Boundary Interactions and Motors of Change in Requirements Elicitation: A Dynamic Perspective on Knowledge Sharing

Christoph Rosenkranz
University of Cologne, Goethe University, rosenkranz@wiso.uni-koeln.de

Helena Vranešić
Windhoff Software Services GmbH, Goethe University, h.vranesic@wind-soft.de

Roland Holten
Goethe University, holten@wiwi.uni-frankfurt.de

Follow this and additional works at: https://aisel.aisnet.org/jais

Boundary Interactions and Motors of Change in Requirements Elicitation: A Dynamic Perspective on Knowledge Sharing

Christoph Rosenkranz
University of Cologne, Goethe University
rosenkranz@wiso.uni-koeln.de

Helena Vranešić
Windhoff Software Services GmbH, Goethe University
h.vranesic@wind-soft.de

Roland Holten
Goethe University
holten@wiwi.uni-frankfurt.de

Abstract

The building of shared understanding between project stakeholders in the requirements elicitation phase is necessary for knowledge sharing and a key factor for successful information systems (IS) development. However, the processes that lead to shared understanding and successful knowledge sharing are still not well understood. We examine how stakeholders interact and use boundary objects during requirements elicitation in data warehouse development projects. We draw on Carlile's (2004) framework for managing knowledge across boundaries and introduce the concept of brokering situations. Using the concept of brokering situations, we examine how shared understanding develops and knowledge is shared through the interplay of brokers, their individual knowledge, and boundary objects as well as through the alignment of project participants' situation models. We contribute to the literature on knowledge sharing and requirements elicitation in three ways: by introducing the concept of brokering situations; by developing a theoretical framework – the boundary interaction framework – that provides an analytical perspective on the dynamics of knowledge sharing in requirements elicitation; and by applying the framework to show that both goal-driven (teleological) and conflict-driven (dialectical) motors of change explain process progress and the changes of brokers as well as boundary objects during the building of shared understanding.

Keywords*: Information Systems Development, Requirements Elicitation, Boundary Spanning, Boundary Objects, Brokering Situation, Boundary Interaction Framework, Motors of Change.*

Boundary Interactions and Motors of Change in Requirements Elicitation: A Dynamic Perspective on Knowledge Sharing

1. Introduction

Building a shared understanding between stakeholders has been identified repeatedly as a key determinant of successful information systems (IS) development (Hirschheim & Klein, 1989; Siau, Long, & Ling, 2010; Tan, 1994; Tiwana & McLean, 2003). Shared understanding is the degree of cognitive overlap and commonality in beliefs, expectations, and perceptions about a given target (Cohen & Gibson, 2003, p. 8). Its purpose in IS development is to allow all stakeholders to determine a precise representation of the knowledge domains needed for the design of an information technology (IT) system (Tan, 1994, p. 160). Since the required pieces of knowledge may reside with different stakeholders (Cabrera & Cabrera, 2002, p. 691), knowledge sharing—the process through which knowledge is exchanged among stakeholders (Argote, McEvily, & Reagans, 2003; Argote & Miron-Spektor, 2011)—becomes a prerequisite for IS development.

Successfully building shared understanding and sharing knowledge is highly contingent on the effectiveness of the communication between stakeholders during IS development (Bostrom, 1989; Gallivan & Keil, 2003; Guinan & Bostrom, 1986; He, Butler, & King, 2007). Specifically, the processes of communicating, discussing, negotiating, analyzing, specifying, and validating requirements play a fundamental role (Abran, Moore, Bourque, & Dupuis, 2004; Alvarez & Urla, 2002; Ambriola & Gervasi, 2006; Cronan & Means, 1984; Gallivan & Keil, 2003; Levina, 2005). This makes requirements elicitation one of the most important and most complex phases of the IS development process (Chakraborty, Sarker, & Sarker, 2010, Mathiassen, Tuunanen, Saarinen, & Rossie, 2007).

However, the literature is replete with examples of users' inability to specify requirements accurately, and developers frequently are criticized for being unable to elicit requirements from users (Davis, Dieste, Hickey, & Juristo, 2006; Davis, 1982; Hansen & Lyytinen, 2010). IS development often fails not due to technical problems, but because IT systems correspond poorly to business users' needs and requirements (Curtis, Krasner, & Iscoe, 1988; Hofmann & Lehner, 2001). Although many research results have found their way into practice (Hansen, Berente, & Lyytinen, 2008), requirements elicitation remains a challenge because of fundamental changes in the requirements engineering environment in recent decades (Hansen & Lyytinen, 2009; Jarke, Loucopoulos, Lyytinen, Mylopoulos, & Robinson, 2010; Jarke & Lyytinen, 2010; Jarke et al., 2009). This has led to calls for "a new systematic science of design requirements engineering…with systematic processes and strict empirical grounding" (Jarke & Lyytinen, 2010, p. 123).

In this paper, we contribute to this grounding with an empirical study of stakeholder interactions. We advance the literature on knowledge sharing and requirements elicitation by developing a theoretical framework that provides an analytical perspective on the dynamics of knowledge sharing. We show that our concepts allow for better explanations of the observed processes during requirements elicitation. We concentrate on *brokering*, an important part of the social interactions that occur during stakeholder communication in requirements elicitation (Fleming & Waguespack, 2007; Pawlowski & Robey, 2004; Pawlowski, Robey, & Raven, 2000). We frame our study with two key concepts related to brokering: brokers (Wenger, 1998), and boundary objects (Star & Griesemer, 1989). The interplay of brokers and the boundary objects they use during brokering has not been investigated in detail (Kimble, Grenier, & Goglio-Primard, 2010) despite the recognition that IS development has mixed outcomes arising from a variety of technical, social, and political problems in working across knowledge boundaries (Doolin & McLeod, 2012; Hussain & Cornelius, 2009; Pee, Kankanhalli, & Kim, 2010). Similarly, the appropriate choice of boundary objects and project participants acting as brokers in IS development projects has not been considered in the previous literature (Gopal & Gosain, 2009).

We make three major contributions. First, we use the literature on knowledge sharing and brokering to develop our own conceptual framework. We develop a novel concept that we call "brokering situation", which is based on the concepts of brokers and boundary objects that brokers use on the knowledge boundaries between communities. Second, we extend the concept of brokering situations to the "boundary interaction framework" and thereby provide a dynamic perspective on knowledge sharing. We develop an improved understanding of the requirements elicitation process by empirically

exploring the interplay between brokers and the boundary objects they use. Specifically, we argue that the changes of either brokers or boundary objects drive the requirements elicitation process. The brokering situations and different "motors of change" are integral components of this dynamic perspective. Third, we show that the boundary interaction framework allows for a better explanation of the observed state of affairs during the building of shared understanding and knowledge sharing than using only one or the other of the two concepts of brokers or boundary objects.

Accordingly, Section 2 discusses challenges for requirements elicitation during IS development. We introduce the concepts of knowledge boundaries, boundary spanners, brokers, and boundary objects, and we discuss their relationship to the challenges. We frame our research in these concepts and establish our understanding of brokering situations. In Sections 3 and 4, we present results from a study of data warehouse development projects, before developing an explanation of motors of change that drive the building of shared understanding and the sharing of knowledge in requirements elicitation in Section 5. This leads to the boundary interaction framework. After discussing our findings in Section 6, we conclude in Section 7 with an indication of limitations and an outlook on further research.

2. Related Work and Theoretical Framework

2.1. Shared Understanding and Knowledge Sharing

Key challenges for successful knowledge sharing during requirements elicitation are stakeholders' "limits of individual cognition" (Hansen & Lyytinen, 2010, p. 3). Users are unable to articulate their needs concisely due to differing perspectives of users and IT professionals (Hansen & Lyytinen, 2010) or differences in internalized "frames" of experiences between IT professionals and users (Davidson, 2002; McMaster & Grinder, 1980). This leads to many of the problems associated with IS development (Boland, 1979; Bostrom & Heinen, 1977). Traditionally, it is up to IT professionals to have the communication competence to establish rapport with users (Guinan & Bostrom, 1986). However, while good communication abilities, technical expertise, and analytical proficiency are important individual skills for IT professionals (Chakraborty et al., 2010; Fisk, Berente, & Lyytinen, 2010; Tan, 1994; White & Leifer, 1986), their lack alone does not sufficiently explain the difficulties of knowledge sharing in requirements elicitation. Since knowledge is "localized, embedded and invested" in practice (Carlile, 2002, p. 445), the differences in the frames of references between stakeholders belonging to different communities of practice with specialized, domain-specific knowledge result in *knowledge boundaries* that separate organizational subunits and communities of practice (Brown & Duguid, 2001).

The relative complexity at a given boundary between communities of practice is characterized by three relational properties of knowledge: difference, dependence, and novelty (Carlile, 2004; Carlile & Rebentisch, 2003). Difference refers to a difference in the amount of knowledge accumulated (e.g., novice-expert distinction) or in the type of domain-specific knowledge accumulated (e.g., specialization in different domains). Increasing difference in the amount or type of knowledge between actors increases the required effort to share and assess each other's knowledge (Carlile, 2004, p. 556). If knowledge is different in kind, and not just in degree, then managing dependence is also needed (Carlile, 2004, p. 556). Dependence, meanwhile, is a condition where two entities must take each other into account if they are to meet their goals (Litwak & Hylton, 1962)—for example, activities of co-workers who rely on each other. The third relational property of knowledge is how novel the circumstances at a boundary are (Carlile, 2004, p. 557). Novelty "comes when an actor [e.g., the customer] is unfamiliar with the common knowledge being used to represent the difference and dependence between domain-specific knowledge" (Carlile, 2004, p. 557). Novelty often is related to "a lack of common knowledge to adequately share and assess domain-specific knowledge at a boundary" (Carlile, 2004, p. 557).

Specifically, the concept of novelty helps to establish that shared knowledge is *always* based on shared understanding. Only if two actors share the understanding of a thing or concept can they share knowledge about the thing/concept, and start to accumulate new knowledge (Holten & Rosenkranz, 2011, p. 568). This implies that there can never be shared knowledge without shared understanding; shared understanding is a logical prerequisite for shared knowledge. All stakeholders

in IS development need to align their "interpretative schemes" (Giddens, 1984, p. 29) in interactions so that they are able to develop "intersubjectively-held mental models" (Gasson, 1999, p. 89) or "shared technological frames" (Orlikowski & Gash, 1994, p. 178) before they can begin to share knowledge, accumulate new knowledge, and build shared or common knowledge.

Socio-cognitive theories of communication help to frame and understand how this process unfolds (e.g., Brennan & Clark, 1996; Clark & Brennan, 1991; Clark & Krych, 2004; Garrod & Pickering, 2009). Actors' frames of reference or interpretative schemes are situation models. A situation model is a mental, multi-dimensional representation of the situation under discussion, including encodings of the key dimensions space, time, causality, intentionality, and reference to main entities under discussion (Pickering & Garrod, 2004, p. 172). Such models are assumed to capture what people are "thinking about" while they communicate; they can be contrasted with linguistic representations on the one hand and common knowledge on the other (Pickering & Garrod, 2004, p. 172). Communication is successful to the extent that actors come to understand relevant aspects of the world in the same way as each other—in other words, the degree to which they align their situation models, create common situation models, and converge on a shared understanding (Brown-Schmidt & Tanenhaus, 2008; Pickering & Garrod, 2004).

Alignment of linguistic representations at syntactic, semantic, and pragmatic levels, therefore, plays a causal role in the attainment of shared understanding. Failure to align at different levels of language, furthermore, may result in failure to communicate successfully (Branigan, Pickering, McLean, & Cleland, 2007; Branigan, Pickering, Pearson, & McLean, 2010; Haywood, Pickering, & Branigan, 2005; Menenti, Garrod, & Pickering, 2012; Pickering & Garrod, 2004; Stephens, Silbert, & Hasson, 2012). The role of linguistic alignment for successful communication has also been shown in IS development (Corvera Charaf, Rosenkranz, & Holten, 2013; Rosenkranz, Corvera Charaf, & Holten, 2013). Of course, actors need not entirely align their situation models: "In (unresolved) arguments, interlocutors have representations that cannot be identical. But they must have the same understanding of what they are discussing in order to disagree about a particular aspect of it" (Pickering & Garrod, 2004, p. 172). For example, if two people are arguing about the merits of SAP versus Oracle software, they must agree about, for instance, what the names refer to and the two products' characteristics so that they can disagree on their evaluations (Pickering & Garrod, 2004, p. 173). We assume, therefore, that successfully sharing knowledge in the case of novel things and concepts involves at least approximate alignment at the level of the situation model (Pickering & Garrod, 2004, p. 174)—the building of shared understanding.

Carlile (2004) already implicitly alluded to this idea by adapting Shannon and Weaver's (1949) three levels of communication complexity—syntactic, semantic, and pragmatic—to introduce his classification of knowledge boundaries (see Figure 1).
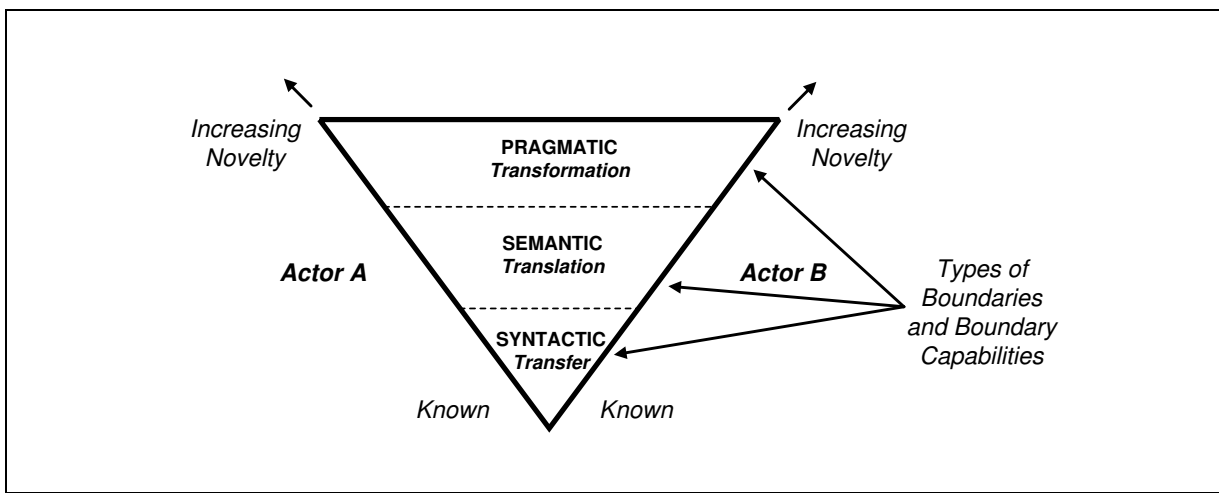

Figure 1. Types of Knowledge Boundaries (Carlile, 2004, p. 558)

The relational properties of knowledge can be represented by sketching a knowledge boundary as a vector between at least two individual actors (Carlile, 2004, p. 557). At the vector's origin, the differences and dependencies are known. As novelty increases, the vector spreads, scaling the increasing relative complexity of the circumstances at the knowledge boundary between actors and the amount of effort required to manage the boundary (Carlile, 2004, p. 557). Depending on the differences, dependence, and novelty of knowledge at the boundary between two actors, the boundary is either syntactic, semantic, or pragmatic in nature. The creation of a "common lexicon" between actors is sufficient to establish shared understanding at a syntactic boundary. Knowledge can then be simply transferred across the boundary because shared understanding and common knowledge already exist (Carlile, 2004, p. 558). While a "common lexicon" is necessary for all three types of boundaries, novelty at a semantic boundary makes some differences and dependencies unclear or some meanings ambiguous. This results in interpretive differences between actors (Carlile, 2004, p. 558). Establishing shared understanding in these situations requires translating and negotiating the different meanings between actors to address interpretive differences in situation models across boundaries (Carlile, 2004, p. 559). At a pragmatic boundary, efforts to overcome differences and dependencies increase further. To transform domain-specific knowledge and common knowledge-in-use induces further cost for the creation of shared understanding and the alignment of situation models. Any actor has to learn what is new and how to transform their current common and domain-specific knowledge to share and asses knowledge at a pragmatic boundary (Carlile, 2004, p. 559).

