Alignment issue. This leads us to the following research question ***RQ:*** *How to perform Business-IT Alignment in Collaborative Networks on the basis of inter-organizational dependencies?*

In this paper we follow a Design Science Research appproach (Wieringa 2010; Hevner et al. 2004), which is of constructivist nature (Kuechler and Vaishnavi 2011) and which provides a solution to the above mentioned practical problem of performing Business-IT Alignment in Collaborative Networks (Wieringa 2008). Our contribution lays in the definition and design of an artifact, namely the Dependency-based Alignment Framework, which enables IT executives (Luftman and Ben-Zvi 2010) as well as business executives (Bartenschlager and Goeken 2009) to holistically align dependency changes in Collaborative Networks.

The reminder of this paper is structured as follows. First we analyze related work on Business-IT Alignment and inter-organizational dependencies by means of structured literature reviews. Then we present our research approach, which is based on nested problem solving. This includes the study of several properties of the designed artifact. We demonstrate the use of our approach by means of a detailed application scenario and evaluate the alignment framework in an argumentative and deductively descriptive way. Finally we conclude the paper and motivate further research.

Related Work and Theoretical Background

Reviewing a scientific field by means of a structured literature search is viewed as the first step in uncovering relevant sources to a topic under study (vom Brocke et al. 2009; Webster and Watson 2002). We approached our literature searches for the fields of Business-IT Alignment and Inter-organizational dependencies by using several electronic indexing services (e.g. Springer Link, ACM Digital Library, Google Scholar, and Citeseer library). Both literature reviews were based on the same structured search process, which is described in the following. The first search concentrates on the field of Business-IT Alignment and includes key words like "align*", "business-it alignment" and "strategic alignment". We also used some alternative terms for alignment like for instance "balance" or "fit". The second search on the topic of inter-organizational dependencies in Collaborative Networks includes key words like "interdependenc*", "inter-organizational relation*" and "dependen*". Further we traced the references in the identified articles in order to identify other relevant sources. The following two subsections provide an overview of the most relevant articles that were retrieved.

Reviewing Business-IT Alignment

In the academic literature Business-IT Alignment has been conceptualized in various ways and represents for more than two decades a top management concern (Chan and Reich 2007; Luftman and Ben-Zvi 2010). Business-IT Alignment refers to the fit between the business and the IT domain, where a distinction between *strategic alignment* and *operational alignment* can be made.

Strategic alignment is more of descriptive nature and tells what has to be done, whereas operational alignment is more of constructivist nature and tells how to do things. The notion of alignment itself was coined in the 1990s by Henderson & Venkatraman (1993). They described alignment from a strategic point of view, which is the predominant category that can be found in the Information Systems community in the Anglo-American area. The driving forces of strategic alignment are either business strategy or IT strategy. The ultimate goal of strategic alignment for an organization can therefore be seen in the achievement of competitive advantage by means of strategy-driven decisions that are expected to be supported by means of IT (Henderson and Venkatraman 1993). Strategic alignment provided, although it tended in its early research to be very theoretical in nature, the basis for later work that is highly relevant from the perspectives of business and IT executives (Campbell et al. 2005).

Operational alignment on the other side is currently mainly covered by the Information Systems Engineering community in the European area (Schlosser and Wagner 2011; Martin et al. 2008; Blankenhorn and Thamm 2008; Wieringa 2008) and is of more systematic nature, because its main aim is to give practical guidelines of how to bring business structures and IT structures in harmony. However, literature on operational alignment still seems to be underrepresented. Schlosser and Wagner (2011) claim for instance that "only little work has been done on the relationship between IT and the business units". The first approaches of operational alignment were provided by diverse structured Information Systems Planning (ISP) methodologies (Lederer and Mendelow 1989). These include approaches such as Strategic Data-Planning (Martin 1982), Business Systems Planning by IBM (Zachman 1982) or

Information Engineering (Martin 1989; Finkelstein 1989). All these methodologies have in common that they follow a top-down approach, neglect legacy systems and were designed for single companies only. However, Tagg and Freyberg (1997) extended these methodologies for distributed, networked environments, but offered again only a top-down systems planning approach. Furthermore, all these approaches are planning the system landscape of a company completely from scratch. These ISP methodologies, mainly stemming from the 1970s and 1980s, neither considered bottom-up nor out-of-the-middle alignment activities. This deficit was first addressed by Enterprise Architecture Frameworks (EAFs), like e.g. the Zachman Framework (Sowa and Zachman 1992), GRAAL (van Eck et al. 2004; Wieringa et al. 2003) and alike. These EAFs are often structured by means of layers. They allow for top-down, bottom-up and out-of-the middle perspectives on Business-IT Alignment. However, layered EAFs mainly provide company-wide documentation structures, but seldom a methodology of how to implement objects, and hence do not provide sufficient support of performing operational Business-IT Alignment. Nevertheless, our literature study as well as our experience in teaching the topic Enterprise Architecture Management show that many operational Business-IT Alignment approaches that have the aim to support top-down, bottom-up and out-of-the-middle alignment activities are situated inside of layered-architecture styled frameworks.

Several Business-IT Alignment definitions have been proposed since this term first appeared (Chan and Reich 2007), but most of them have been shaped from the perspective of strategic alignment (Campbell et al. 2005). As far as we aim to contribute to the category of operational alignment, we stick – in order to be able to provide a crisp definition – only to the structural dimension of alignment (Schlosser and Wagner 2011; Martin et al. 2008; Chan 2001; Jordan and Tricker 1995). In the course of this paper we interpret and define operational Business-IT Alignment as being the functional integration and adjustment of structural configurations, ranging from organization structure to IT structure, and vice versa.

Reviewing Inter-organizational Dependencies in Collaborative Networks

Many publications address dependencies and interdependencies in widely varying ways and apply them to different contexts (Cox et al. 2001). However, there is a subtle semantic difference between dependencies and interdependencies. Dependencies describe unidirectional relationships, whereas interdependencies refer to bidirectional relationships. We define a dependency as being a relationship between two entities, where the state of one entity is correlated to the state of the other. Interdependency refers to a relationship between two entities, where the state of each entity is correlated to the state of the other. This consequently means that two entities are interdependent, when each is dependent on the other. Both definitions are in line with Rinaldi et al. (2001), who thoroughly investigated infrastructure interdependencies.

In order to be able to give a well-structured overview on inter-organizational dependencies and interdependencies in Collaborative Networks, we have categorized our findings. Our categorization is based on conceptual abstraction approaches that are very common in the field of Enterprise Architecture Management, or more precisely in the construction of Enterprise Architecture Frameworks (Zarvić and Wieringa 2006). Thereby we provided a subdivision along four application areas that are relevant for practitioners, such as the previously identified business or IT executives. Our categorization followed further a concept matrix approach (Webster and Watson 2002), where concepts are based on the identified application areas. This enabled us to make a transition from an author- to a concept-centric perspective and to allocate the retrieved dependencies and interdependencies respectively. In doing so the following categorization has been achieved.

Business and organizational dependencies: Emerson (1962) investigated complex community relations and proposed a simple theory of *power-dependence* relations by studying power asymmetries in collaborative structures. *Interdependencies*, as proposed in organizational theory by Thompson (1962), initially took an intra-organizational view to describe three recurring interdependence patterns, which are pooled, sequential and reciprocal interdependence. Kumar and van Dissel (1996) discussed later these configuration patterns – representing organizational structures – for inter-organizational systems settings, directly linking this category to the information systems and application category below.

Product and service dependencies: Dependencies between product parts have thoroughly been investigated by means of Gozinto-graphs (Vazsonyi 1954), which essentially show the product structure including the number of all required product parts. *Resource-dependency theory* assumes that one organization cannot possess or produce all required resources alone. Hence, this gap should be filled by assessing resources that are managed by others (Pfeffer and Salancik 1978). Such resource-resource dependencies are also described by Crowston (1994) within the field of coordination theory. Dependencies have also been investigated with respect to services. The so-called *molecular model, which was* introduced by Shostack (1993), shows relations between services and service elements. Winkler and Schill (2009) have also investigated *service dependencies* in service compositions in the logistics domain. Ludwig and Franczyk (2008) describe dependencies and relations between Service Level Agreement (SLA) elements in atomic and composite services and provide with *COSMA* (Composite SLA Management) a solution for managing these dependencies.

Process, function and task dependencies: Coordination theory provided with *task-task dependencies* and *task-resource dependencies* important input for managing dependencies at process level (Malone and Crowston 1994). Nevertheless, it can be said that the management of dependencies in processes or workflows is an important aspect of BPM (Business Process Management), where we have to deal especially with time-based dependencies (Cox et al. 2001) for managing the order of process steps.

Information systems and application dependencies: ICT (information and communication technology) systems and applications are representing the underlying infrastructure for enabling the process level described above. Consequently, for running inter-organizational processes properly also the dependencies of the underlying ICT infrastructure elements have to be considered (Scheibenberg and Pansa 2008). So-called communication diagrams show e.g. communication channels between ICT systems and applications respectively (Lankhorst et al. 2009). Here an application might be dependent on the data stored in an information system. Another approach that partially tackles dependencies at ICT level is provided by Ball-Rokeach (1985), who introduced the *media-system dependency*. However, this approach is not limited only to the ICT world and elements, but builds on and is linked to the social world.