2.2. Brokers and Boundary Objects

Theories of situated learning (Brown & Duguid, 1991; Wenger, 1998) provide two key concepts to explain further knowledge sharing between actors: brokers (Wenger, 1998) and boundary objects (Fleming & Waguespack, 2007; Pawlowski & Robey, 2004; Star & Griesemer, 1989).

Brokers are individuals who participate in the work of multiple communities and facilitate knowledge sharing across the communities' knowledge boundaries (Brown & Duguid, 1991; Pawlowski & Robey, 2004). Brokering refers to activities by such individuals that involve facilitating transactions and the sharing of knowledge between communities of practice across knowledge boundaries (Fleming & Waguespack, 2007; Pawlowski & Robey, 2004; Pawlowski et al., 2000). It "involves processes of translation, coordination, and alignment between perspectives" (Wenger, 1998, p. 109). Brokers may be weakly linked to several communities at once and full members of none. This is in contrast to so-called boundary spanners (Aldrich & Herker, 1977; Ancona & Caldwell, 1998; Leifer & Delbecq, 1978), who belong to a specific community of practice and span the boundary between communities from the inside of their own communities (e.g., IT professionals in IS development projects who become boundary spanners-in-practice between IT and user communities). Brokers are strategically positioned to facilitate knowledge flow across communities (Pawlowski & Robey, 2004, p. 649). The investigation of brokers has led to the recognition of the skills needed for IT professionals (Chakraborty et al., 2010), and to the identification of role-specific brokering activities such as analysts or developers (Levina, 2005; Levina & Vaast, 2005; Pawlowski & Robey, 2004).

Boundary objects are any "artifacts, documents, terms, concepts, and other forms of reification around which communities of practice can organize their interconnections" (Wenger, 1998, p. 107). They are "both plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites" (Star & Griesemer, 1989, p. 393). These objects are relevant to the practices of multiple communities, but may be used or viewed differently by each of them (Brown & Duguid, 1998; Pawlowski & Robey, 2004, p. 650). Boundary objects, therefore, can have different meanings in different contexts, but their structure is common enough across contexts to make them recognizable (Star & Griesemer, 1989). Requirements specifications, data and process models, diagrams, or program code excerpts are all instances of boundary objects used during requirements elicitation (Bergman, Lyytinen, & Mark, 2007; Gal, Lyytinen, & Yoo, 2008; Levina, 2005; Nicolini, Mengis, & Swan, 2011). The study of boundary objects has resulted in the identification of different types and classifications of such objects (Bergman, King, & Lyytinen, 2002; Carlile, 2002; Carlile, 2004; Garrety & Badham, 2000), and different uses of boundary objects in various IS development contexts (Doolin & McLeod, 2012; Henderson, 1991; Horton & Wood-Harper, 2006; Levina & Vaast, 2005; Pawlowski & Robey, 2004; Yakura, 2002).

Depending on the type of complexity at the knowledge boundary between two actors (Figure 1), boundary objects need not only facilitate the representation of the knowledge but also its potential translation and transformation (Carlile & Rebentisch, 2003). Boundary objects at a syntactic knowledge boundary need to have syntactic capacity to establish a shared syntax or language for individuals to represent their knowledge; for example, a repository that develops a "common lexicon", which is then sufficient to share and assess knowledge (e.g., cost databases, taxonomies) (Carlile, 2002, p. 451; Carlile, 2004, p. 560). At a semantic boundary, boundary objects need semantic capacity to provide a means for actors to specify and learn about their differences and dependencies in knowledge (e.g., standardized forms and methods that provide a shared format for solving problems across different domains) (Carlile, 2002, p. 451; Carlile, 2004, p. 560). At a pragmatic boundary, boundary objects need pragmatic capacity to facilitate the transformation of knowledge (e.g., prototypes as objects that can be jointly transformed) (Carlile, 2002, p. 452; Carlile, 2004, p. 560). Pragmatic boundary objects are the most helpful in dealing with pragmatic boundaries, but they are also the most complicated and expensive to establish (Carlile, 2002, p. 452).

In case of a mismatch between the complexity of the boundary faced and the boundary object used, effectively sharing and assessing each other's domain-specific knowledge can be handicapped. Hence, boundary objects with different capacities are required (Carlile, 2004, p. 565). The different types of boundary objects with their different capacities also draw our attention back to the actors using those objects. At a syntactic knowledge boundary, we do not need brokers because actors can easily transfer their knowledge on their own (Carlile, 2004, p. 560). However, at semantic and pragmatic boundaries, it is not possible just to match the capacity of the boundary object with the type of boundary faced. We also need to make sure that the actors have the *ability* to use it, which depends on the knowledge of the actors in the first place (Carlile, 2004, p. 565), and brokers may be needed. The total capability to manage knowledge at a boundary can then be stated as (Carlile, 2004, p. 565):

$$capacity \times ability = capability \tag{1}$$

Table 1 summarizes the key concepts drawn from the literature on brokering as outlined above.

Table 1. Key Concepts for Brokering in Requirements Elicitation	
Concept and description	**Key references**
Knowledge boundary: • Separate project participants who are members of different organizational entities (different organizations, different organizational subunits, or different communities of practice). • Different types of knowledge boundaries exist (syntactic, semantic, and pragmatic).	(Brown & Duguid, 2001; Carlile, 2002; Carlile, 2004; Carlile & Rebentisch, 2003)
Boundary spanner: • Individuals that engage in boundary spanning activities towards other organizational entities than their own. • This includes knowledge sharing and managing the coordination as well as the political maneuvering needed for the knowledge sharing across the knowledge boundaries.	(Aldrich & Herker, 1977; Ancona & Caldwell, 1992; Ancona & Caldwell, 1998; Argote et al., 2003; Hargadon, 1998; Leifer & Delbecq, 1978; Levina & Vaast, 2005; Yan & Louis, 1999)
Broker: • Individuals among project participants that engage into boundary spanning activities between different communities of practice. • Familiar with and participate in the work of multiple communities of practice. • May be weakly linked to several communities of practice at once (and full members of none), strategically positioned to facilitate sharing of knowledge across communities of practice.	(Brown & Duguid, 1991; Fleming & Waguespack, 2007; Pawlowski & Robey, 2004; Wenger, 1998)
Boundary object: • Objects that are shared and shareable across different problem-solving contexts. • Enable, depending on their capacities, the transfer, translation, and transformation of knowledge across knowledge boundaries. • Different categories of boundary objects with different characteristics and capacities (e.g., repositories with syntactic capacity for representing knowledge).	(Carlile, 2002; Carlile, 2004; Carlile & Rebentisch, 2003; Levina & Vaast, 2005; Star & Griesemer, 1989)

While the concepts of knowledge boundaries, boundary spanners, and brokers explain "who" communicates and shares knowledge and "what types of knowledge" across "what types of boundaries" between different knowledge domains, the concept of boundary object explains one of the actual "means" for doing so. This provides us with a terminology on brokering and a first conceptual lens that help us to describe and explain the interplay of boundary objects and brokers— the who, what, where, and when—in requirements elicitation and their effects on the building of shared understanding and knowledge sharing.

2.3. Brokering Situations as an Analytical Lens for the Interplay at Knowledge Boundaries

The interactions at the knowledge boundaries between diverse occupational groups are an important area of current research (Argote & Miron-Spektor, 2011). Conditions of boundary objects' use and

emergence have been analyzed (Levina, 2005; Levina & Vaast, 2005) and some indications of how they shape the social dynamics of interacting communities have been described (Gal et al., 2008; Subrahmanian et al., 2003). However, we need concepts and theories that help to explain how project participants from different communities interact and use boundary objects to collaborate (Lee, 2007; Tsoukas, 2009). Although the types and the use of boundary objects have been examined in multiple settings, further insights are gained by shifting the research focus from the objects to the actual practices surrounding the use of the objects (Argote & Miron-Spektor, 2011). Researchers should concentrate on examining the *interplay* between boundary objects and brokers, rather than observing them separately (Kimble et al., 2010).

In previous IS research, brokering (Pawlowski et al., 2000, p. 335) has mostly referred to IT professionals such as analysts or developers as potential brokers who manage coordination and knowledge sharing across the borders of different communities of practice. Building on this definition, we define the concept of a brokering situation:

> *A brokering situation is a knowledge-sharing situation with at least two actors from different communities of practice, in which each actor has specific types and amounts of domain-specific knowledge and can become a broker, and where the actors use boundary objects to build shared understanding and to facilitate knowledge sharing.*

To examine the interplay, we have to observe individual actors, used boundary objects, increases or decreases in novelty of the circumstances at the boundary during brokering situations, and thus an increase or decrease in relative complexity (Figure 2). We must examine the existing capability (capacity x ability) at each boundary between actors because we have to take into account different types and changes in the amounts of knowledge of actors and the interdependence between types of knowledge. The concept of brokering situations allows us to investigate the capability present in specific circumstances and the changes of types and amounts of knowledge during the interaction between actors and their interplay with boundary objects—the left part of Equation 1. This allows us to audit these different elements with regard to the building of shared understanding and knowledge sharing.
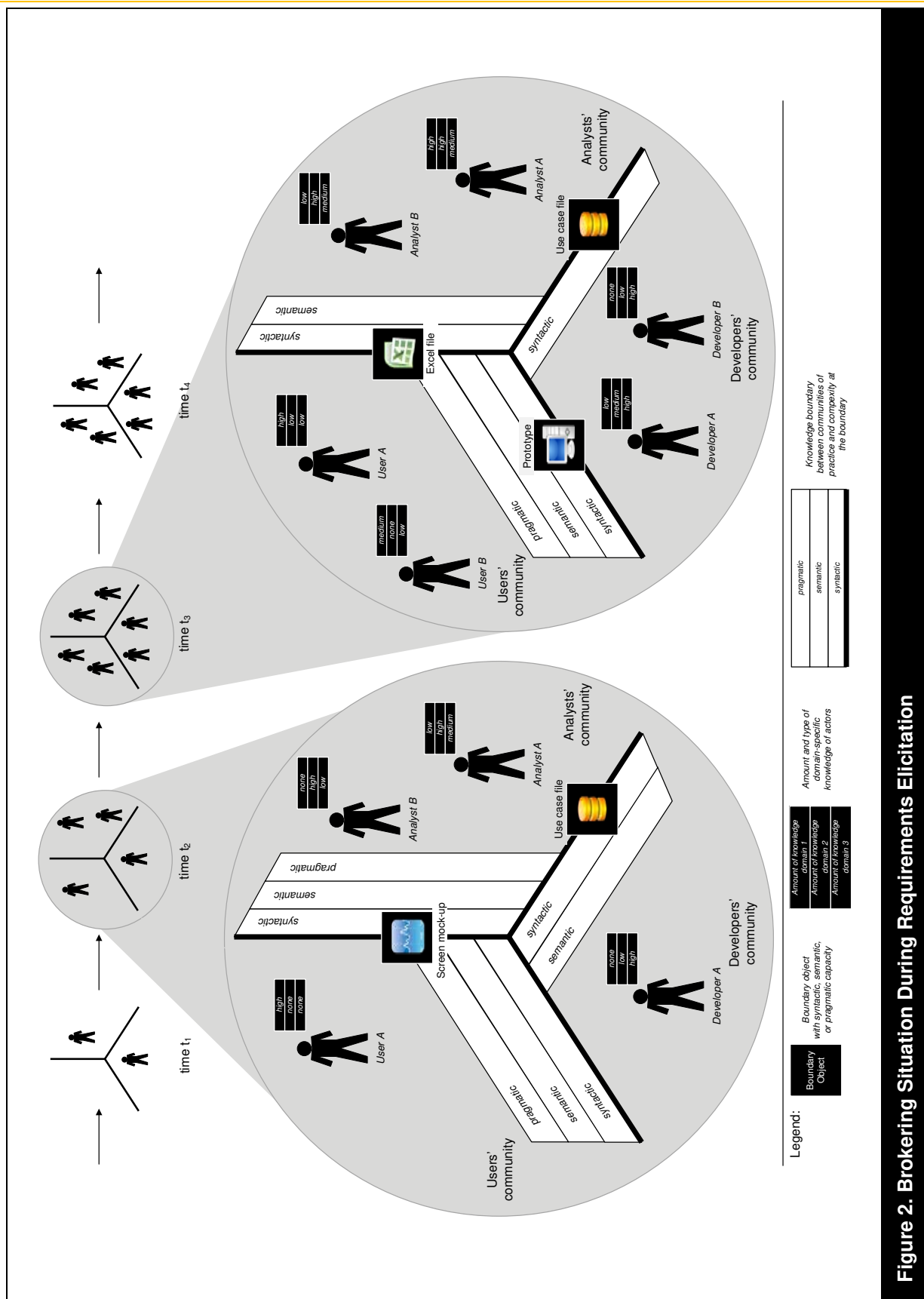
Figure 2. Brokering Situation During Requirements Elicitation

3. Research Study and Design

3.1. General Description

We conducted an interview-based study (Miles & Huberman, 1994, pp. 25-27) in which we applied our conceptual framework to examine more closely the interaction between brokers and boundary objects in brokering situations, and to observe the building of shared understanding and sharing of knowledge during the requirements elicitation phase of IS development. The study was informed by case study research (Yin, 2003, pp. 13-14) in that we investigated a similar scope with multiple cases and developed a conceptual framework a priori. Furthermore, we followed similar procedures for data collection and data analysis (see Appendix A for details on criteria to assess the study).