Other dependencies: *Strategic dependency* situations are widely discussed in literature (e.g. de Padua Alburquerque Oliveira and Cysneiros 2006). However, these strategic dependency situations (Yu and Mylopoulos 1993) refer to dependencies (e.g. actor dependency, resource dependency, task dependency) that are situated in one of the dependency categories described above.

Table 1. Categories of Inter-organizational Dependencies Based on a Concept-centric Perspective

Category/Layer	*Dependencies found in the literature*	*Example modelling notations*
Business and organizational dependencies	e.g. Organizational interdependencies (Kumar and van Dissel 1996; Thompson 1962), dependency paths in *e³-value* models (Gordijin and Akkermans 2003), power dependencies (Emerson 1962)	e.g. SD model in i* (Yu and Mylopoulos 1993), *e³-value* modeling notation (Gordijn and Akkermans 2001)
Product and service dependencies	e.g. resource-resource dependencies (Crowston 1994; Pfeffer and Salancik 1978), product structures (Vazsonyi 1954), service compositions and SLAs (Ludwig and Franczyk 2008; Winkler and Schill 2009)	e.g. Gozinto-graphs (Vazsonyi 1954), SD model in i* (resource dependum) (Yu and Mylopoulos 1993), Molecular models (Shostack 1993)
Process and task dependencies	e.g. task-task and task-resource dependencies (Crowston 1994), time-based dependencies (Cox et al. 2001)	e.g. SD model in i* (task dependum) (Yu and Mylopoulos 1993), UML activity diagrams, Process Flow diagrams, EPC
Information Systems and application dependencies	e.g. dependencies of IT infrastructure elements (Scheibenberg and Pansa 2008), media-system dependency (Ball-Rokeach 1985)	e.g. UML deployment diagrams, IT communication diagrams

In order to give a more comprehensive overview, we have added for each category example modeling notations as listed in Table 1. Nevertheless, it should be mentioned that we observed during our literature review that some authors use the term dependency while talking about a relationship that fulfills the semantic requirements of the interdependency definition. We found two reasons for this phenomenon: On the one hand sometimes only a unidirectional perspective in an interdependency relationship is represented, like for instance a unidirectional dependency between entities in a supply chain (Kumar and van Dissel 1996). On the other hand some authors insert adjectives such as “reciprocal” or “mutual” in front of the term dependency, which results in a synonymous expression. However, this is a bit problematic as those adjectives are after their initial usage often left out in the remainder of the works (cf. Kumar and van Dissel, 1996; Emerson 1962).

Research Methodology

An unanimous agreement among researchers attests that Design Science Research is inherently problem-driven (Holmström et al. 2009). Kuechler and Vaishnavi (2011) define this research method shortly as “the construction of an information technology artifact and its evaluation”. The main objective of Design Science Research is the development of “technology-based solutions to important and relevant business problems” leading to an improved environment by means of creating innovative artifacts, which can be constructs, models, methods and instantiations (Hevner et al. 2004). In this paper we apply this research method as a means to deal with the initially stated problems concerning Business-IT Alignment in Collaborative Networks. More specifically, in order to be able to provide an adequate solution for the initially stated ***RQ***, namely *how to perform Business-IT Alignment in Collaborative Networks on the basis of inter-organizational dependencies*, we follow a nested problem solving strategy (Wieringa 2009). Nested problem solving in Desing Science Research has its origins in the Human Problem Solving approach introduced by Newell & Simon (1972).

Schlitt (2003) describes problem solving in analogy to human behavior. Humans tend to solve problems in an iterative process. At the beginning neither the space of alternatives nor a final design goal is fully determined. In the course of problem solving the problem is decomposed into interrelated sub-problems. This decomposition is repeated until sub-problems arise, which are considered to be solvable by stakeholders. Finally, the solutions of these sub-problems are integrated by means of their derived relations into a total solution. Such a decomposition procedure is from its very nature component-oriented and follows a typical systems perspective. The field of Business-IT Alignment with a focus on structural fit and functional integration between different components and perspectives is also inherently component-oriented (Chan and Reich 2007).

After having described the generally applicable rationale and systematics of problem decomposition, we now specify how we decomposed in particular our ***RQ*** into several sub-problems. These sub-problems can – according to the guidelines regarding methodological soundness in requirements engineering – be sub-divided into practical problems (PPs) and knowledge problems (KPs) (Wieringa and Heerkens 2006; Wieringa 2005). According to Wieringa (2009), in design science PPs and KPs are usually nested into one another. A PP requires some change in the world. It describes the difference between the world under consideration as it is at the moment and as we would like it to be. A KP on the other side represents a deficiency of knowledge about the world. In order to provide an answer to a KP, it is necessary to study the world. A KP does not necessarily require a change of the world. Nevertheless, there is a mutually recursive relation between KPs and PPs, so that it is possible that there is also a PP behind a KP (Wieringa and Heerkens 2006). Holmström et al. (2009) have also recently described this difference. For them PPs and design science belong to exploratory research, where the creation of an artifact is essential. On the other hand they juxtapose explanatory research that equals KPs, where something that already exists is studied. It is interesting to note the author’s claim that “exploration and explanation are not mutually exclusive”, but are “highly complementary” (Holmström et al. 2009).

Our research process follows the above described distinction and is depicted in Figure 1, where we adopted the so-called Refined IS Design Science Research Framework by Wieringa (2010). The first step in decomposing the ***RQ*** is characterized by a practical problem. Relevance is given by the need to provide a structure, which suits the purpose of dependency-based alignment. This leads us to the formulation of ***PP1:*** *How to structure the different inter-organizational dependencies systematically by means of a* *framework?* For providing such a structure we first need to know about the state of the art in Business-IT Alignment as well as about all conceivable dependencies and interdepencies that may be present in Collaborative Networks. These issues have sufficiently been answered in the Related Work and Theoretical Background section above. Wieringa (2010) points out that Design Science Research investigates properties of constructed artifacts, so that we need to learn about relationships and structural properties in frameworks. This justifies our ***KP1:*** *How many alignment perspectives are conceivable in the previously developed structure in PP1?* However, for being able to answer this PP, we make use of the existing knowledge base in this domain and simultaneously add new knowledge to it. Nevertheless, at this point it is also important to gain insight about when the usage of the framework is most relevant. This is a temporal question that can be investigated by mapping the dependency layers against the life cycle of Collaborative Networks, which leads us to ***KP2:*** *Which impact do the identified dependency types have on the life cycle of a Collaborative Network?* At this point – after having dealt with the first three sub-problems – we would be able to perform operational alignment manually, so that our last goal is concerned with the support that IT could provide us for getting an overall solution. This leads us finally to ***PP2:*** *How does alignment need to be performed with the framework from PP1 in order to manage changes in Collaborative Networks and how can this activity be formally supported by means of IT?*

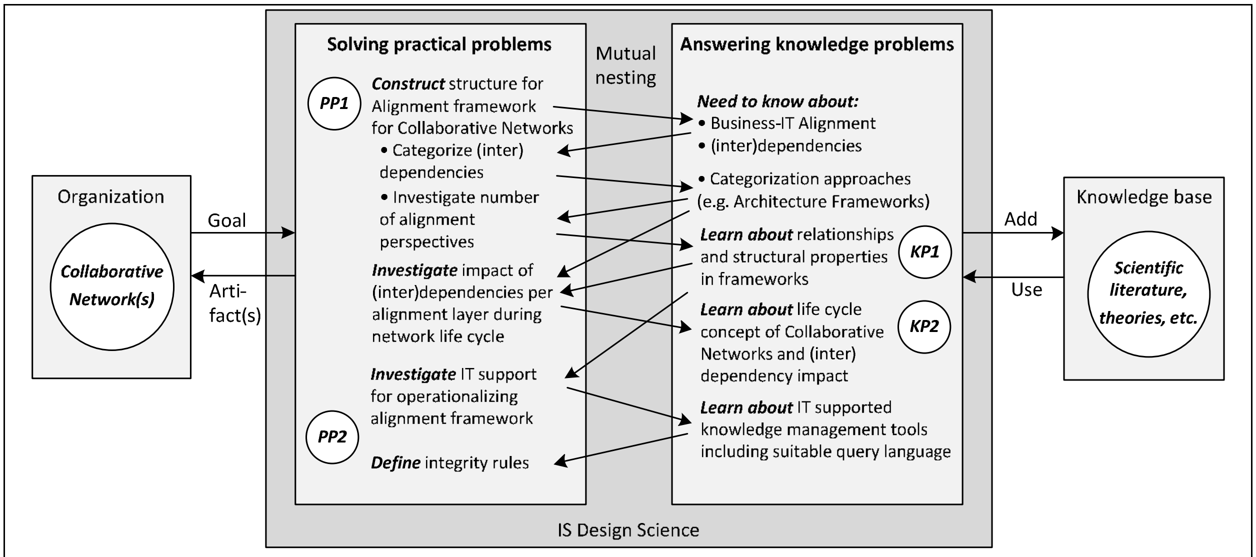

Figure 1. Research Steps Following a Nested Problem Solving Approach in the Refined Design Science Research Framework by Wieringa (2010)