We found the development of data warehouses to be an adequate setting for this investigation. A data warehouse (DWH) collects and integrates data from various, heterogeneous source IT systems and provides them for analytical applications. Analytical applications, in practice often described as "business intelligence", allow users to access these data and use them to make decisions (Watson & Wixom, 2007). This results in a complex IT ecosystem with a broad scope of content and multiple project participants from different communities with differing expertise and domain-specific knowledge; for example, end users, business experts, analysts, DWH developers, or IT experts for operational systems (March & Hevner, 2007; Wixom & Watson, 2001).

We can identify three archetypical roles with corresponding communities for DWH development projects: DWH professionals, system professionals, and business experts. DWH professionals are analysts, designers, or developers of the to-be-developed DWH. As IT professionals, they need to possess broad knowledge from both business and IT domains (Pawlowski & Robey, 2004, p. 666) and are predestined to assume the part of brokers during these projects. On the one hand, for DWH professionals, eliciting end users' requirements means meeting end users, who are mostly business experts with domain-specific business knowledge. Definitions of sample reports, for example, are instances of required information in these meetings and thus can be used as boundary objects. On the other hand, DWH professionals design the technical extraction, transformation, and loading (ETL) processes that are necessary for data integration (March & Hevner, 2007). This occurs in conjunction with interaction with system professionals who are IT experts that develop, operate, and maintain source data systems.

3.2. Data Collection

We collected data from interviews in three interconnected phases (phases I to III). Appendix B gives details on the data collection. We conducted the interviews in Croatia, which lasted from 30 to 120 minutes (M = 52 minutes; SD = 24 minutes). We audio-recorded and transcribed all interviews. This resulted in 477 transcript pages with 189,862 words of text. In the first two phases, one researcher conducted interviews with 20 experienced DWH professionals (Table 2) who had been working on various development projects in different industries, either in-house or as consultants. All interviewees held a master's degree in computer science or related fields. The number of participants in the reported projects ranged from four to 70, with an average of 15. The duration of projects varied from six months to six years. Phase I covered eight open interviews. We applied a variant of the critical incident technique (Flanagan, 1954) and used no interview guideline. Instead, we generally asked informants to identify and discuss their activities and the objects they used in the early phases of their DWH development projects; for their impressions of the goals of business experts and system professionals, and of their own project goal; and to describe the activities of DWH professionals. Unprepared probing followed each of these open questions.

Interviewee[a]	Years in IS	Years in DWH	Reported projects[b]	Industries	Interview in phase I	Interview in phase II
James	6	2.5	Lima, Mars	Banking / telecommunications	✓	✓
Ryan	8	8	Oscar, Echo	Banking / energy	✓	✓
Tyler	8	8	Unity, Victory	Telecommunications / health insurance	✓	
Jack	12	2	Zulu	Higher education	✓	
Kevin	8	8	West	Higher education	✓	
Alex	9	7	Clover, Delta, Xray	Banking / insurance / trading	✓	✓
Justin	10	7	India	Telecommunications	✓	
Isaac	10	10	Alpha, Yankee	Banking / energy	✓	
Eric	5	5	Foxtrot	Trading		✓
Ian	15	10	November	Telecommunications		✓
Luis	11	8	Purple, Oscar	Trading / metal industry		✓
Adam	4	3	Hotel, Golf	Banking		✓
Julia	6	4	Jazz, Kilo	Banking		✓
David	6	6	Gamma	Banking		✓
Emma	3	3	Romeo	Telecommunications		✓
Sara	8	8	Echo, Sierra, Tango	Energy		✓
Caroline	5	4	Quebec	Telecommunications		✓
Kyle	12	10	Alpha, Bravo	Banking / tobacco industry		✓
Jake	11	11	Alpha, Bravo	Banking / tobacco industry		✓
Amber	3	2	Clover	Banking		✓

Table 2. Overview of Interview Subjects and Reported Projects (Phase I and II)

[a][b] All names and project abbreviations are fictitious and anonymized for confidentiality

In phase II, we directed our questions towards revealing all possible events and factors that enable DWH professionals to elicit requirements successfully. We also examined events and factors that facilitate the building of shared understanding, in addition to events and factors that prompt system professionals and business experts to share their knowledge with each other and with DWH professionals successfully. Phase II covered 15 semi-structured interviews and focused on exploring brokering situations in more detail. We used topic guidelines, and asked interviewees to be as inclusive as possible in their descriptions of the team's skills and familiarity with the project's business domain; individuals from the communities of system professionals and business experts, including their familiarity with the business domain and their familiarity with the IT background of the data delivered by source systems; and the objects they used during requirements elicitation.

In phase III, we closely investigated two specific projects (Golf and Echo) and one additional, new project in much more detail (Table 3, Project Omega). We collected additional data on a new project

during phase III because this allowed us to validate our coding scheme, which emerged from phases I and II (triangulation across projects). This time we also conducted semi-structured interviews with team members from the communities of system professionals and business experts to ensure data sampling from the members of all three participating communities (triangulation within projects). In addition to the topic guidelines used in phase II, we designed the semi-structured interviews in phase III to investigate the subsequent events and changes over time regarding participants' familiarity with the domain knowledge of all three involved communities, all changes of personnel, and all changes of boundary objects in the project.

Table 3. Overview of Interview Subjects and Reported Projects (Phase III)			
Interviewee[a]	Community of practice	Reported projects[b]	Industry
Mary	Business expert	Golf, Omega	Banking
Molly	DWH professional	Omega	Banking
Mark	DWH professional	Omega	Banking
Lillian	System professional	Omega	Banking
John	Business expert	Echo	Energy
Nicole	System professional	Echo	Energy
[a b] All names and project abbreviations are fictitious and anonymized for confidentiality			

3.3. Data Analysis

Two researchers coded the interviews in all three phases using MaxQDA as a software and open coding (Miles & Huberman, 1994, pp. 56-67). First, two researchers coded the first round of interviews (phase I) in Croatian, whereas two other researchers coded the transcripts of the second and the third rounds (phase II and III) after they were professionally translated into English. We started with initial seed codes based on our framework of brokering situations. Our complete coding scheme gradually emerged, and from phase II onwards we concentrated on exploring brokering situations in more detail by coding boundary objects, different types and amounts of knowledge that participants brought into the brokering situation, and outcomes of brokering situations (e.g., misunderstanding, developed shared understanding, differences in amounts of knowledge and types of knowledge). If objects were characterized as boundary objects, then we scrutinized the data for statements relating to their capacity and we coded them as "syntactic boundary object", "semantic boundary object", or "pragmatic boundary object" according to Carlile's (2002, 2004) categorization. In phase II, we also used causal mapping to explore relations in the data. (Appendix B gives details on the data analysis and the full list of codes of the coding scheme.)

Figure 3a shows an excerpt of our preliminary coding scheme for brokering situations from phase I. The final coding scheme for this excerpt includes domain knowledge for all three participating communities (DWH professionals, business experts, and system professionals), and the three categories of boundary objects (syntactic, semantic, and pragmatic). Both coders iteratively revised the coding scheme until they determined that it reflected all relevant themes or issues (Eisenhardt, 1989). We gradually added new codes to the coding scheme (cf. Figure 3b, where several additional codes for "knowledge" have been added as categories). After we conducted and analyzed 20 interviews, we jointly posited that data saturation had for the most part occurred because new themes did not emerge and the coding scheme remained stable (Guest, Bunce, & Johnson, 2006). In the next paragraph, we discuss our analysis and show the genesis of the coding scheme for brokering situations in more detail.
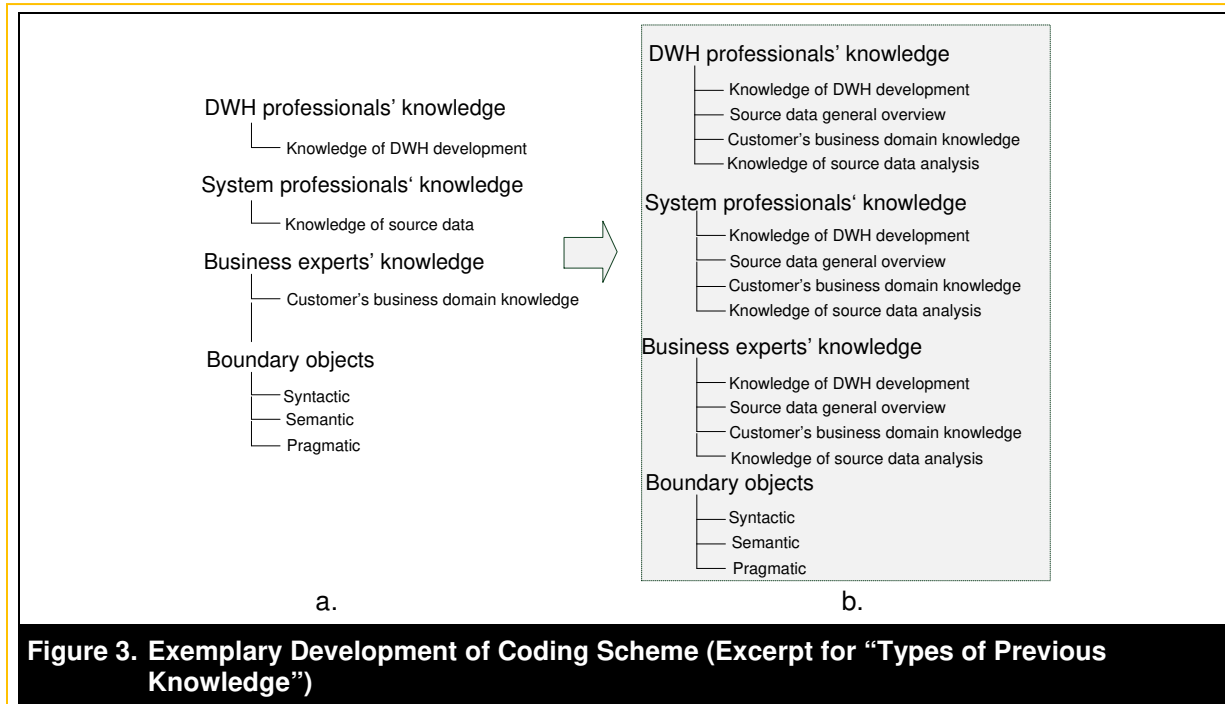
Figure 3. Exemplary Development of Coding Scheme (Excerpt for "Types of Previous Knowledge")

4. Analysis and Results

4.1. Types of Domain-specific Knowledge and Knowledge Boundaries

First, we need to determine the occurring types of knowledge in order to assess the ability of project participants, the capacity of boundary objects, and the resulting capability to share knowledge. The first type of domain-specific knowledge in DWH development projects relates to DWH professionals and their technical knowledge of DWH development (Figure 3a: knowledge of DWH development). However, requirements elicitation in DWH development necessitates "a clear definition of business needs" (Hwang & Xu, 2008, p. 52), exposing DWH professionals to problems that could be outside the domain of their natural competences. DWH professionals also need to be familiar with the respective business domains in order to develop a DWH that fulfills business experts' requirements. This is the second type of domain-specific knowledge in DWH development. Several interviews across different projects from phases I and II confirmed that the DWH professionals' community brought in those members who were most familiar with the customer's business domain, and that they acted as brokers between DWH professionals and business experts (extension of the DWH coding schema in Figure 3b: customer's business domain knowledge). For example, Alex said: "If you have experience in the customer's business branch, then you partially know what customers want from you. Although they have problems expressing themselves, you can adjust their statements and produce more fitting reports" (Alex, Project Delta).

The third and fourth types of domain-specific knowledge also relate to technical knowledge. DWH professionals need to possess at least peripheral knowledge about the source systems and have an overview of how to analyze source data. Several informants confirm this. For example, in the case of the in-house Project West, the majority of the community of DWH professionals consisted of employees from the company and no external consultants. Our key informant in this project notes that the DWH professionals had at least a basic understanding and a general overview of source data and their analysis. If not, this creates a knowledge boundary between DWH professionals and system professionals, as reported for Project Purple:

> *Interpretation and allocating… and explaining the sources. Cause you get a table, named literally "TAB1" which has columns: "COL1" to "COL30"—which actually stands for, and I am making this up now—a table of invoices that has something in it. You, without someone able to explain this to you, simply cannot get what it is.* (Louis, Project Purple)

Thus, we further extended the coding scheme for DWH professionals with the categories "source data general overview" and "knowledge of source data analysis" (Figure 3b).

Business experts typically have strong business domain knowledge (Figure 3a: customer's business domain knowledge). However, they generally do not posses extensive technical knowledge and usually do not engage in detail with technical components or IT systems. This typically creates a knowledge boundary between business experts and DWH professionals or system professionals. Nevertheless, our analysis revealed that some business experts had previously conducted source data analyses and were, therefore, very familiar with the origin and meaning of the source data required for further DWH development. As a result, we extended the coding scheme for business experts also with the category knowledge of source data analysis (Figure 3b):

> *Perhaps they have worked with similar* [reporting] *systems before or they have managed to get the same functionality they later received with the new system by using, for example, Excel. The goal is the understanding of data, and you generally have to do that on your own. Sometimes such people* [i.e., technically knowledgeable business experts] *unexpectedly help you.* (James, Project Mars)

Similarly, system professionals usually have extensive technical knowledge about source systems. However, they typically lack business domain knowledge, but can expand their "natural" competence domain (Figure 3a: knowledge of source data) with the domain of the customer's business in day-to-day work and data preparation for business users (extension of the system professionals' coding schema, Figure 3b: customer's business domain knowledge; knowledge of source data analysis). These systems professionals, moreover, play an important role in data integration in aligning the shared understanding of all relevant concepts and terms for all project participants right from the start of the project:

> *They had people who were in charge of certain segments of that* [ERP] *system. We actually never talked to anyone who would be the architect of the entire system. … They did seem like islands over there. … I remember that there were problems ... it's hard to catch someone who knows how to work with it.* (Jake, Project Alpha)

Thus, we changed the preliminary code for the system professionals' previous domain knowledge from "knowledge of source data" (Figure 3a) to "source data general overview" (Figure 3b). Furthermore, several companies already have running DWHs because DWH technology by now has been maturing for several years. Several statements from informants mention that system professionals with work experience on previous versions of a DWH or in data integration influence the development of shared understanding and the sharing of knowledge between participants. This led to further extension of the system professionals' coding schema (cf. Figure 3b: knowledge of DWH development):

> […] *for a long number of years, actually, the company did have a data warehouse and a reporting system, however, only for Sales, but some of the people were already familiar with it, part of the IT* [system professionals] *… They worked on the ERP system and the warehouse. So, they managed well with the user needs* [...] *They can manage the whole story and they have some inside information on how all of it works.* (Kyle, Project Bravo)

Based on the iteratively revised codes from phases I and II (Figure 3), we conclude that these four different categories of domain-specific knowledge are prevalent in DWH development and create knowledge boundaries between project participants:

(1) Customer's business domain knowledge

(2) Knowledge of source data analysis

(3) Source data general overview, and

(4) Knowledge of (previously conducted) DWH development.