The Dependency-based Alignment Framework

Framework Requirements

The specification of requirements is according to Lauesen (2002) a very difficult task. Requirements include several capabilities that the design artifact must meet. Frameworks are artifacts that are well-known in the IS research area and are often concerned with structuring architectural descriptions (Lankhorst et al. 2009). Generally a framework is described as representing “a real or conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful” (Trienekens et al. 2008). In the scope of this paper we view a framework as a conceptual model that has the aim to provide an abstract structure for the placement of relevant elements, of relationships between these elements as well as their properties for a specific application area. The arrangement of the above mentioned objects of interest inside a framework represents a design decision and can therefore freely be chosen by the framework designer, who has to take care to provide a skeleton-like structure that gives an adequate overview. However, the framework definition leads us also to the first requirement ***R1*** for the framework. Furthermore, for the formation of a framework often multiple perspectives or viewpoints (Gordijin and Akkermans 2003), or a hierarchy of models that is following the systems hierarchy concept, where the components realise the composite system (Wieringa 2003), are used. This justifies ***R2***, where a distinction of (inter)dependencies within one perspective as well as between perspectives is required. The motivation for the final requirement is provided by a top key issue for IT executives, where it is stated that one big problem of alignment is that “organizations need to recognize that it is not how IT is aligned with the business; it is how IT and business are aligned with each other” (Luftman and Ben-Zvi 2010). Hence, ***R3*** requires that alignment is not only a top-down concern that is triggered by business people. Therefore the solution for ***PP1*** is characterized by these three initial requirements, which are also summarized in the next table.

Table 2. Requirements Definition for the Framework

Requirement	*Textual Description*
R1	The framework should provide a skeleton-like structure for all the identified dependency categories.
R2	The framework should provide a distinction between dependencies within a category and between different categories.
R3	The framework should enable top-down, bottom-up and out-of-the-middle alignment activities.

Defining the Dependency-based Alignment Framework

The starting point for defining the *Dependency-based Alignment Framework* has been to decide how to organize and manage the complexity of the multitude of inter-organizational dependencies found in the literature. We started from the notions of *system* and *architecture of a system*. A system is a basic and fundamental concept in systems engineering, which enables us to reason about the world, or even parts of it. We consider a Collaborative Network to be such a system. The systems’ architecture on the other side is the structure of components and their properties including the interactions between these components that realize system-level properties. Wieringa (2003) states that there are two relationships between components and a composite system when decomposing a system into components: (i) components deliver services to the composite system, meaning that services from the composite system cannot be achieved without using the services of other components. Further, (ii) the composite system encapsulates components and hinders external entities to interact with those components. The definition of a service-provision relationship and disconnecting these two relationships leads us to a layered structure that does not entail encapsulation, and where entities from a lower level provide service and thus realise, support or can at least be used to justify the existence of entities at higher levels. Such layered architecture styles are, as already mentioned, well-known in the context of EAFs. Take for instance the Zachman Framework with its different layers, where business processes are supported and realized by the underlying software applications, which in turn are supported and realized by the underlying hardware systems (Zachman 1987; Sowa and Zachman 1992), each representing a distinct stakeholder perspective. A layered-architecture style therefore fulfills ***R1***. If fulfills also ***R3***, because a layered style does not prescribe the starting point for alignment, as it was the case with ISP methodologies. A distinction between *intra-layer dependencies* and *inter-layer dependencies* has also been made for meeting ***R2***. Intra-layer dependencies are dependencies between entities that are observable within one layer, whereas inter-layer dependencies are dependency relations between entities in different layers. In this paper our main focus is on inter-layer dependencies. Figure 2 depicts our *Dependency-based Alignment Framework*, where a layered-architecture style was used for constructing it. This construct represents the solution for ***PP1***. The number of conceivable alignment perspectives is $n(n-1)$, where n represents the number of layers. This amounts in our case to 12 different alignment perspectives. The different alignment perspectives are unique, because they follow the concept of causation. This means that a change caused in a dependency relation in one dependency layer may cause also an effect in a dependency relation that is located in another dependency layer. The correlation between the location where a change is caused (indicated by a node) and the location where an effect may occur (indicated by the location of the arrow) explains the difference between alignment perspectives that are embracing the same layers. Accordingly, perspectives 1 and 12, 2 and 9, 3 and 7, 4 and 11, 5 and 8, and 6 and 10 comprise in each case the same layers, but with interchanged locations of causes and effects. This provides an answer to ***KP1***.

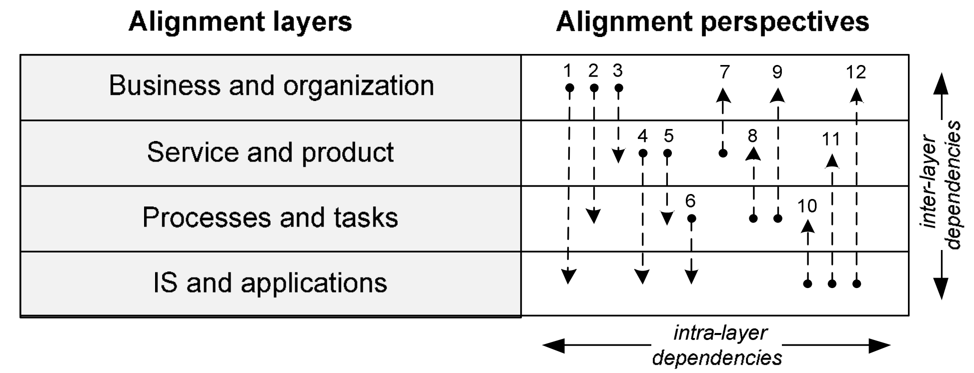

Figure 2. The Dependency-based Alignment Framework

Note that this can be viewed as a refinement of the well-known Strategic Alignment Model by Henderson and Venkatraman (1993), who provided just four alignment perspectives that are driven solely by strategic decisions. The number of layers in our framework is furthermore freely adjustable; layers may be merged or new layers may be added. Consequently, this would change the number of possible alignment perspectives according to the simple formula stated above.

Assessing Dependency Impact During the Life Cycle of Collaborative Networks

Analogical thinking and reasoning has proven to be useful in solving problems for a long time, so that "models in business and management processes often draw on analogs for explanation" (Liang and Konsynski 1993). The concept of life cycles stems from the natural sciences, more precisely from biology. It found many application areas, like for instance in economy, engineering or marketing. Nowadays life cycles are omnipresent in research and practice, where product life cycles, project life cycles, software development life cycles represent only a few examples. Nevertheless, it should be noted that they are all based on the biological life cycle concept that differentiates specific periods or phases – starting from genesis until dissolution – that a species typically goes through. The biological life cycle concept also found its analog application in exploring and explaining the life phases of Collaborative Networks, like suggested among others by Thoben and Jagdev (2001) or Camarinha-Matos and Afsarmanesh (2007).

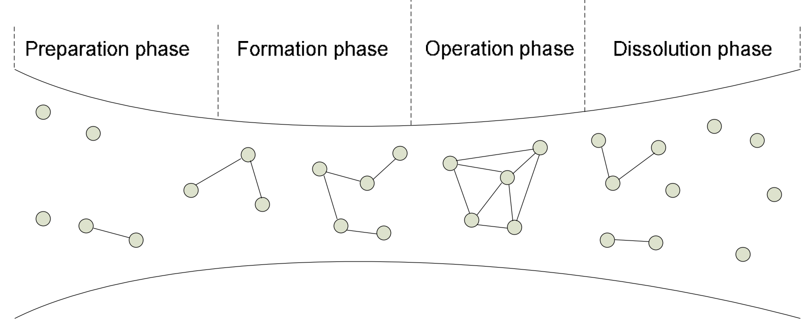

Figure 3. Life Cycle Phases of a Collaborative Network (cf. Thoben and Jagdev 2001)

Thoben and Jagdev (2001) describe the formal duration of collaborations between different companies or organizations by what they call the life cycle of an enterprise network. They distinguish thereby four different life cycle phases: a *preparation* phase (preparing, first sourcing of partners, etc.), a *formation/setting up* phase (final partner selection, legal issues, contracts, etc.), an *operation* phase (day-to-day management of the network), and a *dissolution* phase (decomposition of the network). These phases are shown in the conceptual representation in Figure 3, where nodes represent independent enterprises and edges represent collaborative relationships between the enterprises. Most of the reviewed life cycles in the context of Collaborative Networks consist of the same phases like proposed here. However, Camarinha-Matos and Afsarmanesh (2007) have also defined four life cycle phases like Thoben and Jagdev (2001), but divide the last phase into two distinct phase possibilities, which are either *dissolution* or *metamorphosis*, because not all Collaborative Networks dissolve completely after accomplishing their goal, but changes in membership and common objectives lead to a new network.

The exploration of the life cycle of a Collaborative Network is an important aspect, because different phases require different operational management practices for being able to deal with them properly. In our case this is due to the relevance and impact that the different inter-organizational dependencies and interdepencies hold along the life cycle, because these are not equally relevant. We have mapped the alignment layers – representing the dependency categories – against the above described life cycle phases. We have taken the categorized results from our literature review. On this basis we have assigned where possible the impact (high, medium, low) of each dependency category to the different life cycle phases. As far as the literature did not always give sufficient insight about impacts of dependency categories along the lifetime dimension of Collaborative Networks, we have appointed the missing impact values by forming a focus group of experts in the field of inter-organizational collaboration as shown in Table 3.

Table 3. Dependency Impact Analysis Along the Life Cylce of Collaborative Networks

Alignment Layer	*Preparation phase*	*Formation phase*	*Operation phase*	*Dissolution phase*	*Metamor-phosis phase*
Business and organization	⦿	●	●	○	●
Services and products	⦿	●	●	⦿	●
Processes and tasks	○	⦿	●	○	⦿
IS and applications	○	⦿	●	○	⦿
Legend: ● high impact ⦿ medium impact ○ low impact					

Generally said, inter-organizational dependencies and interdependencies are present throughout all phases, but the degree of relevance differs. During the preparation phase the impact is usually low, except for the layers of business & organization and services & products. Often on the basis of past collaboration activities the first sourcing of partners happens here (Thoben and Jagdev 2001). Knowledge about organizational interdependencies as well as about complementary competencies with respect to service & product dependencies justify the medium valuation. When the common business goal gets more concrete and a network shall be formed, the impact increases. Business constellations are often set up on task-resource dependencies (Zarvić et al. 2010). Then, a high impact of all dependency categories is present during the operation phase. The day-to-day management of the network comprises in-depth knowledge about dependencies in all layers and categories respectively. Applications and IS need to communicate and depend on each other (Scheibenberg and Panda 2008), but also time-based dependencies in workflows (Cox et al. 2001) have a tremendeous impact on dependencies from other layers. Interdependencies and dependencies in the top two layers are established and need to be monitored permanently. The lowest impact can be observed in the dissolution phase, when the network breakes up. We have indicated here a medium impact for the services & products layer, because most business opportunities that lead to a Collaborative Network are driven by dependencies out of this category. Consequently, even a network dissolves, after sales services and warantee for such products and services are still important. In case of a metamorphosis at the end of the life cycle, the valuation is – due to similar activities and objectives – comparable to the one from the formation phase. This answers ***KP2***.

Framework Application

In the following, we will demonstrate the applicability of the Dependency-based Alignment Framework by means of an example case. Therefore we will instantiate several alignment perspectives of our framework. The paradigm of inter-organizational collaboration is omnipresent and encompasses all kinds of inter-organizational relationships. Camarinha-Matos and Afsarmanesh (2009) presented a taxonomy that included networked business manifestations such as Virtual Organizations or traditional Supply Chains. Manufacturing enterprises, for instance, often depend on input from third parties to be able to fabricate complex products that are demanded by the market. There are multiple reasons for such a dependency relation including the trend of reducing in-house production depth (Quinn 1999) or simply switching outsourcing suppliers (Sia et al. 2010a). Also new business opportunities might require competencies from outside. The following example case illustrates such a situation and deals with the formation as well as with the operation phases of a Collaborative Network, because our impact analysis showed that inter-organizational dependencies and interdependencies are highly relevant during these two phases.

Example Case: Modification of Product Structures

A manufacturing company, which we will call NoZa Scooter Productions, is specialized for the last 50 years in the production of nostalgically looking scooters. As far as NoZa does not produce all scooter components on its own, it developed long term partnerships with several suppliers that cater such components also for competing scooter manufacturers. NoZa thereby produces just the motor and the chassis for the retro scooters on its own; all other components are supplied by four different vendors with whom long term collaborations are existing. The urge to expand and not to focus solely on the segment of retro scooters, but also on scooters containing more modern IT components brought NoZa into the situation to consider the development of new products. This decision is based on a recently performed market analysis, which brought following results: the market demands more modern designed scooters, with modern IT, such as for instance an electronic speedometer that incorporates a board computer with visual navigation functionality. In order to achieve a competitive advantage and to respond properly to market demands, NoZa now wishes to analyze effects of such a product that has different properties than the products from the current portfolio. However, this represents a change with respect to current business practices, affecting the inter-organisational dependencies and interdependencies that currently exist. Two main questions can be posed for an initial analysis of the described change. First we need to know how the change in the new product structure influences the structure of the supplier network (perspective 7) and then it needs to be identified what consequences would a change in the supplier network structure have on the supporting structure of IT (perspective 1).

From Product Structure to Collaborative Network Structure

Clearly, the envisioned modification of the product structure is reflected by the bill of material. Thereby all the components needed for the production of the scooter represent also resources. A missing component leads to the fact that the scooter cannot be entirely produced. Thus, there exists a dependency relationship between the resources needed, which we can call in this case resource-resource dependency (cf. Crowston 1994). With respect to manufacturing resource-resource dependencies could be modeled as Gozinto graphs (Vaszonyi 1954), which are often very similar to visualized bill of material trees. However, such Gozinto graphs also allow to reason about production structures, which can on the other side easily be allocated to the different business actors in a Collaborative Network structure. Hence we deal with a change in the service and product dependencies layer and investigate whether this change led to some effect in the business and organisational layer. For managing this change we need to depict both the product and the organizational structures. We do so by making use of the *e3-value* modelling approach, which has been proposed by Gordijn and Akkermans (2001; 2003). Figure 4 shows our *e3-value* model of the new inter-organizational collaboration structure. In *e3-value* a business *actor* – usually a profit and loss responsible company – is represented as a rectangle. NoZa and the different suppliers of scooter components are such business actors. The concept of *market segment* on the other side represents a group of business actors that share the same needs. It is represented as three stacked actors. Connecting these two *e3-value* concepts would already suffice to represent networked collaboration structures. Such a connection is done by *value object* exchanges (e.g. resource for money) between actors and market segments. A value object is anything that is of value to at least one actor, such as money, a service or a product. 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. The exchange of physical value objects represents already gives an impression which resources are needed to manufacture the scooter at NoZa. With the concept of *dependency path* we can also illustrate resource-resource dependencies. Such a dependency path begins with a *start stimulus* – representing some kind of need – and ends with *end points*. AND and OR forks/joins enable the modeler to connect dependency paths or to split a path into several conceivable sub-scenarios.

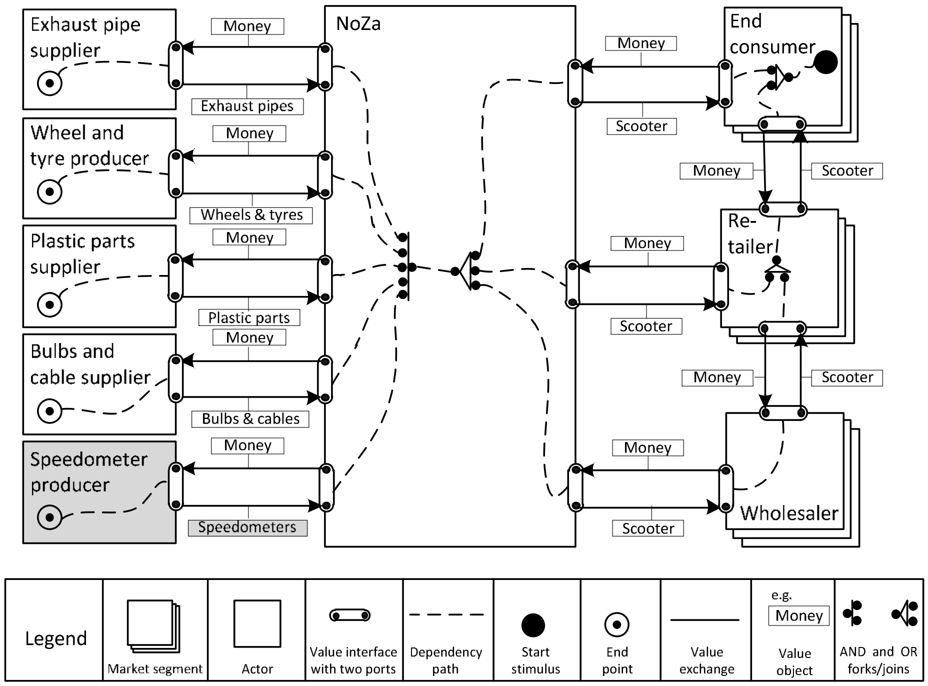

Figure 4. Change of Resource-resource Dependency at Product Level Requires Change in Sequential Interdependency at Organizational Level

As far as we first consider alignment perspective 7 in this example case, we can observe that the modification of the product structure by means of adding the new resource speedometer (colored in gray) brings with it that this resource needs to be supplied by a new vendor, which we named the speedometer producer (colored in gray). This of course changes the current constellation of the sequential interdependency in the overall supply chain. We talk at product level about resource dependencies (Pfeffer and Salancik 1978), because a unidirectional perspective is taken. The final product scooter depends on the resource speedometer, but not vice versa. At organizational level on the other side, we use the term interdependency, which takes here a serial form (Thompson 1962), and where all participating companies are interdependent on each other for achieving the common goal of scooter production. Hence, the used dependency terminology is in line with the original terms taken from literature. For giving a complete picture of this inter-organizational collaboration constellation we have also depicted the market segments retailer and wholesaler. Next to the integration of dependencies and interdependencies belonging to umpteen layers – in this case the business & organizational and the services & products dependency layers - the *e³-value* approach offers the basis for calculating the feasibilty of the new network from a financial point of view. It is based on the concept of economic reciprocity (Gordijn and Akkermans 2001) and offers the possibility of calculating revenues against expenses. This example shows that Business-IT alignment is not limited only to the pure Business and IT domains. It shows that changing the structure of services and/or products might require a change in the organizational structure of the Collaborative Network. Hence, each service and/or product component must be assignable to one participating company, which facilitates it and represents one fact that needs to be considered when defining integrity rules lateron.