4.2. Community Representatives and Brokers

Previous research has mostly grouped enabling or inhibiting factors according to participants' roles in a project (e.g., "business analyst", "user", "developer"; see Section 2.2 on related work) (Chakraborty et al., 2010; Guinan & Scudder, 1989; Levina & Vaast, 2005; Sawyer, Guinan, & Cooprider, 2010; White & Leifer, 1986). In contrast, several statements in our data show that the line between those participants who possess necessary domain-specific knowledge and those who do not was not restricted by their archetypical role, and was moving dynamically in the course of different projects. Hence, several informants mentioned that not only DWH professionals (as designated IT professionals) acted as brokers. For example, one interviewee reported a situation for Project Echo in which a business expert was not only very familiar with the source data, but had apparently participated in the development of the corresponding source systems. This "business expert broker" was able to define his requirements by directly referring to the source systems and by explicitly naming specific data fields. In addition, he helped the DWH professionals define the ETL processes. Hence, he acted as a broker and exhibited membership in both the business experts and system professionals communities:

> *We've worked with quite an advanced user that had a degree in math, worked at first in the production department, afterwards in the IT department and de facto developed their information system, and now works in the department of strategic planning. He understands both IT and the company's business process, being able to sketch examples of reports he expected. We pretty much understood it all.* (Ryan, Project Echo)

Subsequent probing in phase II confirmed that the amount of knowledge in the four aforementioned categories, and the ability to act as brokers, varied for members of each participating community from project to project. For example, the already reported episode from Project Bravo, where DWH technology had already existed for several years, shows that system professionals can also become familiar with DWH development (cf. interview with Kyle, Project Bravo). An episode from in-house Project Foxtrot further demonstrates that only few business departments (business experts) had knowledge of source data analysis due to the nature of their job. As a result, two departments (D1 and D2) had business experts that acted as brokers, and who were able to articulate future system requirements:

> *Most of them* [users from D1, D2] *knew exactly what they wanted and what they could get from the information system. They didn't have unrealistic requirements because they knew the system's limits (restrictions) and such things … they had that somehow in their heads. Most of the users who work with reports understand neither the DWH nor what is going on in the background* [in D3 through D6]*, whereas they* [users from D1, D2] *had a good idea about it. There* [in D3 through D6] *were mainly business clients who knew how to define very good reports, but communication with someone, who has background in informatics* [as in D1, D2]*, was much easier…* (Eric, Project Foxtrot)

Our analysis also reveals that, in the absence of business experts or DWH professionals acting as brokers (as for D3, D4, D5, and D6), system professionals who were familiar with those businesses were able to compensate for ambiguous statements in business experts' requirement definitions:

> *There were cases when users had a request that we could not understand. Then we explained to them what they could and could not get. Finally we have adjusted their requirements. Some of us were part of the department of informatics* [system professionals] *and we were supposed to know the business side of the data very well. We could explain to the user what can be produced and what couldn't since we knew what data were available.* (Eric, Project Foxtrot)

As in the above section for DWH professionals (see Section 4.1), these observations led to extensions of the coding schemes for business experts and system professionals (Figure 3). To sum up, the amount of knowledge of a participant is not restricted by their usual, role-based community; individuals from each of the three different communities can have knowledge in each of the four different types, enabling them to act as brokers. DWH professionals, as the proverbial IT professional, are not the only brokers.

4.3. Boundary Objects

Informants from all projects mentioned a wide range of boundary objects that were used during requirements elicitation—for example, prototypes, diagrams, database documentations, ETL mappings, report printouts. The results of our analysis confirm Carlile's (2002, 2004) classification of the relative complexity of the circumstances at the knowledge boundary, and the importance of the fit between the boundary's complexity and the capacity of the used boundary object for successful knowledge sharing.

For Project Clover, for example, informants referred to the use of "Excel spreadsheets" with examples of required calculations in reports. During the initial meetings of Project Clover, representatives from the business expert community were reported to have shared requirements specifications with DWH professionals in the form of old reports that had been defined in Excel spreadsheets. The spreadsheets were intended to represent a cross-boundary repository of requirements specifications (syntactic boundary object). However, our analysis indicates that the DWH professionals were initially unable to interpret the business experts' requirements from the Excel spreadsheets:

> *Well, one of the biggest mistakes, I wouldn't call them mistakes but misunderstandings, happened due to the fact that we understood something differently* [from what they had in mind]. *We had literally implemented according to their definitions in that* [Excel] *table. When they saw an example in our prototype, they responded: "well, we don't want this in this way, but in the other way"… A second problem was that they changed the* [Excel] *table with the calculations 4 to 5 times… Only after we put the developed prototype into production, they realized that they had made a mistake* [in the definitions]. *So they changed the* [Excel] *table.* (Alex, Project Clover)

The initial knowledge boundary between DWH professionals and business experts in this case was not syntactic but semantic (or even pragmatic) in nature. The DWH professionals were unable to look at an existing report and understand its structure. In other words, although the DWH professionals had some general business domain knowledge because they previously had worked in similar industries, they were not equally proficient as members of the business experts' community in terms of particular knowledge about the specific business, which was a novelty for them. This deficit in shared understanding and common knowledge with business experts hampered their ability to interpret the implicit knowledge embodied in the used Excel spreadsheets (syntactic boundary object). The novelty and knowledge imbalance created a semantic knowledge boundary across which knowledge not only had to be transferred, but also had to be transformed into a "common lexicon" (Carlile, 2004, p. 558) that DWH professionals could interpret. The quote above also shows that overcoming this semantic boundary became possible when participants used a boundary object with pragmatic capacity—the prototype system.

Moreover, this situation shows that project participants in such situations may think that they have understood what they have been told, have read, or have seen, even though they have not. DWH professionals might believe that they have reached a shared understanding with the business experts, whereas, in fact, they have not aligned their situation models and only experience an "illusion of evidence" (Bromme, Jucks, & Runde, 2005, p. 90). For example, only by the time DWH professionals and business experts jointly reviewed the first prototype system (pragmatic boundary object) in Project Clover had they realized that the DWH professionals had misinterpreted the business experts' requirements, which they had deduced from old reports:

> *We were lucky to have the milestone meetings while we were developing. We were presenting them the results of our development, allowing them just-in-time corrections. The end result was of a very good quality and they were very satisfied with the final product as well… Those milestones, during which they continuously corrected our work, saved us literally. They were able to express what they thought made sense and what didn't..., and only when they saw the* [finished] *system, they figured out that they had expressed themselves inaccurately.* (Alex, Project Clover)

In this case, the prototype system facilitated a process where individuals could jointly transform their knowledge. Only after switching to boundary objects that had the adequate (pragmatic) capacity for discussions about the meaning of the calculations, the DWH professionals were able to increase shared understanding and to share knowledge with the business experts.

4.4. Brokering Situations: The Interplay of Brokers and Boundary Objects

The initial analysis of our results from phases I and II reveals that boundary objects do not only have positive effects on knowledge sharing in brokering situations, but also can have a negative impact in case they are mismatched and their capacity is inadequate for handling the complexity on the knowledge boundary. Specifically, the change of brokers can result in a change of shared understanding, common knowledge, and abilities to share knowledge, causing boundary objects that have been adequate to become mismatched as regards capacity. For example, key informants for Project Oscar report that the departure of a knowledgeable broker (an IT director who also was a system professional) suddenly increased the relative complexity of the circumstances at the boundary, leading to a project standstill. The DWH professionals were unable to interpret the data in a COBOL (Common Business Oriented Language) application on their own (shared repository; syntactic boundary object):

> **A:** *What happened, when the new owners arrived, they brought along a man whom they appointed as IT director at the forge* [...] *However, all the others, his new IT department, they were mostly older people, who were, IT-wise, run over by time. They knew their old Cobol database from which they knew how to pull out some data, print it out on paper, but they had serious issues with Excel, something unheard of for us. You have an IT specialist who can't find his way around Excel, let alone doing something in more modern databases?*

> **Q:** *So who did you do the analysis with, then, with this manager* (IT director)?

> **A:** *More or less, with him, since he had the technical skills and has been troubling himself with these people, yanking information out of them, making them… And that project did not end very well, since… I mean there is a data warehouse now, being filled as we speak, but that man* [the IT director] *has given up in the meantime, he is through with quarreling with the IT-locals there. Now the whole thing is hanging in the air.* (Luis, Project Oscar)

This episode reveals that the departure of such a relevant broker, with the ability to cope with the complexity at the boundary, can result in a case of mismatched boundary objects and can even cause a project stop or termination.

We also find that special brokers' abilities are important for shared understanding to emerge and for closing the occurring knowledge gaps. Other participating communities would have been struggling to exchange knowledge without either the active participation of system professionals familiar with the business experts communities' business (see interview with Eric, Project Foxtrot, in Section 4.1), or "business experts broker" with experience in development of operational IT systems (see interview with Ryan, Project Echo, in Section 4.1). In other words, the knowledge boundary between DWH professionals and business experts would have been pragmatic if system professionals in Project Foxtrot had not been involved in the brokering situation. In that case, developers would have had to switch from the document containing requirement definitions (syntactic boundary object) to boundary objects with more adequate capacity (e.g., prototypes, as in Project Clover). However, the used syntactic boundary object was sufficient due to the system professionals' involvement as brokers.

The capability to manage knowledge really depends, therefore, on the right combination of brokers (possessing the ability, which depends to a large degree on their knowledge) and boundary objects (with adequate capacity). Moreover, as we show in Section 4.2, the distinction between participants with and without necessary domain-specific knowledge is dynamic and not necessarily based on archetypical roles such as business analyst (see Section 2.2). We found that each participant from one community of practice may become familiar with domain knowledge of another community and

can compensate for their missing or inaccurate interpretations of discussed issues during brokering situations, taking on the role of a broker.

Our findings from phases I and II confirm that, on the one hand, the interplay of brokers that have the ability with boundary objects that have adequate capacity is very important for the emergence of shared understanding and successful knowledge sharing between different communities. On the other hand, we found that only looking at either brokers' abilities or boundary objects' capacities cannot explain all effects we discovered in the data in cases where significant gaps in common knowledge and shared understanding were apparent (e.g., in projects Echo, Foxtrot, and Oscar). We must take into account the complete brokering situation and the interplay of brokers and boundary objects to observe how shared understanding emerges and how knowledge is shared.

5. Analyzing the Emergence of Knowledge Sharing: The Boundary Interaction Framework

5.1. Motors of Change in Brokering Situations

We used the data gathered in phase III to conduct a more granular analysis of the types and changes in the amount of participants' domain-specific knowledge. We looked also at the types and changes of boundary objects that participants use during brokering situations. Our intentions were twofold: to get more detailed insights into the interplay of brokers and boundary objects, and to observe how the interplay in the process of requirements elicitation dynamically proceeds and affects the emergence of shared understanding and knowledge sharing.

We can see requirements elicitation as a sequence of brokering situations, each representing a unique combination of boundary objects and participants in time. The knowledge sharing on the boundary between communities then depends on the capacity of the boundary object, the involved participants' abilities, and the boundary itself. We can explain how requirements elicitation progresses or deteriorates by studying how the brokering situations change in time; that is, how the change of the brokering situation affects the emergence of shared understanding and the sharing of knowledge between participating communities. In order to understand and explain the dynamics of this process better, we propose augmenting our concept of brokering situations with teleological and dialectical ideal process types (Van de Ven, 1992; Van de Ven & Poole, 1995; Van de Ven & Sun, 2011) as motors of change in brokering situations.

As regards the first motor of change, requirements elicitation as a sequence of brokering situations in time moves towards one primary goal: the successful elicitation of all relevant users' requirements for an IT system. We can see requirements elicitation, therefore, as a teleological process (Van de Ven, 1992, p. 178; Van de Ven & Sun, 2011, p. 60). The requirements are elicited and a consensus on, and shared understanding of, requirements emerge through purposeful social construction among project participants as the process unfolds (Van de Ven & Sun, 2011, p. 61). The brokering situation is the "organizational entity" (Van de Ven & Poole, 1995, p. 529), and the teleological motors of change explain the change of the brokering situation in terms of movement toward some final goal or a state of "rest" (Van de Ven, 1992, p. 178). We can deconstruct the brokering situation to examine its members (boundary objects and project participants) without losing any explanatory power (Van de Ven & Poole, 1995, p. 522). If we observe the changes of project participants and brokers, the changes of participants' knowledge, and the changes of boundary objects, then we can now observe the teleological change of the brokering situation itself in time and its move toward the goal.

The whole process can break down because project participants do not recognize the need for changes, they make erroneous decisions, they do not reach shared understanding or agreement on goals or actions, or they do not share knowledge (Van de Ven & Sun, 2011, p. 62). This brings the second motor of change to the forefront. While project participants communicate, negotiate, discuss, and use different boundary objects during brokering situations, they drive the elicitation of requirements forward. As our analysis shows, project participants run into misunderstandings during the elicitation of requirements from time to time because of missing abilities or inadequate capacities. This often leads to a conflict between multiple project participants (Van de Ven & Sun, 2011, p. 63).

Such conflicts or "crises" signal a dialectical motor of change (Greiner, 1972; Van de Ven, 1992, p. 184; Van de Ven & Sun, 2011, p. 60), where the crisis represents a mark (signal) of an "antithesis". This is in conflict with the main "thesis". In Project Clover, for example, Alex (cf. interview with Alex above) reports that business experts used Excel spreadsheets as a boundary object on the boundary between DWH professionals and business experts (thesis). However, the DWH professionals were not able to interpret the business experts' requirements from these Excel spreadsheets (antithesis), leading to a crisis in the project.

Including dialectic motors of change accounts for the fact that the brokering situation and its members exist in a pluralistic world of colliding events, forces, contradictory values, or interpretative schemes; they compete with each other for domination (Van de Ven, 1992, p. 178). Change occurs when opposing forces gain sufficient power (antithesis) to rebalance the stability of the status quo (thesis). The resolution of the conflict (synthesis) can then become the new thesis as the dialectical process continues (Van de Ven, 1992, pp. 178-179). To be a constructive force, conflict has to be resolved effectively (Van de Ven & Sun, 2011, p. 63). For example, a first prototype was needed to overcome the reported crisis in Project Clover. In order to create the synthesis and rebuild the balance in brokering situations, we argue that change or replacement of either boundary objects (with adequate capacity) and/or project participants (brokers with knowledge) is required. These different patterns for resolving dialectical oppositions can then push the whole process of requirements elicitation toward:

(1) Equilibrium (e.g., emergence of shared understanding and sharing of knowledge between involved participants, leading to elicited requirements)

(2) Oscillation in cycles between opposites, or

(3) Production of such oscillations that push the process out of equilibrium forever (Van de Ven, 1992, p. 179) (e.g., when the departure of a knowledgeable broker causes the project's termination; cf. interview with Luis above).