From Collaborative Network Structure to ICT systems landscape

The alignment perspective 7 led to a change in the sequential interdependence in our top layer business and organization, because a new partner for speedometer supply needed to be added. However, this change might again cause changes in dependency relations at other levels. NoZa – as the central lead partner in this inter-organizational collaboration structure – represents the central point that specifies the order of the sequential interdependency at hand. This order is shown in Figure 5a and can be supported by means of so-called inter-organizational systems (IOS). This point of view complies with alignment perspective 1. Johnston and Vitale (1988) define an IOS to be "built around information technology, i.e. around computer and communications technology that facilitates the creation, storage, transformation, and transmission of information. An IOS differs from an internal, distributed information system by allowing information to be sent across organisational boundaries". Kumar and van Dissel (1996) provided an IOS typology that is based on Thompson's (1962) interdependencies and hence complies with them.

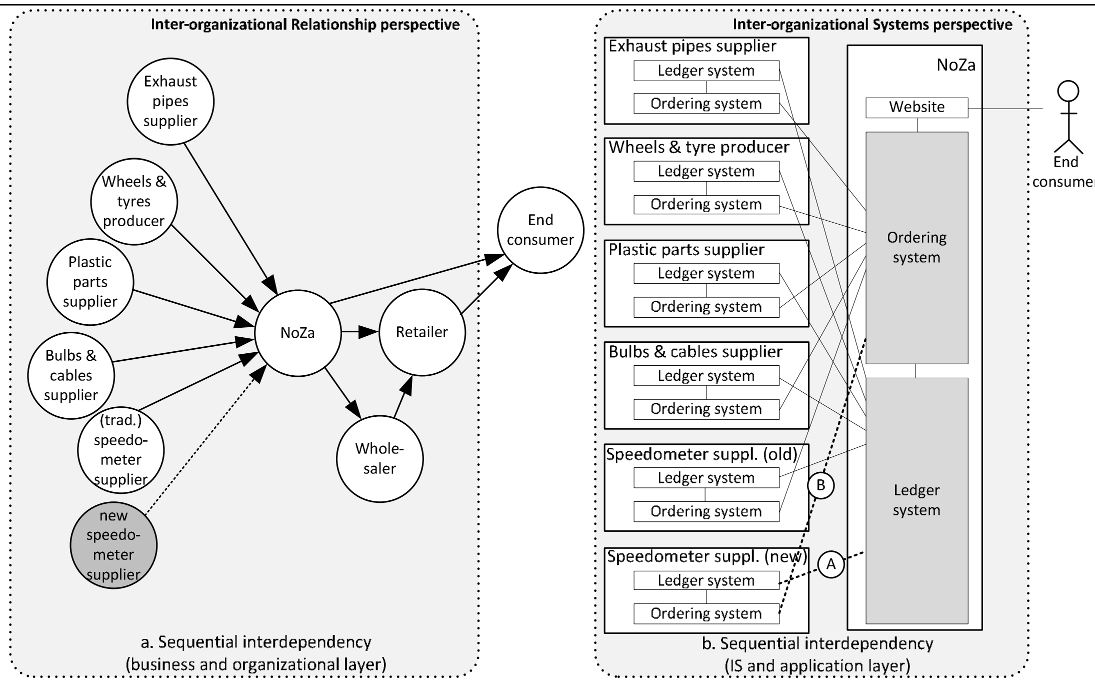

Figure 5. a) Sequential Interdependency at Organizational Level Mapped on b) Application and IS Level

The order of interdependencies at organizational level is shown in Figure 5a. The arrows thereby denotate the sequential order. In Figure 5b on the right a communication diagram depicts the communication channels between the systems located at the different companies. Following applications are currently implemented at each partner: a *ledger system* for financial administration and an *ordering system* for order management. These two systems realize the required communication between the NoZa and the other participating companies. Clearly a communication channel between the ledger systems (A) and the ordering systems (B) is needed in order to integrate the new speedometer supplier into the Collaborative Network structure. Hence, the sequential interdependency between network partners at organizational level is aligned with the IOS landscape, if the communication needs of all ordered organizational interdependencies are supported by means of communication channels between the IS and applications at each company. Communication diagrams represent a useful modeling notation here, because they incorporate most dependeny concepts relevant to alignment perspective 1. This represents also an important insight for solving lateron ***PP2*** and emphasizes on the nested problem solving approach taken.

Implementing the Concepts: Selection of Supporting IT

To apply our framework effectively in the example case described, where a huge amount of interdependencies and dependencies must be handled, we first have to select an appropriate IT-support. A first basic criterion for the selection is that the system provides for capturing arbitrary dependencies and interdependencies in an easy way in order to empower the management to handle and manage these relations. A second criterion is that the system should provide for specifying integrity rules on the basis of the captured dependencies in order to automatically detect problems and inconsistencies which require the management's attention. Regarding the first criterion, Semantic Wikis are in general highly relevant as they are on the one hand easy to use (which distinguishes them from most traditional expert systems) and on the other hand provide for the specification of arbitrary typed semantic relations between categorized wiki pages (cf. e.g. Krötzsch et al. 2007). These relations can be used to express dependencies between concepts represented by wiki pages. Hence they allow representing the inter-organizational dependencies. Further, the category to which a wiki page is assigned can represent the level of the dependency framework. Thus, Semantic Wikis allow for operationalizing the result of ***PP1***. Regarding the second criterion, Semantic Wikis offer rich capabilities for querying their content with the help query languages. One such language is the SPARQL query language, which is standardized by the W3C and supported for example by OntoWiki, Kiwi Wiki or Semantic Media Wiki (via the SPARQL backend). SPARQL supports (amongst other things) logical conjunction, disjunction, negation as failure via the NOT EXISTS construct and provides aggregate functions e.g. for counting. It is thus sufficient to express integrity rules such as "if there exists an entity of a type *x*, then there must be an entity of type *y* connected by a relation *z*". Using such integrity rules for detecting problems caused by modifications supports the task of managing changes and therefore contributes to and operationalizes ***PP2***.

To conclude the selection of supporting IT technology, we choose Semantic Wikis for supporting our approach as they provide a user friendly way to capture dependencies and are moreover able to perform basic integrity checks by using query languages (for an overview of Semantic Wikis see e.g. Buffa 2006 or Panagiotou and Mentzas 2007). Specifically, we choose Semantic Media Wiki (Krötzsch et al. 2006) as it provides for an easy to use, Wikipedia-style interface for specifying dependencies. For example, the fact that a product *Scooter* is assigned to an organizational unit *Production* can easily be stated by inserting the text `[assignedTo::Production]` into the source text of the wiki page (cf. Figure 6) identified by an URI like `http://example.org/wiki/Scooter`.

Such formal statements can be used in SPARQL queries which basically consist of triple patterns in form of *subject predicate object* separated by dots. For example, if we would like to know the organizational unit being assigned to the Scooter product we could simply write the following SPARQL query:

```
PREFIX : <http://example.org/wiki/>
SELECT ?Org WHERE { :Scooter :is_assigned_to ?Org }
```

Due to space limitations, we cannot give a further introduction of SPARQL but the interested reader is referred to Prud'hommeaux and Seaborne (2008). Note also that in the remainder we use the default namespace ":" in our examples and omit the prefix declaration.