This combination of brokering situations and motors of change gives us an analytical framework by which to analyze the dynamics of knowledge sharing in more detail. We call the resulting perspective the boundary interaction framework.

We note that we deliberately reduce both teleological and dialectical motors of changes with respect to the three interlinked dimensions of the duality of social structure and human interaction: signification, domination, and legitimation (cf. Giddens, 1984, p. 30). The dimension of signification focuses the creation of meaning and codes of discourse in the structure (Hussain & Cornelius, 2009, p. 201). Given our focus on the emergence of shared understanding as a prerequisite for knowledge sharing, we analytically concentrate on signification—including communication, interactions, and interpretative schemes (Giddens, 1984, p. 29)—and exclude the other two intertwined dimensions of domination and legitimation. We focus on signification and separate it analytically from domination and legitimation because the creation of meaning and codes, conceptualized as modes of discourse, is a precondition of the other two aspects, which are embedded in these semiotic relations (Giddens, 1984, pp. 32-34).

5.2. Application of the Boundary Interaction Framework

The requirements elicitation phase is a sequence of periods of convergence, punctuated by short periods of time during which changes of boundary objects or project participants occur. Using the boundary interaction framework, we audited episodes of requirements elicitation in the reported projects from phase III in order to observe the interchange of teleological and dialectical motors of change within brokering situations, and their impact on the emergence of shared understanding and knowledge sharing (cf. Appendix B). We analyzed for each project the chain of brokering situations by auditing:

(1) All project participants

(2) All boundary objects the participants used, along with their capacity

(3) All changes of boundary objects or project participants, and

(4) Increase or decrease in the types and amounts of knowledge of the participants.

We used the four previously defined categories of domain-specific knowledge (cf. Figure 3b) regarding project participants and their respective types and amounts of knowledge (customer's business domain knowledge, knowledge of source data analysis, source data general overview, and knowledge of DWH development). We distinguished between "none", "low", "medium", and "high" amounts of knowledge (cf. Black, Carlile, & Repenning, 2004, p. 598). Both coders coded these amounts of knowledge via subcategories of the knowledge categories in the coding tree. Depending on the existing types and amounts of knowledge possessed by participants, a sufficient level of common knowledge between them exists at the beginning of any brokering situation. Next, we audited for all boundary objects used by the participants within a reported brokering situation with regard to their capacities, given that boundary objects with adequate/inadequate capacity can increase/decrease knowledge sharing. Third, we assessed the development of shared understanding or misunderstandings between project participants in brokering situations ("crises" in the data). We coded text sections with categories and subcategories of "shared understanding" for each brokering situation within each episode.

5.3. Emergence of Shared Understanding and Knowledge Sharing: An Exemplary Episode

We illustrate exemplarily and discuss both motors of change based on an episode from Project Omega. The illustrated episode was conducted at a bank. Some members of the DWH professionals' community from external company Infocom had been developing and working on the bank's DWH for several years. However, for the development of a set of new reports, the bank employed a group of DWH professionals from another external company, Softcom. A third community that participated in the project was the business experts from the accounting department of the bank that needed the new reports. Figure 4, which summarizes the process of requirements elicitation during the episode in Project Omega, comes from our application of the boundary interaction framework. It shows project participants' domain-specific knowledge at the beginning of the episode (left side of Figure 4). We highlight changes in the amount of knowledge of participants and in the building of shared understanding between participants over time (middle of Figure 4), and changes in common knowledge (bottom of Figure 4).

Different states (S_0-S_6) between reported brokering situations are positioned on the time line and the emergence of shared understanding and common knowledge was estimated according to the relative difference in reported "fit" of knowledge after every brokering situation. Each brokering situation represents a unique combination of boundary objects and a group of project participants (e.g., Joe, Mark, Emily, and "descriptive requirements specification" in the first brokering situation). These periods of convergence towards the final goal (definition of a final requirements specification for the set of new reports) become interrupted by changes or replacements of boundary objects or project participants (e.g., replacement of Emily with Mary in state S_1). These changes occur due to a teleological or a dialectical motor of change. We indicate each of these motors of change with a circle (top of Figure 4). The symbol in the middle of a circle represents either a teleological or a dialectical motor of change. The used boundary objects are listed on the bottom. The brokering situations that occurred during the episode are described in more detail as follows.

Figure 4. An Exemplary Episode from Project Omega
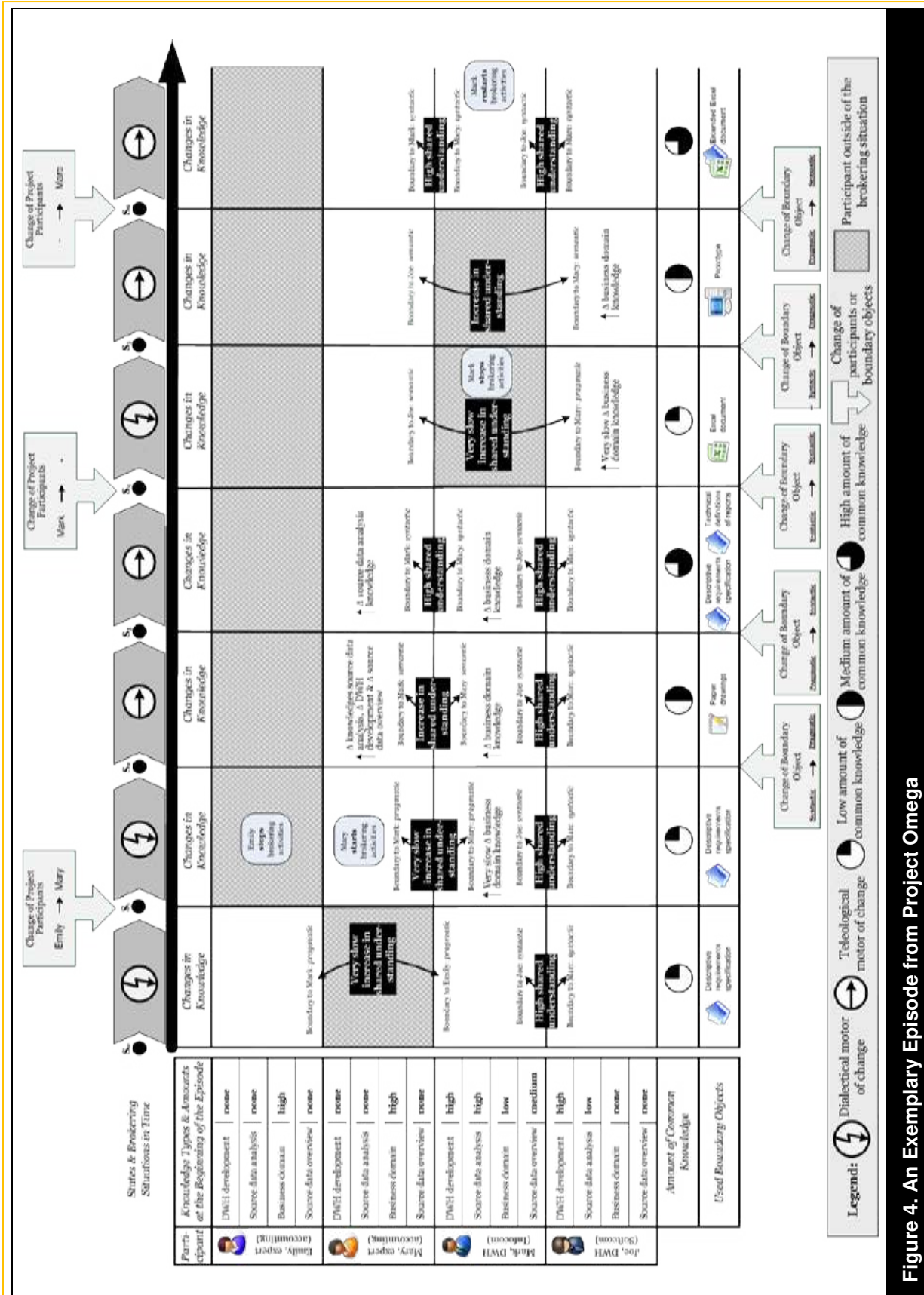
At different stages of the episode, we found evidence of changes of either boundary objects or project participants (or both). As our analysis proceeded, we concentrated on these changes and divided the episode into several different brokering situations, ending or beginning with a change. These changes occurred in a very short period and impacted the rest of the brokering situations that followed. By contrast, the brokering situations are longer periods of time wherein the previous changes are played out. For example, the whole project was on holdup at the beginning of the episode: "[…] In fact, I had a feeling that they were going in circles. Everyone was waiting for the other to do something again, so the circle closed at the back. This is what it looked like roughly." (Mary, citation 1).

The key informants reported that this standstill occurred because of the missing shared understanding between the initially involved business expert (Emily) and the DHW professional from Infocom (Mark):

> *Well the problems were that, despite the fact that we were in the same bank and were of the same profession, we all have our own terminology, our own way of talking, our – bigger problem is until the way to talk about something is harmonized. What that word means to us, what it means to them, and so on. That is the first wall that needs to be broken down.* (Lillian, citation 1)

On the one hand, Mark and others reported that Emily lacked the ability to analyze the delivered reporting results from the DWH and had refused any additional help. We set Emily's "knowledge of source data analysis" in Figure 4, therefore, to "none" (and all the other IT-related knowledge). On the other hand, Mark stated that he did not have the overall knowledge of the business experts' business and was unable to find business-related mistakes on his own (Figure 4, Mark's "business domain knowledge" set to "basic"). The overlap of the amounts of different knowledge types and the resulting common knowledge between Emily and Mark was very small, resulting in very high complexity on the boundary between them (pragmatic knowledge boundary). Additionally, the used boundary objects were written, descriptive business requirements specifications that Emily gave to Mark (syntactic boundary object). In this light, it is understandable as to why neither Mark nor Emily had gained significant new knowledge up to this point. The missing shared understanding and low common knowledge between Mark and Emily resulted in a first dialectical conflict (Figure 4, brokering situation between states S_0 and S_1).

The crisis began slowly to be resolved when a new business expert, Mary, joined the project and replaced Emily (synthesis; Figure 4: change of project participants in state S_1):

> *Everyone needs to step into everyone else's territory. So for example, with Mary it was easy to communicate—since she had an IT-related background and she knew, if necessary, to write a SELECT* [an SQL command]*, so she was easy to talk to. While before, at the beginning of the project, Mary was not involved at all. There was a different person* [Emily]*, a leader, who utterly refused "I don't know anything. I am not interested in anything. Why would I need this?" And then it's very difficult.* (Mark, citation 1)

Mary had some rudimentary IT knowledge due to her previous high school education, but her preliminary IT knowledge that was relevant for Project Omega was reportedly not very different from Emily's (Figure 4: participants' knowledge at the beginning of the episode). However, Mary succeeded in developing analytical skills in the following brokering situation, thus creating shared understanding between her and Mark and subsequently increasing her amount of knowledge of DWH development and of source data. This was something that her colleague Emily had never achieved.

Mary reported to have initially inherited "the old way" of how the requirements were gathered on the project: *"*From the beginning [...] as was the way I inherited, that was the way we worked [...] We say what we want in a descriptive way. Then Mark would communicate with Softcom then they agreed on the technical part*"* (Mary, citation 2).

Because Mark did not have the overall business knowledge of the business experts and Mary had just joined the team, the complexity on the boundary (cf. Figure 4, brokering situation between states

S_1 and S_2) was initially too high for them to use descriptive requirements specifications (syntactic boundary object). We display the mismatch as the second dialectical conflict (Figure 4, brokering situation between states S_1 and S_2). By switching the boundary object to paper drawings (pragmatic boundary object), Mark successfully introduced Mary to the source systems in the bank and gave her a rough overview of how the DWH system extracts the source data. This, as a result, created an initial shared understanding (Figure 4, switch between descriptive requirements and paper drawings in state S_2):

> *I said: "you know what, you have to spend a little time. I will come to you. Give me an hour, draw for me how it functions in the first place." Because when, in the end, he gets that report, I wanted to know which data of those came in directly from the source through the DWH, and which data was a result of the application's work. Because when something is wrong and something needs to be developed,* [it's important] *to know whom to contact: either the DWH team, the sources, or* [...] *And that's where the intense collaboration started. When we drew that up, when we got that scheme, drawing and when after that we sat down and started organizing the code lists, that's where the whole thing actually picked up.* (Mary, citation 3)

In the subsequent brokering situation (Figure 4, brokering situation between states S_2 and S_3), Mary and Mark discussed using the paper drawings (pragmatic boundary object), which was adequate for the level of complexity at the boundary between both. After the brokering situation, the total amount of common knowledge between Mary as the "business expert broker" and Mark as the "DWH professional broker" had increased. This enabled them to move forward faster with defining the requirements specification. In other words, both the change of the project participant in S_1 and the change of the boundary object in S_2 enabled the building of shared understanding between participants. This made both Mary and Mark brokers engaging in knowledge sharing, and resulted in an increase in common knowledge. We propose that both the change of the project participant and the switch to an adequate boundary object influenced the emergence of shared understanding and the sharing of knowledge in the project positively. If the change of the boundary object had never occurred, participants would have needed to invest more time to reach the goal.

The complexity of the circumstances on the boundary decreased due to the increase in the amount of common knowledge. Mary and Mark were now able to together discuss further issues and share knowledge using other boundary objects with less capacity (e.g., definition of requirements in the "descriptive way"). Even then, Mary and Mark could have continued using the paper drawings (pragmatic boundary object). However, Mary now decided to define new reports in the descriptive form (syntactic boundary object). This decision to change back to the syntactic boundary object reduced the time that the DWH professionals from Infocom needed to spend for collecting the business experts' requirements. We display this switch in Figure 4 not as a crisis, but as an acceleration towards the goal (teleological motor of change; Figure 4, S_2 and S_3).

In the brokering situation that occurred between the states S_3 and S_4, all three involved project participants met to define the requirements specifications using two different syntactic boundary objects. Mary defined new reports in a descriptive form ("descriptive requirements specification"). Mark successfully interpreted and translated the requirements in a form that the other DWH professionals from Softcom (Joe, Figure 4) were able to understand ("technical definitions of new reports"; see also Mary, citation 2). We categorized this brokering situation as part of the teleological process.

Despite the sound organization of the project, which resulted in the increase of shared understanding, sharing of knowledge, and a higher amount of common knowledge (Figure 4, brokering situation S_3-S_4), the project managers decided to integrate the two boundary objects (descriptive and technical requirements specification) into a single document. The supposed purpose of the new, joined boundary object ("Excel document") is stated to have directly connected the business experts with Joe (DWH professional from Softcom), without the interaction from Mark, who was needed for other tasks. In Figure 4, Mark's status stays "active", but he does not perform brokering between Mary and Joe (the gray shading between S_4-S_6). In the eyes of management, this decision should have brought

an acceleration towards the goal.

We group the change of boundary objects, Mark's exit, and the brokering situation that followed (Figure 4, brokering situation between states S_4 and S_5) as a dialectical motor of change. The analysis of the interviews revealed that the first versions of the Excel document caused only misunderstandings between Mary and Softcom developers:

> *It's just a thing of format and basically how detailed is that specification, in what way it is described. We give them* [DWH professionals from Softcom] *the specification. They go through it. But after that: "let's sit down together. Let's go through it together. Is there anything that isn't clear to you?" Because we had situations, that some things weren't interpreted correctly. Then we thought, everything was great, crystal clear. And then, when the time came they do their development, it was all wonderful. It comes to us for testing, and it wasn't that at all. Let's return it.* (Mary, citation 4)

The reason for the misunderstandings was due to the document's capacity. On the one hand, the new Excel document (Figure 4, syntactic boundary object in S_4) would have been adequate for the (syntactic) knowledge boundary between Mary and Mark. On the other hand, when Mark stopped his brokering activities and when the same boundary object was used on the other (pragmatic) boundary between Mary and Joe, Joe was not able to interpret the business experts' requirements written in the document (Figure 4, brokering situation in S_4-S_5). This is explained by the fact that the complexity on the boundary was too high for the boundary object being used. The result was again a holdup for the project: the shared understanding between participants and the amount of common knowledge were reduced drastically. Joe struggled hard with interpreting the requirements contained in the Excel document, only very slowly increasing his understanding of the business and the common knowledge between him and Mary. We display the result of this change of both project participants and used boundary objects with a decrease of shared understanding and common knowledge in the transition from state S_4 to S_5 compared with the S_3-S_4 transition (Figure 4)—another dialectical motor, with the conflict between the complexity on the boundary between project participants and the capacity of the used boundary object.

Mary was only able to realize that the problem existed at all when she was confronted with a prototype (pragmatic boundary object) in the following brokering situation (Mary, citation 4; Figure 4, brokering situation between S_5 and S_6). However, although the boundary object being used in this brokering situation to resolve the misunderstandings was adequate (pragmatic boundary object), the gap in shared understanding and common knowledge turned out to be too large to be closed in the time calculated for this project phase (slow increase of common knowledge in S_5-S_6; cf. Figure 4) Informants mention that this deceleration meant that they would not have been able to finish the project in time. Project management, therefore, at last, decided to bring Mark back. Mark's further involvement had a positive influence on bridging the gap in shared understanding and reducing the gap in common knowledge between Mary and DWH professionals from Softcom. Mark extended the Excel document with additional descriptive columns and SQL statements, turning it into a semantic boundary object between him and Joe. The resulting version of the document further helped Joe to increase his understanding of the customer business. This resulted in an acceleration of the process (Figure 4, transition from S_6 on):

> *We decreased the risk in a sense, through the manner of writing those specifications, for them* [Softcom] *not to take the wrong data… we managed to have Mark write something in the final column, the technical part. Like up to now, he works on the queries for the base, with some SQL. I mean with that IT language, they* [Softcom] *would get the data out.* (Mary, citation 5)

6. Discussion

The building of shared understanding is known to be a key determinant of successful IS development and a prerequisite for sharing knowledge in IS development projects (Hirschheim & Klein, 1989; Siau et al., 2010; Tan, 1994; Tiwana & McLean, 2003). The phase of requirements elicitation is of the highest importance for building shared understanding and sharing knowledge because the goal is to

communicate, negotiate, and specify requirements (Abran et al., 2004; Alvarez & Urla, 2002; Ambriola & Gervasi, 2006; Chakraborty et al., 2010; Cronan & Means, 1984; Gallivan & Keil, 2003; Levina, 2005; Mathiassen et al., 2007). Notwithstanding this importance, the understanding of how project participants from different communities of practice align their different frames of references and interpretative schemes is still limited. We argue that successfully sharing knowledge in case of novel entities and concepts involves approximate alignment at the level of the situation models (Pickering & Garrod, 2004) in the minds of project participants. We draw on Carlile's (2004) framework to characterize knowledge boundaries by different degrees of complexity. At a given boundary with either syntactic, semantic, or pragmatic complexity, boundary objects with given capacity, and the ability of brokers as actors, determine the capability to manage knowledge between communities of practice.

Nevertheless, our findings show that neither the characteristics of the boundary objects nor the sets of actors and brokers are fixed in time; actors are dynamic in the sense that their abilities can develop and both actors and boundary objects can be changed or replaced. Researching only one or the other as determining factors cannot explain the dynamic process of emerging shared understanding and knowledge sharing. We introduced, therefore, the concept of brokering situation as a lens to analyze the interplay of actors and boundary objects as a whole. The dynamics of actors' knowledge types, the degree of common and shared knowledge between actors, the type of boundaries faced by actors of different communities of practice, and the capacity of boundary objects-in-use characterize brokering situations. So far, scant research has investigated the interplay of brokers and boundary objects or the social dynamics associated with the change or replacement of either brokers or boundary objects (Argote & Miron-Spektor, 2011; Gal et al., 2008; Kimble et al., 2010; Subrahmanian et al., 2003).

We then introduced the boundary interaction framework to analyze further the dynamics of the building of shared understanding and knowledge sharing in time. The boundary interaction framework combines brokering situations with goal-driven (teleological) and conflict-driven (dialectical) motors of change (Van de Ven, 1992; Van de Ven and Poole, 1995; Van de Ven & Sun, 2011) to describe and explain how crises in requirements elicitation processes emerge and how they can be overcome. Using the boundary interaction framework, we describe the requirements elicitation process as sequences of brokering situations, which are separated by specific events and motors of change that influence the characterizing components of brokering situations.

We demonstrate that the concept of brokering situations allows us to better analyze the interactions and interplay at knowledge boundaries than the concepts of brokers and boundary objects alone. By this, we contribute to the literature on brokering and knowledge sharing in requirements elicitation. Specifically, we show that the higher the shared understanding and the common knowledge between different project participants and, at the same time, the better the fit between a given knowledge boundary and used boundary objects, the better the capability to share knowledge in a brokering situation. Indeed, capability equals capacity times ability (Carlile, 2004, p. 565).

Every brokering situation is an encounter that provides an opportunity for a change of the relationship between the project participants (Newman & Robey, 1992, p. 254). Until now, it has been problematic to predict when and how actors will use encounters to modify social structure (Newman & Robey, 1992; Orlikowski & Robey, 1991). The boundary interaction framework (consisting of the concept of brokering situations and the motors of change) allows us to describe, explain, and predict when and how project participants use brokering situations to modify social structure. This can be, for instance, when crises arise due to mismatches in abilities of participants or capacities of boundary objects (dialectical motor of change), when project participants want to speed up the whole process to reach the goal (teleological motor of change), or by changing the composition of the brokering situation (participants and/or boundary objects). These insights help to analyze each brokering situation and to take the appropriate choice of fitting boundary objects and project participants that are able to act as brokers in the given setting (Gopal & Gosain, 2009).

Carlile (2004) only suggests that the problems on a knowledge boundary with a low level of shared understanding and low amount of common knowledge between participants can be resolved by an investment of time in the discussion over a boundary object with more capacity. An important practical question is, however: what happens on such pragmatic boundaries if there is no *time* available for

discussions over a pragmatic boundary object because of project deadlines? We show that the involvement of additional project participants—"special" brokers or experts in a field—can resolve such problems as well. For example, the knowledge gap between the "designated" brokers Mark and Emily in Project Omega was too large for Mark alone to close in the given timeframe of the project (see Section 5.3). Mark alone could not act as a broker "in-practice"; he needed Mary (and her knowledge) to replace Emily and time for the emergence of shared understanding and enough common knowledge for knowledge sharing. Both Mary and Mark together then became brokers "in-practice" (in the sense of Levina and Vaast's (2005) boundary spanners-in-practice). Changing one of the two input parameters (here: project participants or used boundary object) can resolve crises and accelerate the process. Accordingly, we extend Carlile's (2004) framework and include explicitly the dimension of time as relevant for determining the total capability to manage knowledge:

$$\text{capacity in time x ability in time} = \text{capability in time} \qquad (2)$$

Hence, we argue that the capability to manage and share knowledge between different project participants represents a function of composition of project participants' knowledge and used boundary objects in time.

> **Proposition 1:** *Brokering situations with low levels of shared understanding and low amounts of common knowledge between project participants lead to crises in time-restricted requirements elicitation phases. The involvement of a knowledgeable broker in brokering situations that follow is very important in order to resolve equivocations or dissensions (as causes of the crises) during the project run-time.*

Sometimes, however, the boundary objects or project participants are not changed due to conflict or crises (dialectical motor of change), but in the interest of accelerating toward the goal (teleological motor of change). For example, Mark and Mary initially used paper drawings as pragmatic boundary objects (Section 5.3; Figure 4). Later, the used boundary object possessed more capacity than the complexity on the boundary required because the amount of common knowledge had increased. Mark's and Mary's decision to change back to a syntactic boundary object reduced the time that they needed to spend for eliciting the requirements. In such cases, the change of boundary objects is not a signal for a dialectical motor of change, but for a teleological one.

> **Proposition 2:** *If project participants in a brokering situation use boundary objects with high capacity at boundaries with low complexity, the change of boundary objects in encounters that follow occurs in order to accelerate the requirements elicitation process and finish the project on time.*

Although Carlile (2004) argues that boundary objects with less capacity should be used at the syntactic boundary, he did not exclude the use of boundary objects with more capacity. If this is the case, however, we suggest that the project participants do not use boundary objects for the purpose of knowledge sharing, but for the purpose of knowledge codification as explicit, formal documentation. In that case, participants try to preserve knowledge in case that any knowledgeable actor happens to leave the project.

In sum, we contribute to research with an improved understanding of the dynamics in knowledge sharing during requirements elicitation processes by introducing and applying the boundary interaction framework as a suitable analytical instrument. A misalignment can develop between the complexity of the knowledge boundary among project participants and the capacity of the used boundary objects. When the used boundary objects are changed, capacity can change. Likewise, when project participants change, abilities can change. Both changes can disrupt a brokering situation. As our findings show, such misalignments can cause severe misunderstandings between participants, with subsequent breakdowns in knowledge sharing. In a brokering situation, both the composition of project participants (i.e., the community representatives and their knowledge types) and the boundary objects they use influence how fast shared understanding emerges, how a better fit between common knowledge is created, and how the capability to share knowledge arises. The two

motors of change explain what impacts on capacity and ability. We need to keep those motors of change in mind and, if possible and necessary, try to leverage and steer them, supporting a fitting combination of project participants and boundary objects for a given situation. The boundary interaction framework can be used to carefully observe and describe the interplay of brokers and boundary objects. Hence, we advance the emerging "science of design requirements engineering" (Jarke & Lyytinen, 2010, p. 123) by systematically analyzing requirements elicitation processes and by grounding our concepts in empirical findings.

As contributions to practice, the boundary interaction framework might help to better understand current situations of requirements elicitation in IS development projects. The framework and its application could help to control for the dynamics of IS development projects better, which, we hope, will lead to better decision-making in development projects. For example, analyzing a chain of brokering situations offers a chance for a detailed audit of all project participants and their knowledge types, all used boundary objects and their capacity, and all changes of boundary objects or project participants in time due to dialectical and teleological motors of change. The boundary interaction framework provides a holistic view of how project participants actually can become brokers "in-practice", how they interact with each other, and how this interaction and the use of boundary objects drive the process in time. This should benefit practitioners by helping them to build a better shared understanding of requirements more quickly because "[the] more deeply project managers understand these behavioral processes, the greater their insight into the factors that determine their success" (Curtis et al., 1988, p. 1284). Our study offers clear patterns for the reasons behind the problems of building shared understanding and sharing knowledge. We also provide a systematic way to analyze and trace the problems' emergence. For example, an introductory session on "brokering situations" and "motors of change" could be a topic for a training unit for all project stakeholders. The attention of training participants could be drawn to the causes for conflicts and breakdowns in brokering situations, on the dynamics resulting from the motors of change, and on their own role concerning the improvement of knowledge sharing. We suggest that sensitizing project stakeholders about possible sources of deficiencies in brokering situations can influence the outcome of requirements elicitation positively.

7. Conclusion

The IS development process should not be viewed as a black box, but instead as a process of creation, innovation, knowledge sharing, and learning (Siau et al., 2010). The requirements generated during this process from communication and collaboration between diverse project participants are a major determinant of IS development success, and refocusing development methods from pure engineering to socially centered methods seems appropriate (Sawyer & Guinan, 1998). The boundary interaction framework provides a sharp lens for analyzing the interplay of boundary objects and brokers in time, and their relations with regard to knowledge sharing during requirements elicitation. Observing the complexity on the knowledge boundary between project participants and the used boundary objects in a sequence of brokering situations, driven by dialectical and teleological motors of change, allows us to analyze and explain how brokers and boundary objects interact, why equivocation or dissension episodes occur, and how these crises are resolved in time.

Our study has several limitations. For instance, we used only interviews, and did not include direct observations. Explanatory power of our findings is restricted to the DWH domain and closely linked to the data sample. The study itself, although illustrative, does not in any way test the discovered findings. In order to alleviate these issues, we plan to conduct more intensive case studies and field experiments in several other domains and also to contrast different approaches to IS development (e.g., plan-driven versus agile development). Moreover, in contrast to other studies investigating, for instance, politics and power in IS development (e.g., Davidson, 2002; Hussain & Cornelius, 2009), control (e.g., Chua, Lim, Soh, & Sia, 2012; Cram & Brohman, 2012; Kirsch, Sambamurthy, Ko, & Purvis, 2002), or the formulation and transmission of information from users to developers (e.g., Appan & Browne, 2012; Boland & Tenkasi, 1995; Gallivan & Keil, 2003; Metiu & Rothbard, 2013; Strode, Huff, Hope, & Link, 2012), our research excludes several of these aspects of social interaction. As a result, our study is limited to the interplay of project participants and boundary objects that affect the emergence of shared understanding and the sharing of knowledge. Our analysis is limited to the signification dimension of social structure and human interaction. Future research should broaden the

understanding of development processes by considering the domination and legitimation dimensions as well, including power, culture, or norms.

Further studies can build on our insights to design and provide effective tools, techniques, and elicitation methods that support choosing adequate brokers and boundary objects, and matching capacities and abilities for successfully building shared understanding and sharing knowledge. Future research might also use the boundary interaction framework to analyze IS development processes based on recent research on small groups (Arrow, McGrath, & Berdahl, 2000). Instead of analyzing the causal effects of single and independent variables, brokering situations, in combination with the motors of change, enable a more holistic description of IS development projects, which are all unique if we take into consideration their immanent complexity.