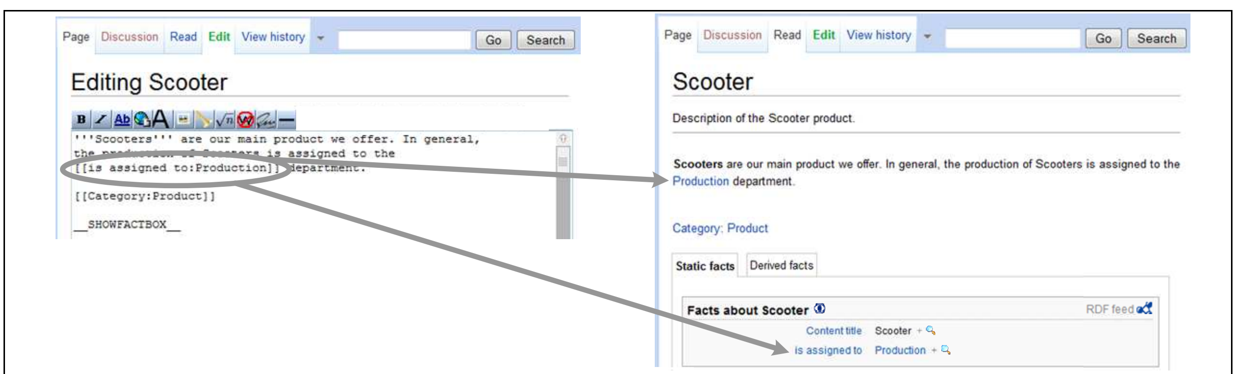

Figure 6. Editing Dependencies (Left) and Displaying Them (Right) by Using Semantic Media Wiki

Specification of Integrity Rules

In this section, we show how to specify integrity rules that target the alignment criteria between dependencies from the different layers of the framework. The framework supports defining integrity rules in a systematic way by examining each alignment perspective between the different layers. We formalize the integrity rules using SPARQL queries (with natural language descriptions) as shown in Table 4. Such queries can be executed e.g. against the backend of Semantic Media Wiki providing SPARQL support.

Table 4. Integrity Rules in Form of SPARQL queries

Integrity Rules for Perspectives 1 to 6	*Integrity Rules for Perspectives 7 to 12*

1. Each organization(al unit) must be supported by at least one ICT system.
`SELECT ?Org`
`WHERE { ?Org a :OrganizationalUnit .`
`    NOT EXISTS { ?Org :isSupportedBy ?x }.`
`    ?x a :ICTSystem .`
`}`
	7. Each product or service component must be assigned to one responsible organization(al unit).
	`SELECT ?ProdOrSrv`
	`WHERE {{ ?ProdOrSrv a :Product } UNION {`
	`         ?ProdOrSrv a :Service }`
	`         ?Org a :Organization`
	`    NOT EXISTS {`
	`      ?ProdOrSrv :isAssignedTo ?Org }`
	`}`

2. Each organization(al unit) must be assigned to at least one task or process.
`SELECT ?Org`
`WHERE { ?Org a :OrganizationalUnit .`
`    NOT EXISTS { ?Org :isAssignedTo ?x }`
`     { ?x a :Task } UNION {`
`       ?x a :Process}`
`}`
	8. Each process or task must produce at least one service or product (sold internally or externally to the organization).
	`SELECT ?ProcOrTask`
	`WHERE {{ ?ProcOrTask a :Process } UNION {`
	`         ?ProcOrTask a :Task }`
	`    NOT EXISTS {`
	`       ?ProcOrTask :produces ?ProdOrSrv }`
	`    {  ?ProdOrSrv a :Product } UNION {`
	`       ?ProdOrSrv a :Service }`
	`}`

3. Each organization(al unit) must be responsible for at least one service or product (being sold internally or externally to the organization).
`SELECT ?Org`
`WHERE { ?Org a :OrganizationalUnit .`
`    NOT EXISTS {`
`        ?Org :isResponsibleFor ?x }`
`    {   ?x a :Service } UNION {`
`        ?x a :Product }`
`}`
	9. Each process or task must be executed by an organization(al unit).
	`SELECT ?ProcOrTask`
	`WHERE {{ ?ProcOrTask a :Process } UNION {`
	`         ?ProcOrTask a :Task }`
	`         ?Org a :OrganizationalUnit`
	`    NOT EXISTS {`
	`         ?ProcOrTask :hasAgent ?Org }`
	`}`

4. Each product or service must have an identifier of the product database being a number greater than zero.
`SELECT ?ProdOrSrv`
`WHERE {{ ?ProdOrSrv a :Product } UNION {`
`         ?ProdOrSrv a :Service }`
`         ?ProdOrSrv :hasDbId ?Id`
`       FILTER (?Id <= 0)`
`}`
	10. Each ICT-system must support at least one task or process (applies also to operating systems that indirectly support tasks via application software running on them).
	`SELECT ?IctSys`
	`WHERE { ?IctSys a :ICTSystem`
	`    NOT EXISTS { ?IctSys :supports ?x }`
	`    { ?x a :Task } UNION { ?x a :Process}`
	`}`

5. Each product or service must be produced by at least one process.
`SELECT ?ProdOrSrv`
`WHERE {{ ?ProdOrSrv a :Product } UNION {`
`         ?ProdOrSrv a :Service }`
`    NOT EXISTS {`
`         ?ProdOrSrv :producedBy ?Process }`
`    ?Process a :Process`
`}`
	11. Each Ordering System must support the ordering of at least one product or service (part)
	`SELECT ?OrdSys`
	`WHERE {{ ?ProdOrSrv a :Product } UNION {`
	`         ?ProdOrSrv a :Service }`
	`    ?OrdSys a :OrderingSystem .`
	`    NOT EXISTS {`
	`        ?OrdSys :supportsOrdering ?ProdOrSrv }`
	`}`

6. Each process or task must be supported by at least one ICT-system.
`SELECT ?ProcOrTask`
`WHERE {{?ProcOrTask a :Process } UNION{`
`        ?ProcOrTask a :Task }`
`    NOT EXISTS {`
`        ?ProcOrTask :assignedTo ?ICTSystem }`
`    ?ICTSystem a :ICTSystem`
`}`
	12. Each ICT-system must be assigned to one responsible organization(al unit).
	`SELECT ?IctSys`
	`WHERE { ?IctSys a :ICTSystem`
	`    NOT EXISTS {`
	`       ?IctSys  :assignedTo ?Org }`
	`    ?Org a :OrganizationalUnit`
	`}`

The queries presented in Table 4 can be executed externally from a semantic wiki by exporting the contents of the wiki into RDF or OWL, a format that is supported widely by semantic wiki engines. Another way would be to use a SPARQL backend (e.g. provided by Semantic Media Wiki), to execute the queries directly in the Wiki (e.g. provided by OntoWiki) or to embed them into wiki pages used for displaying the integrity issues (e.g. provided by Kiwi Wiki). In the following section, we show how these queries help to detect and resolve the integrity issues that occur in our case study.

Iterative Integrity Checking and Conflict Resolution

In the following, a complete example is provided how the Dependency-based Alignment Framework is used in our example case to check (inter)dependencies and to detect integrity rule violations. First, NoZa decides to produce a new scooter type. Therefore, a new product is created in the wiki and added to the category “Product”. As the new type of scooter requires a new speedometer with navigation functionality, this part is also created in the wiki as a separate page.

Having created the page for the new speedometer part, the query 7 retrieves this page as a result as each product or product part must be assigned to a responsible organizational unit, which initially is not the case. Unfortunately, the needed product part cannot be manufactured in-house, so a new external manufacturer has to be created as a wiki page. It is then linked to the page describing the speedometer part as being the responsible organizational unit for that part. After the new organizational unit is created, the queries 1-3 retrieve this organizational unit as several integrity rules are violated.

According to query 1 each organizational unit must be supported by at least one ICT system. The user solves this issue by creating a link from the newly introduced external organizational unit to one of the (internal) IS, e.g. the ordering system where the external manufacturer retrieves current orders from the scooter producer. Then, each organizational unit must be assigned to at least one task or process (query 2). To solve this issue and to remove the external manufacturer from the query result set, the user simply assigns the external manufacturer to the scooter production process (i.e. creates a link between the two wiki pages describing the external manufacturer and the manufacturing process). Additionally, each organizational unit must be responsible for at least one service or product (query 3). To solve this issue, the user adds the external manufacturer to the set of organizational units which are responsible for producing the new scooter type. Note that in Semantic Wikis that support advanced reasoning capabilities, the relation can also be created automatically by using an inference engine inferring the responsibility of the external manufacturer based on the fact that he delivers the speedometer part of the scooter. After the user has solved all integrity rule violations, the change in the Collaborative Network should be aligned. Summing up, this section provided a proof-of-concept and a detailed solution for ***PP2***.

Evaluation

The nested problem structure contains two design problems (PPs) that have been tackled. The research described in this paper can therefore be characterized as an IS design science approach that focused on the creation of a new framework for performing alignment in Collaborative Networks. Evaluation is a crucial and simultaneously difficult component of the research process in design science. Hevner et al. (2004) describe and classify several evaluation methods to be used in this context, which are (i) observational, (ii) analytical, (iii) experimental, (iv) testing, and (v) descriptive. Frank discusses these approaches and claims that most of them test design-oriented research results against reality, which in turn corresponds to the behaviourism stream in IS Research (Frank 2006). For the context of research practices, there exists the problem that innovative artifacts need enough time to get accepted in practice as well as the need for developing future worlds. This in turn means that evaluation of research results is in most cases left to practice. However, we are of the opinion that research results need not only undergo a practical evaluation, but also need to be evaluated in an academic context during early development stages. Here the descriptive design evaluation method (v) that was suggested by Hevner et al. (2004) seems appropriate and is also in line with the proposal by Frank to evaluate the results of design-oriented research in an argumentative and descriptive way by looking at three key postulates of scientific research. These postulates are *abstraction*, *originality* and *justification*. Such an evaluation approach has already been accepted by the IS research community, like can be seen for instance in the work by Vanderhaeghen et al. (2010). In the following we also follow such an evaluation approach.