Acknowledgements

References

Abran, A., Moore, J. W., Bourque, P., & Dupuis, R. (Eds.). (2004). *Guide to the software engineering body of knowledge.* Los Alamitos, CA: IEEE.

Aldrich, H., & Herker, D. (1977). Boundary spanning roles and organization structure. *Academy of Management Review, 2*(2), 217-230.

Alvarez, R., & Urla, J. (2002). Tell me a good story: Using narrative analysis to examine information requirements interviews during an ERP implementation. *The DATA BASE for Advances in Information Systems, 33*(1), 38-52.

Ambriola, V., & Gervasi, V. (2006). On the systematic analysis of natural language requirements with CIRCE. *Automated Software Engineering, 13*(1), 107-167.

Ancona, D. G., & Caldwell, D. F. (1992). Demography and design: Predictors of new product team performance. *Organization Science, 3*(3), 321-341.

Ancona, D. G., & Caldwell, D. F. (1998). Rethinking team composition from the outside in. In D. H. Gruenfield (Ed.), *Research on managing groups' and teams' composition* (Vol. 1, pp. 21–37). Stamford, CT: Jai Press.

Appan, R., & Browne, G. J. (2012). THE impact of analyst-induced misinformation on the requirements elicitation process. *MIS Quarterly, 36*(1), 85-106.

Argote, L., McEvily, B., & Reagans, R. (2003). Managing knowledge in organizations: An integrative framework and review of emerging themes. *Management science, 49*(4), 571-582.

Argote, L., & Miron-Spektor, E. (2011). Organizational learning: From experience to knowledge. *Organization Science, 22*(5), 1123-1137.

Arrow, H., McGrath, J. E., & Berdahl, J. L. (2000). *Small groups as complex systems*. Thousand Oaks, CA: Sage Publications.

Bergman, M., King, J. L., & Lyytinen, K. (2002). Large-scale requirements analysis revisited: The need for understanding the political ecology of requirements engineering. *Requirements Engineering, 7*(3), 152-171.

Bergman, M., Lyytinen, K., & Mark, G. (2007). Boundary objects in design: An ecological view of design artifacts. *Journal of the Association for Information Systems, 8*(11), 546-568.

Black, L. J., Carlile, P. R., & Repenning, N. P. (2004). A dynamic theory of expertise and occupational boundaries in new technology implementation: Building on Barley's study of CT scanning. *Administrative Science Quarterly, 49*(4), 572-607.

Boland, R. J., & Tenkasi, R. V. (1995). Perspective making and perspective taking in communities of knowing. *Organization Science, 6*(4), 350-372.

Boland, R. J. J. (1979). Control, causality, and information system requirements. *Accounting, Organization and Society, 4*(4), 259-272.

Bostrom, R., & Heinen, J. S. (1977). MIS problems and failures: A socio-technical perspective. *MIS Quarterly,* 1(3), 17-32.

Bostrom, R. P. (1989). Successful application of communication techniques to improve the systems development process. *Information & Management, 16*(5), 279-295.

Branigan, H. P., Pickering, M. J., McLean, J. F., & Cleland, A. A. (2007). Syntactic alignment and participant role in dialogue. *Cognition, 104*(2), 163-197.

Branigan, H. P., Pickering, M. J., Pearson, J., & McLean, J. F. (2010). Linguistic alignment between people and computers. *Journal of Pragmatics, 42*(9), 2355-2368.

Brennan, S. E., & Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22*(6), 1482-1493.

Bromme, R., Jucks, R., & Runde, A. (2005). Barriers and biases in computer-mediated expert-layperson-communication. In R. Bromme, F. W. Hesse, & H. Spada (Eds.) *Barriers and biases in computer-mediated knowledge communication (*Vol. 5, pp. 89-119). New York, NY: Springer.

Brown-Schmidt, S., & Tanenhaus, M. K. (2008). Real-time investigation of referential domains in unscripted conversation: A targeted language game approach. *Cognitive science, 32*(4), 643-684.

Brown, J. S., & Duguid, P. (1991). Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. *Organization Science, 2*(1), 40-57.

Brown, J. S., & Duguid, P. (1998). Organizing knowledge. *California Management Review, 40*(3), 90-111.

Brown, J. S., & Duguid, P. (2001). Knowledge and organization: A social-practice perspective. *Organization Science, 12*(2), 198-213.

Cabrera, A., & Cabrera, E. F. (2002). Knowledge-sharing dilemmas. *Organization Studies, 23*(5), 687-710.

Carlile, P. R. (2002). A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science, 13*(4), 442-455.

Carlile, P. R. (2004). Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization Science, 15*(5), 555-568.

Carlile, P. R., & Rebentisch, E. S. (2003). Into the black box: The knowledge transformation cycle. *Management Science, 49*(9), 1180-1195.

Chakraborty, S., Sarker, S., & Sarker, S. (2010). An exploration into the process of requirements elicitation: A grounded approach. *Journal of the Association for Information Systems, 11*(4), 212-249.

Chua, C., Lim, W. K., Soh, C., & Sia, S. K. (2012). Enacting clan control in complex IT projects: A social capital perspective. *Management Information Systems Quarterly, 36*(2), 577-600.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 127-149). Washington, DC: APA.

Clark, H. H., & Krych, M. A. (2004). Speaking while monitoring addressees for understanding. *Journal of Memory and Language, 50*(1), 62-81.

Cohen, S. G., & Gibson, C. B. (2003). In the beginning: Introduction and framework. In C. B. Gibson & S. G. Cohen (Eds.), *Virtual teams that work: Creating conditions for virtual team effectiveness* (pp. 1-14). San Francisco, CA: Jossey-Bass.

Corvera Charaf, M., Rosenkranz, C., & Holten, R. (2013). The emergence of shared understanding: Applying functional pragmatics to study the requirements development process. *Information Systems Journal, 23*(2), 115-135.

Cram, W. A., & Brohman, M. K. (2012). Controlling information systems development: A new typology for an evolving field. *Information Systems Journal, 23*(2), 137-154.

Cronan, T. P., & Means, T. L. (1984). System development: An empirical study of user communication. *SIGMIS Database, 15*(3), 25-33.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM, 31*(11), 1268-1287.

Davidson, E. J. (2002). Technology frames and framing: A socio-cognitive investigation of requirements determination. *MIS Quarterly, 26*(4), 329-358.

Davis, A., Dieste, O., Hickey, A., & Juristo, N. (2006). Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review. *Proceedings of the 14th IEEE International Requirements Engineering Conference, Minnesota, USA,* 179-188.

Davis, G. B. (1982). Strategies for information requirements determination. *IBM Systems Journal,* 21(1), 4-30.

Doolin, B., & McLeod, L. (2012). Sociomateriality and boundary objects in information systems development. *European Journal of Information Systems, 21*(5), 570-586.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review, 14*(4), 532-550.

Fahey, L., & Narayanan, V. K. (1989). Linking changes in revealed causal maps and environmental change: An empirical study. *Journal of Management Studies, 26*(4), 361-378.

Fisk, A., Berente, N., & Lyytinen, K. (2010). Boundary spanning competencies and information system development project success. *International Conference on Information Systems.*

Flanagan, J. C. (1954). The critical incident technique. *Psychological Bulletin, 51*(4), 327-358.

Fleming, L., & Waguespack, D. M. (2007). Brokerage, boundary spanning, and leadership in open innovation communities. *Organization Science, 18*(2), 165-180.

Gal, U., Lyytinen, K., & Yoo, Y. (2008). The dynamics of IT boundary objects, information infrastructures, and organisational identities: The introduction of 3D modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems, 17*(3), 290-304.

Gallivan, M. J., & Keil, M. (2003). The user-developer communication process: A critical case study. *Information Systems Journal, 13*(1), 37-68.

Garrety, K., & Badham, R. (2000). The politics of sociotechnical intervention: An interactionist view. *Technology Analysis & Strategic Management, 12*(1), 103-118.

Garrod, S., & Pickering, M. J. (2009). Joint action, interactive alignment, and dialog. *Topics in Cognitive Science, 1*(2), 292-304.

Gasson, S. (1999). A social action model of situated information systems design. *The DATA BASE for Advances in Information Systems, 30*(2), 82-97.

Giddens, A. (1984). *The constitution of society: Outline of theory of structuration*. Berkley, CA: University of California Press.

Gopal, A., & Gosain, S. (2009). The role of organizational controls and boundary spanning in software development outsourcing: Implications for project performance. *Information Systems Research, 20*(8), 1-23.

Greiner, L. (1972). Evolution and revolution as organizations grow. *Harvard Business Review, 50*(4), 37-46.

Guest, G., Bunce, A., & Johnson, L. (2006). How many interviews are enough? *Field Methods, 18*(1), 59-82.

Guinan, P., & Bostrom, R. P. (1986). Development of computer-based information systems: A communication framework. *SIGMIS Database, 17*(3), 3-16.

Guinan, P. J., & Scudder, J. N. (1989). Client-oriented interactional behaviors for professional-client settings. *Human Communication Research, 15*(3), 444-462.

Hansen, S., Berente, N., & Lyytinen, K. J. (2008). Requirements in the 21st century: Current practice & emerging trends. In K. J. Lyytinen, P. Loucopoulos, J. Mylopoulos, & W. Robinson (Eds.), *Design requirements engineering: A ten-year perspective* (pp. 44-87). Berlin, Germany: Springer.

Hansen, S., & Lyytinen, K. (2009). Distributed cognition in the management of design requirements. *Perspectives Workshop: Science of Design: High-Impact Requirements for Software-Intensive Systems, Dagstuhl, Germany.*

Hansen, S., & Lyytinen, K. (2010). Challenges in contemporary requirements practice. *Proceedings of the 43th Hawaii International Conference on System Sciences, Koloa, HI, USA.*

Hargadon, A. B. (1998). Firms as knowledge brokers. *California Management Review, 40*(3), 209-227.

Haywood, S. L., Pickering, M. J., & Branigan, H. P. (2005). Do speakers avoid ambiguities during dialogue? *Psychological Science, 16*(5), 362-366.

He, J., Butler, B. S., & King, W. R. (2007). Team cognition: Development and evolution in software project teams. *Journal of Management Information Systems, 24*(2), 261-292.

Henderson, K. (1991). Flexible sketches and inflexible data bases: Visual communication, conscription devices, and boundary objects in design engineering. *Science, Technology & Human Values, 16*(4), 448-473.

Hirschheim, R., & Klein, H. K. (1989). Four paradigms of information systems development. *Communications of the ACM, 32*(10), 1199-1216.

Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software, 18*(4), 58-66.

Holten, R., & Rosenkranz, C. (2011). Designing viable social systems: The role of linguistic communication for self-organization. *Kybernetes, 40*(3/4), 559-580.

Horton, K. S., & Wood-Harper, T. A. (2006). The shaping of IT trajectories: Evidence from the UK public sector. *European Journal of Information Systems, 15*(2), 214-224.

Hussain, Z. I., & Cornelius, N. (2009). The use of domination and legitimation in information systems implementation. *Information Systems Journal, 19*(2), 197-224.

Hwang, M. I., & Xu, H. (2008). A structural model of data warehousing success. *Journal of Computer Information Systems, 49*(1), 48-56.

Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., & Robinson, W. (2010). The brave new world of design requirements: Four key principles. In *Advanced Information Systems Engineering* (Vol. 6051, pp. 470-482). Berlin: Springer Berlin Heidelberg.

Jarke, M., & Lyytinen, K. (2010). High Impact Requirements Engineering. *WIRTSCHAFTSINFORMATIK, 52*(3), 115-116.

Jarke, M., Lyytinen, K., Mylopoulos, J., Kappel, G., Leite, J., Mark, G., Ramesh, B., Schmitz, D., & Sutcliffe, A. G. (2009). High-impact requirements for software-intensive systems: Seminar outlines and working group summaries. *Perspectives Workshop: Science of Design: High-Impact Requirements for Software-Intensive Systems, Dagstuhl, Germany, 2009*.

Kimble, C., Grenier, C., & Goglio-Primard, K. (2010). Innovation and knowledge sharing across professional boundaries: Political interplay between boundary objects and brokers. *International Journal of Information Management, 30*(5), 437-444.

Kirsch, L. J., Sambamurthy, V., Ko, D.-G., & Purvis, R. L. (2002). Controlling information systems development projects: The view from the client. *Management Science, 48*(4), 484-498.

Lee, C. P. (2007). Boundary negotiating artifacts: Unbinding the routine of boundary objects and embracing chaos in collaborative work. *Computer Supported Cooperative Work, 16*(3), 307-339.

Lee, A. S. (1989). A scientific methodology for MIS case studies. *MIS Quarterly, 13*(1), 32-50.

Leifer, R., & Delbecq, A. (1978). Organizational/environmental interchange: A model of boundary spanning activity. *Academy of Management Review, 3*(1), 40-50.

Levina, N. (2005). Collaborating on multiparty information systems development projects: A collective reflection-in-action view. *Information Systems Research, 16*(2), 109-130.

Levina, N., & Vaast, E. (2005). The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems. *MIS Quarterly, 29*(2), 335-363.

Litwak, E., & Hylton, L. F. (1962). Interorganizational analysis: A hypothesis on co-ordinating agencies. *Administrative Science Quarterly, 6*(4), 395-420.

March, S. T., & Hevner, A. R. (2007). Integrated decision support systems: A data warehousing perspective. *Decision Support Systems, 43*(3), 1031-1043.

Mathiassen, L., Tuunanen, T., Saarinen, T., & Rossi, M. (2007). A contingency model for requirements development. *Journal of the Association for Information Systems, 8*(11), 569 - 597.

McMaster, M., & Grinder, J. (1980). *Precision: A new approach to communication.* Beverly Hills, CA: Precision Models.

Menenti, L., Garrod, S. C., & Pickering, M. J. (2012). Towards a neural basis of interactive alignment in conversation. *Frontiers in Human Neuroscience, 6*, 1-9.

Metiu, A., & Rothbard, N. P. (2013). Task bubbles, artifacts, shared emotion, and mutual focus of attention: A comparative study of the microprocesses of group engagement. *Organization Science, 24*(2), 455-475.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: A sourcebook of new methods.* Beverly Hills, CA: Sage.

Newman, M., & Robey, D. (1992). A social process model of user-analyst relationships. *MIS Quarterly, 16*(2), 249-266.

Nelson, K. M., Nadkarni, S., Narayanan, V. K., & Ghods, M. (2000). Understanding software operations support expertise: A revealed causal mapping approach. *MIS Quarterly, 24*(3), 475-507.