Abstraction: We started our research with a literature review and analysis of relevant work in the fields of Business-IT Alignment and inter-organizational dependencies to be able to generalize from this knowledge. The common features that have been identified flew into the definition of the three core requirements ***R1***, ***R2***, and ***R3***, which have fully been implemented during the deductive construction of the central design artifact, namely our alignment framework. This result represents a generalized structure that abstracts the complexity that is present in the real world in order to make it manageable in a better way.

Originality: Our proposed Dependency-based Alignment Framework has to be seen as inventive, because it uses and structures existing knowledge in an integrative way. The usage of frameworks is per se not necessarily novel, but the application to different scenarios or fields, like the one described in this article, can be characterized to be novel.

Justification: Our justification perspective includes a pragmatic view on the results presented in this paper. The methodologically sound research approach presented in this paper followed the generally accepted engineering cycle phases (cf. Wieringa and Heerkens 2006) problem investigation, solution design and solution validation/evaluation. We argue that the framework including a possible IT support option seems – in comparison to other approaches – advantageous for Collaborative Networks.

However, we also want to point to possible limitations of our work, in order to provide a complete and objective evaluation. Here we want to discuss especially the aspects of *pluralism* and *dynamics*.

Pluralism: There exist several issues that cannot be fully eliminated, even when the researchers are following the abstraction postulate (see above). An approach, aiming to simplify by abstraction and formalization, often runs the risk of a terminology clash. This means that sometimes one single term might refer to different things, so that one researcher would for instance place a specific term into one layer of our framework, whereas some other researcher would place it in another layer. Reasons for such kind of pluralism might be that the cited researchers stem from different research fields or cultures. In our work we tried to eliminate as much pluralism as possible.

Dynamics: Process flows are usually changing permanently and therefore need to be adapted constantly. This means that as a consequence the dependencies might change at process level and might therefore also have an impact on dependencies and interdependencies situated in other layers of our framework, which in turn means that processes and tasks need to be monitored constantly. Such changes need to be maintained in a coherent and consistent way in the Semantic Media Wiki to ensure fully operational IT support.

Summary and Conclusion

Business-IT Alignment is still an actual topic. In this paper we proposed a framework for performing alignment in Collaborative Networks. Our framework integrates inter-organizational dependencies and interdependencies by using a layered architecture style. The top-level research question that was broken down in multiple subproblems has been answered fully, because we have provided answers for the KPs and solutions for the PPs.

Future work will imply a further elaboration on the intra- and inter-layer dependencies, which will be accompanied by the definition of crisp integrity rules. It should be noted that the presented framework supports the alignment process by indicating in which layer changes of the dependency constellation are likely to occur and have an impact. With respect to the example in this paper, we identified e.g. that a partner is needed, which in turn means that the Collaborative Network structure needs to be aligned to the new product structure, but we did not identify whom exactly to choose. For the activity of partner selection specific methods have been developed.

References

Aggarwal, H. 2009. "Contemporary Research Issues in Business-IT Alignment," *Global Journal of Enterprise Information Systems* (1:1), pp. 101-109.

Ball-Rokeach, S. J. 1985. "The origins of individual media-system dependency - A sociological

framework," *Communication Research* (12:4), pp. 485-510.
Bartenschlager, J., and Goeken, M. 2009. "Designing Artifacts of IT Strategy for Achieving Business/IT Alignment," in *Proceedings of the Americas Conference on Information Systems (AMCIS 2009)*.
Blankenhorn, H., and Thamm, J. E. 2008. "Business-IT Alignment - Aufbau und Operationalisierung der IT-Strategie," *Information Management & Consulting* (23:1), pp. 9-16.
Buffa, M., Crova, G., Gandon, F., Lecompte, C., and Passeron, J. 2006. "SweetWiki : Semantic Web Enabled Technologies in Wiki," in *Proceedings of the First Workshop on Semantic Wikis - From Wiki to Semantics (SemWiki2006), in Conjunction with the 3rd European Semantic Web Conference (ESWC 2006),* June 11-14, Budva, Montenegro, pp. 74-88.
Camarinha-Matos, L., and Afsarmanesh, H. 2004. "The emerging discipline of collaborative networks," in *Proceedings of the 5th Working Conference on Virtual Enterprises*, August 22-27, Toulouse, France, pp. 3-16.
Camarinha-Matos, L., and Afsarmanesh, H. 2005. "Collaborative Networks: a new scientific discipline," *Journal of Intelligent Manufacturing* (16:4-5), pp. 439-452.
Camarinha-Matos, L., and Afsarmanesh, H. 2007. "A comprehensive modeling framework for collaborative networked organizations," *Journal of Intelligent Manufacturing* (18:5), pp. 529-542.
Camarinha-Matos, L., Afsarmanesh, H., Galeano, N., and Molina, A. 2009. "Collaborative networked organizations – Concepts and practice in manufacturing enterprises," *Computers & Industrial Engineering* (57:1), pp. 46-60.
Campbell, B., Kay, R., and Avison, D. 2005. "Strategic Alignment: a practitioner's perspective," *Journal of Enterprise Information Management* (18:6), pp. 653-664.
Chan, Y. 2001. "Information Systems Strategy, Structure and Alignment," in *Strategic Information Technology: Opportunities for Competitive Advantage*, P. Raymond (ed.), PA: Idea Group Publishing, pp. 56-81.
Chan, Y., and Reich, B. 2007. "IT alignment - What have we learned?," *Information Technology* (22:4), pp. 297-315.
Cox, L., Delugach, H. S., and Skipper, D. 2001. "Dependency Analysis Using Conceptual Graphs," in *Proceedings of the 9th International Conference on Conceptual Structures, ICCS 2001,* July 30-August 3, Stanford, CA, USA, pp. 117-130.
Crowston, K. 1994. *A taxonomy of organizational dependencies and coordination mechanisms*, MIT Sloan School of Management, TR #174, Cambridge, Massachusetts.
de Padua Alburquerque Oliveira, A., and Cysneiros, L. M. 2006. "Defining Strategic Dependency Situations in Requirements Elicitation," in *Proceedings of WER 2006 - Workshop em Engenharia der Requisitos*, July 13-14, Rio de Janeiro, Brazil, pp. 12-23.
Emerson, R. M. 1962. "Power-Dependence Relations," *American Sociological Review* (27:1), pp. 31-41.
Finkelstein, C. 1989. An Introduction to Information Engineering - From Strategic Planning to Information Systems, Sydney: Addison-Wesley.
Frank, U. 2006. *Towards a Pluralistic Conception of Research Methods in Information Systems,* Report No. 7, Universität Duisburg-Essen.
Gordijin, J., and Akkermans, H. 2001. "Designing and Evaluating E-Business Models," *IEEE Intelligent Systems* (16:4), pp. 11-17.
Gordijin, J., and Akkermans, H. 2003. "Value-based requirements engineering - exploring innovative e-commerce ideas," *Requirements Engineering* (8:2), pp. 114-134.
Henderson, J. C., and Venkatraman, N. 1993. "Strategic Alignment: Leveraging Information Technology for Transforming Organisations," *IBM Systems Journal* (32:1), pp. 272-284.
Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research", *MIS Quarterly (*28:1), pp. 75-105.
Holmström, J., Ketokivi, M., and Hameri, A.-P. 2009. "Bridging Practice and Theory: A Design Science Approach," *Decision Sciences* (40:1), pp. 65-87.
Johnston, H., and Vitale, M. R. 1988. "Creating Competitive Advantage with Interorganizational Information Systems," *MIS Quarterly* (12:2), pp. 153-165.
Jordan, E. and Tricker, B. 1995. "Information strategy: alignment with organization structure," *Journal of Strategic Information Systems* (4:4), pp. 357-382.
Kanter, R. 1994. "Collaborative advantage - The art of alliances," *Harvard Business Review* (72:4), pp. 96-108.
Krötzsch, M., Schaffert, S., and Vrandecic, D. 2007. "Reasoning in Semantic Wikis," in *Reasoning Web: Third International Summer School 2007,* Dresden, Germany, September 3-7, 2007, Tutorial

Lectures (Lecture Notes in Computer Science) Berlin: Springer, pp. 310-329.
Krötzsch, M., Vrandecic, D., and Völkel, M. 2006. "Semantic MediaWiki," in *Proceedings of the 5th International SemanticWeb Conference (ISWC 2006),* November 5-9, Athens, GA, USA, pp. 935-942.
Kuechler, B., Vaishnavi, V. 2011. "Promoting Relevance in IS Research: An Informing System for Design Science Research," *Informing Science: the International Journal of an Emerging Transdiscipline* (14:1), pp. 125-138.
Kumar, K., and van Dissel, H. G. 1996. „Sustainable collaboration - managing conflict and cooperation in interorganizational systems," *MIS Quarterly* (20:3), pp. 297-300.
Lankhorst, M., et al. 2009. *Enterprise Architecture at Work - Modelling, Communication and Analysis*, 2nd edition, Dordrecht: Springer.
Lauesen, S. 2002. *Software Requirements: Styles and Techniques*, Harlow, England: Addison Wesley.
Lederer, A., and Mendelow, A. 1989. "Coordination of Information Systems Plans with Business Plans," *Journal of Management Information Systems* (6:2), pp. 5-19.
Liang, T.-P., and Konsynski, B. R. 1993. "Modeling by analogy – Use of analogical reasoning in model management systems," *Decision Support Systems* (9:1), pp. 113-125.
Ludwig, A., and Franczyk, B. 2008. "COSMA - An Approach for Managing SLAs in Composite Services," in *ICSOC '08 Proceedings of the 6th International Conference on Service-Oriented Computing,* December 1-5, Sydney, Australia, pp. 626-632.
Luftman, J. N., and Ben-Zvi, T. 2010. "Key Issues for IT Executives 2010: Judicious IT Investments Continue Post Recession," *MIS Quarterly Executive* (9:4), pp. 263-273.
Mahoney, J., and Kitzis, E. 2009, "Integrating the Transformatin of Business and IT," Gartner research report, 23 pages.
Malone, T. W., and Crowston, K. 1994. "The interdisciplinary study of coordination," *ACM Computing Surveys* (26:1), pp. 87-119.
Martin, J. 1982. *Strategic Data Planning Methodologies,* Englewood Cliffs, NJ: Prentice Hall.
Martin, J. 1989. *Information Engineering*. Englewood Cliffs, NJ: Prentice Hall.
Martin, S. F., Wagner, H.-T., and Biemborn, D. 2008. "Process Documentation, Operational Alignment, and Flexibility in IT Outsourcing Relationships: A Knowledge-Based Perspective," in *Proceedings of the 29th International Conference on Information Systems*, Paris, 2008.
Newell, A., and Simon, H. A. 1972. *Human problem solving*, Englewood Cliffs, NJ: Prentice-Hall.
Panagiotou, D., and Mentzas, G. 2007. "A comparison of Semantic Wiki Engines," *s*lides presented at *the 22nd European Conference on Operational Research,* July 2007, Prague, Czech Republic.
Pfeffer, J., and Salancik, G. R. 1978. The external control of organizations - A resource dependence perspective, New York: Harper & Row.
Prud'hommeaux, E., and Seaborne, A. 2008. *SPARQL Query Language for RDF,* W3C Recommendation, January 15, 2008, W3C.
Quinn, J. B. 1999. "Strategic Outsourcing - leveraging knowledge capabilities," *MIT Sloan Management Review (*40:4), pp. 9-21.
Rinaldi, S. M., Peerenboom, J. P., and Kelly, T. K. 2001. "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies," *IEEE Control Systems Magazine* (21:6), pp. 11-25.
Scheibenberg, K., and Pansa, I. 2008. "Modelling dependencies of IT Infrastructure elements," in *Proceedings of the 3rd International Workshop on Business-driven IT Management (BDIM 2008),* April 7, Salvador, Brazil, pp. 112-113.
Schlitt, M. 2003. *Fundamentals and methods of interpreting and constructing information system models*, PhD Thesis (in German), Bamberg: University of Bamberg Press.
Schlosser, F., and Wagner, H.-T. 2011. "IT Governance Practices for Improving Strategic and Operational Business-IT Alignment," in *Proceedings of 15th Pacific Asia Conference on Information Systems*, Brisbane, Australia.
Shostack, L. G. 1993. "How to design a service," *European Journal of Marketing* (16:1), pp. 49-63.
Sia, S. K., Lim, W. K., and Kanapaty, P. P. 2010a. "Switching IT Outsourcing Suppliers: Enhancing Transition Readiness," *MIS Quarterly Executive* (9:1), pp. 23-33.
Sia, S. K., Soh, C., and Weill, P. 2010b. "Global IT Management – Structuring for Scale, Responsiveness, and Innovation," *Communications of the ACM* (53:3), pp. 59-64.
Sowa, J. F., and Zachman, J. A. 1992. "Extending and formalizing the framework for information systems architecture," *IBM Systems Journal* (31:3), pp. 590-616.
Tagg, R., and Freyberg, C. 1997. *Designing Distributed And Cooperative Information Systems*. International Thomson Computer Press.

Tapscott, D., Ticoll, D., and Lowy, A. 2000. *Digital Capital – Harnessing the power of business webs*. Boston: Harvard Business School Press.
Thoben, K. D., and Jagdev, S. 2001. "Typological issues in enterprise networks," *Production Planinng & Control* (12:5), pp. 421-436.
Thompson, J. 1962. *Organizations in action*, New York: McGraw-Hill.
Tiwana, A., and Konsynski, B. 2010. "Complementarities Between Organizational IT Architecture and Governance Structures," *Information Systems Research* (21:2), pp. 288-304.
Trienekens, J., Hvolby, H.-H., Steger-Jensen, K., and Falster, P. 2008."Architectural Frameworks for Business Information System Analysis and Design," in *Lean Business Systems and Beyond*, T. Koch (ed.), pp. 413-421.
van Eck, P., Blanken, H., and Wieringa, R. 2004. "Project GRAAL: Towards Operational Architecture Alignment," *International Journal of Cooperative Information Systems* (13:3), pp. 235-255.
Vanderhaeghen, D., Fettke, P., and Loos, P. 2010. "Organizational and Technological Options for Business Process Management from the Perspective of Web 2.0," *Business & Information Systems Engineering* (2:1), pp. 15-28.
Vazsonyi, A. 1954. "The Use of Mathematics in Production and Inventory Control," *Management Science* (1:1), pp. 70-85.
vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., and Cleven A. 2009. „Reconstructing the Giant: On the importance of rigour in documenting the literature search process," in *Proceedings of the 17th European Conference on Information Systems*.
Webster, J., and Watson, R. 2002. "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *MIS Quarterly* (26:2), pp. xiii-xxiii.
Wieringa, R. 2003. *Design Methods for Reactive Systems*, San Francisco: Morgan Kaufmann.
Wieringa, R., Blanken, H., Fokkinga, M., and Grefen, P. 2003. „Aligning Application Architecture to the Business Context," in *Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003)*, June 16-18, Klagenfurt, Austria, pp. 209-225.
Wieringa, R. 2005. "Requirements researchers: are we really doing research?," *Requirements Engineering* (10:4), pp. 304-306.
Wieringa, R., and Heerkens, J. 2006. "The methodological soundness of requirements engineering papers: a conceptual framework and two case studies," *Requirements Engineering* (11:4), pp. 295-307.
Wieringa, R. 2008. "Operational Business-IT Alignment in Value Webs," in *Information Systems and e-business technologies,* R. Kaschek, C. Kop, C. Steinberger, G. Fliedl (eds.), pp. 371-378.
Wieringa, R. 2009. "Design Science as Nested Problem Solving," in Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2009), May 6-8, Philadelphia, PA, USA, pp. 1-12.
Wieringa, R. 2010. "Relevance and Problem Choice in Design Science," in *Proceedings of the 5th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2010)*, June 4-5, St. Gallen, Switzerland, pp. 61-76.
Winkler, M., and Schill, A. 2009. "Towards dependency management in service compositions," in *ICE-B 2009 Proceedings of the International Conference on e-Business*, July 7-10, Milan, Italy, pp. 79-84.
Yu, E., and Mylopoulos, J. 1993. "An Actor Dependency Model of Organizational Work - With Application to Business Process Reengineering," in *Proceedings of the Conference on Organizational Computing Systems (COOCS 93)*, November 1-4, California, US, pp. 258-268.
Zachman, J. A. 1982. "Business Systems Planning and Business Information Control Study - A comparison," *IBM Systems Journal* (21:1), pp. 31-53.
Zachman, J. A. 1987. "A Framework for Information Systems Architecture," *IBM Systems Journal* (26:3), pp. 276-292.
Zarvić, N., and Wieringa, R. 2006. "An Integrated Enterprise Architecture Framework for Business-IT Alignment," in *Proceedings of the CAiSE'06 Workshop on Business/IT Alignment and Interoperability (BUSITAL'06),* June 5-9, Luxembourg, pp. 262-270.
Zarvić, N., Daneva, M., and Wieringa, R. 2007. "Value-Based Requirements Engineering for Value Webs," in *Proceedings of 13th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'07)*, June 11-12, Trondheim, Norway, pp. 116-128.
Zarvić, N., Seifert, M., and Thoben, K.-D. 2010. "A task-resource dependency perspective on partner selection during the formation of networked business constellations," *International Journal of Networking and Virtual Organisations* (7:5), pp. 399-414.