Nicolini, D., Mengis, J., & Swan, J. (2011). Understanding the role of objects in cross-disciplinary collaboration. *Organization Science, 23*(3), 1-18.

Orlikowski, W. J., & Gash, D. C. (1994). Technological frames: Making sense of information technology in organizations. *ACM Transactions on Information Systems, 12*(2), 174-207.

Orlikowski, W. J., & Robey, D. (1991). Information technology and the structuring of organizations. *Information Systems Research, 2*(2), 143-169.

Pawlowski, S., & Robey, D. (2004). Bridging user organizations: Knowledge brokering and the work of information technology professionals. *MIS Quarterly, 28*(4), 645-672.

Pawlowski, S. D., Robey, D., & Raven, A. (2000). Supporting shared information systems: Boundary objects, communities, and brokering. *Proceedings of the International Conference on Information Systems, Brisbane, Australia,* 329-338.

Pee, L. G., Kankanhalli, A., & Kim, H.-W. (2010). Knowledge sharing in information systems development: A social interdependence perspective. *Journal of the Association for Information Systems,* 11(10), 550-575.

Pickering, M. J., & Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences, 27*(2), 169-226.

Rosenkranz, C., Corvera Charaf, M., & Holten, R. (2013). Language quality in requirements development: Tracing communication in the process of information systems development. *Journal of Information Technology, 28,* 198-223.

Sawyer, S., & Guinan, P. J. (1998). Software development: Processes and performance. *IBM Systems Journal, 37*(4), 552-569.

Sawyer, S., Guinan, P. J., & Cooprider, J. (2010). Social interactions of information systems development teams: A performance perspective. *Information Systems Journal, 20*(1), 81-107.

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication.* Urbana, IL: University of Illinois Press.

Siau, K., Long, Y., & Ling, M. (2010). Toward a unified model of information systems development success. *Journal of Database Management, 21*(1), 80-101.

Star, S. L., & Griesemer, J. (1989). Institutional ecology, "translation," and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907–1939. *Social Studies of Science, 19*(3), 387–420.

Stephens, G. J., Silbert, L. J., & Hasson, U. (2012). Speaker-listener neural coupling underlies successful communication. *Proceedings of the National Academy of Sciences, 107*(32), 14425-14430.

Strode, D. E., Huff, S. L., Hope, B., & Link, S. (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software, 85*(6), 1222-1238.

Subrahmanian, E., Monarch, I., Konda, S., Granger, H., Milliken, R., Westerberg, A., & The N-Dim Group. (2003). Boundary objects and prototypes at the interfaces of engineering design. *Computer Supported Cooperative Work, 12*(2), 185-203.

Tan, M. (1994). Establishing mutual understanding in systems design: An empirical study. *Journal of Management Information Systems, 10*(4), 159-182.

Tiwana, A., & McLean, E. R. (2003). Expertise integration and creativity in information systems development. *Journal of Management Information Systems, 22*(1), 13-43.

Tsoukas, H. (2009). A dialogical approach to the creation of new knowledge in organizations. *Organization Science, 20*(6), 941-957.

Van de Ven, A. H. (1992). Suggestions for studying strategy process: A research note. *Strategic Management Journal, 13*(S1), 169-188.

Van de Ven, A. H., & Poole, M. S. (1995). Explaining development and change in organizations. *The Academy of Management Review, 20*(3), 510-540.

Van de Ven, A. H., & Sun, K. (2011). Breakdowns in implementing models of organization change. *Academy of Management Perspectives, 25*(3), 58-74.

Watson, H. J., & Wixom, B. H. (2007). The current state of business intelligence. *Computer, 40*(9), 96-99.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge, MA: Cambridge University Press.

White, K. B., & Leifer, R. (1986). Information systems development success: Perspectives from project team participants. *MIS Quarterly, 10*(3), 215-223.

Wixom, B. H., & Watson, H. J. (2001). An empirical investigation of the factors affecting data warehousing success. *MIS Quarterly, 25*(1), 17-41.

Yakura, E. K. (2002). Charting time: Timelines as temporal boundary objects. *Academy of Management Journal, 45*(5), 956-970.

Yan, A., & Louis, M. R. (1999). The migration of organizational functions to the work unit level: Buffering, spanning, and bringing up boundaries. *Human Relations, 52*(1), 25-47.

Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Thousand Oaks, CA: Sage.

Appendices

Appendix A: Attributes Used to Assess the Study

The study was informed by case study research. The quality of the selected research design can therefore be judged according to the following logical tests for construct validity, internal validity, external validity, and reliability (cf. Lee, 1989; Yin, 2003, pp. 40-44 for the following points).

Construct Validity

We followed three tactics to increase construct validity. We used multiple sources of evidence (multiple key informants in multiple projects) and established a chain of evidence during data collection and analysis (with the help of MaxQDA as our project diary and the use of causal mapping). Furthermore, key informants reviewed draft reports of the analysis.

Internal Validity

Internal validity is a concern for qualitative studies, where an investigator is trying to determine whether pattern X led to pattern Y. Basically, a qualitative study involves an inference every time an event cannot be directly observed; thus, we inferred that a particular event resulted from some earlier occurrence, based on interview evidence. In our data analysis, such pattern matching became possible by linking the categories of the coding scheme to data from the project diary and by using causal mapping. This was supported by explicit explanation-building in the study analysis and narratives, cross-checked by using two coders in each phase who resolved their differences (inter-subjectivity of interpretation).

External Validity

We used replication logic in the setup of the interviews. Therefore, we explicitly chose the study design to ensure analytical generalization across data warehouse development projects (triangulation across data).

Reliability

One prerequisite for allowing other investigators to repeat an earlier study is the need to document the procedures followed in the earlier case. For each interview in this study, we collected transcripts and protocols from the interviews. Furthermore, we developed the project diary and used MaxQDA for collecting all relevant data, giving appropriate samples of quotes in the narrative and linking our analysis to the data. This data is available from the authors on request.

Appendix B: Data Collection and Analysis

We collected and analyzed our data in parallel. We analyzed and discussed the collected data already in-between the three phases of interviews. Two researchers developed the categorical codes during the analysis of the interview data (Miles & Huberman, 1994, pp. 56-67). The initial coding scheme was based on our conceptual framework of brokering situations, which we continually changed and refined during the analysis as new categories emerged. In the process of open coding during phase I and II, we analyzed the collected data for recurring patterns, ideas, and concepts. The process of data analysis was clustered in three parts and was supported by the software MaxQDA. Each of the two coding researchers reread and recoded the interview transcripts several times to uncover codes and to link empirical evidence to the codes. After each interview phase, we used regular face-to-face sessions between the two coders to discuss the interviews in order to resolve differences in interpretation and codes. Thereby, we compared new data with code categories that emerged from previous phases of data collection in order to spot potential contradictions and to extend the evolving coding scheme accordingly. Furthermore, the data collected in phase III confirmed the results from the analysis of the interviews in phase I and II, with no additional categories or concepts having occurred. Therefore, we jointly concluded that theoretical saturation had been reached. Coding examples are illustrated in Table B-1, and Table B-2 presents the final coding scheme.

We additionally analyzed the transcripts of each of the 15 interviewees in phase II using causal mapping (Fahey & Narayanan, 1989; Nelson, Nadkarni, Narayanan, & Ghods, 2000). Mapping examples are presented in Figure B-1. As specific cause-effect pairs began to surface in the coding process, we conducted clarifying discussions with the same informants via instant messaging and/or e-mail in order to clarify open issues. Both coders iteratively revised the cause-effects pairs until they determined that all relevant themes were reflected.

For the transcripts of each of the interviews in phase III and the detailed audit of these three projects, we also coded the changes in types and amounts of knowledge for each project participant and changes in shared understanding. For example, if an interviewee reported that a DWH professional had never worked in a business domain (amount of knowledge "none"), the text section was coded with subcategory "Has no experience with customer's business" in the category "(customer's) business domain knowledge". Later, if it was reported that the DWH professional increased their knowledge about the business, then the subcategory "Developed experience with customer's business" was used (amount of knowledge "low"). Further on, if the DWH professional was participating in later brokering situations, their knowledge about business was coded as "Has general experience with customer's business". This implies that the DWH professional possessed now a "medium" amount of knowledge. "High" amounts of knowledge were coded if an informant reported on the participant's general knowledge (e.g., "Has detailed experience with customer's business").

Table B-1. Sample of Coding of Interview Data		
Example quote (key phrase italic)	**Category/code**	**Description**
"I said: 'Maja, don't look now. Connect this, this and this'. I have been in IT for a long time so I helped her a lot, her and her colleagues and make one "universe", that's what it's called in Business Objects, which talks about manufacture orders. *So, the buyer would order something, for now that's a Manufacture order and we said: 'in the manufacture order you have: the buyer, the name, address... So he ordered something for our catalogue. Here is our catalogue'*…"	Participant's knowledge – BEDF broker's knowledge – BEDF type of business – production	Type of domain-specific knowledge of a business expert that resides with business experts in decision-making fields for production.
"In short... Maybe I am not the typical example. *Considering that I made the system, which is still in function. Then I didn't need the information system (explained)*... I designed it myself with my colleagues. I didn't need classic teams formed for myself, as I formed them after all."	Participant's knowledge – BEDF broker's knowledge – source data general overview – has source data general overview	Type of domain-specific knowledge of a business expert that resides with system professionals for operational source systems.
"*Now, how it looked exactly I can't remember, but it was definitely a list of sources, where the data for the reports for that group would come from, and what sorts of reports would be manufactured for them. That was that.*"	Boundary objects – syntactic - repositories – functional specification	Objects that allow to represent knowledge and to transfer knowledge across boundaries.
"…well, there was a pretty big mistake, as a result of our misunderstanding [gap] of the data from that table, let say, ..because we understood some things differently. We implemented literally the way they defined the calculations in the table. Only when they saw *an example applying a calculation from that table* [a report], they've complained: "well, we don't want this in that way. We want it the other way around". That was the first problem. Second problem was the fact that during our development, they've changed table definitions 4-5 times."	Boundary objects – pragmatic – prototype of new report	Objects that allow to represent, learn, and transform knowledge and to transfer, translate, and transform knowledge across boundaries.
"In principle, as we had people who understood the business side, *and I was more-or-less getting the hang of things, because I worked on reports in Konzum, so I learned what is what. What is this value, what is that value, account, invoices, etc*. And then when they showed us, everything was more or less clear, we just had to see what each report was used for and could it possibly be used for anything else, etc."	Shared understanding – developed shared understanding over time	Development of shared frames, overlap and commonality in meaning, beliefs, expectations, and perceptions.

Table B-2. Final Coding Scheme

Code System
    Brokering Situation
        Number of participants
            Number of BUSINESS EXPERT brokers
            Number of DWH brokers
            Number of SYSTEM PROFESSIONAL brokers
        Type of brokering situation
            Implementing ERP
            Discussing DWH architecture
            Reviewing functional specification
            Reviewing calculations
            Reviewing interface tables
            Reviewing old reports
            Pilot meeting
            Milestones Meetings
            Defining functional specification
            Prototype reviewing
        Frequency of brokering situation
            Rarely
            Regularly
        Participants' experience
            BUSINESS EXPERT broker's knowledge
                Change of BUSINESS EXPERT broker
                BUSINESS EXPERT type of business
                    Damage
                    Statistics
                    Trading
                    Logistic
                    Production
                    Accounting
                    Construction
                    Management
                    Prepaid
                    Postpaid
                    Fixed Line
                    Internet
                    Claims
                    Controlling
                    Finance
                    Sales
                    Marketing
              Source data general overview
                Develop source data overview over time
                Has source data overview
                Has no source data overview
                Has no source data general overview
                Has source data general overview
                Developed source data general overview over time
             Knowledge of source data analysis
                Has no general experience with analysis of source data
                Has general experience with analysis of source data
                Developed general experience with analysis of source data over time
                Has no experience with analysis of source data
                Has experience with analysis of source data
                Developed experience with analysis of source data over time
             Customer's business domain knowledge
                Has no general experience with customer's business
                Has general experience of customer's business
                Developed general experience with customer's business over time
                Has no experience with the customer's business
                Has experience with the customer's business
                Developed experience with customer's business over time
             Knowledge of DWH development
                Has no general experience with DWH development
                Has general experience with DWH development
                Develop general experience with DWH development over time

Table B-2. Final Coding Scheme (cont.)

SYSTEM PROFESSIONAL broker's knowledge
    Change of SYSTEM PROFESSIONAL broker
    SYSTEM PROFESSIONAL type of source system
        Outsourced source system
        In-house source system
    Source data general overview
        Has no source data overview
        Has source data overview
        Develop source data overview over time
        Has no source data general overview
        Has source data general overview
        Developed source data general overview over time
    Knowledge of source data analysis
        Has no general experience with analysis of source data
        Has general experience with analysis of source data
        Developed general experience with analysis of source data over time
        Has no experience with analysis of source data
        Has experience with analysis of source data
        Developed experience with analysis of source data over time
    Customer's business domain knowledge
        Has no general experience of customer's business
        Has general experience of customer's business
        Developed general experience with customer's business over time
        Has experience with the customer's business
        Has no experience with the customer's business
        Developed experience with customer's business over time
    Knowledge of DWH development
        Has no general experience with DWH development
        Has general experience with DWH development
        Develop general experience with DWH development over time
DWH PROFESSIONALK broker's knowledge
    Change of DWH broker
    DWH type of developers
        In-sourced
        Outsourced
        DWH technician (pure developers)
        DWH professionals
    Source data general overview
        Develop source data overview over time
        Has source data overview
        Has no source data overview
        Has no source data general overview
        Has source data general overview
        Developed source data general overview over time
    Knowledge of source data analysis
        Has no general experience with analysis of source data
        Has general experience with analysis of source data
        Developed general experience with analysis of source data over time
        Has no experience with analysis of source data
        Has experience with analysis of source data
        Developed experience with analysis of source data over time
    Customer's business domain knowledge
        Has general experience of customer's business
        Has no general experience of customer's business
        Developed general experience with customer's business over time
        Has no experience with the customer's business
        Has experience with the customer's business
        Developed experience with customer's business over time
    Knowledge of DWH development
        Has no general experience with DWH development
        Has general experience with DWH development
        Develop general experience with DWH development over time

Table B-2. Final Coding Scheme (cont.)

```
            Boundary objects
                BOs
                        Syntactic - Repositories
                                Email
                                Definition of new report (technical)
                                Definition of interface tables
                                Definition of new report
                                Software specification
                                Calculations
                                Database documentation
                                Old reports
                        Semantic
                                ETL mapping (code to map sources to targets)
                                Functional specification
                                Technical specification
                        Pragmatic
                                Prototype (new product, new reports)
                                Reporting methodology support-application
                                Paper drawing (editable by all participants)
                                Application screens
                        Change of BO
                        Change of BO (itself)
    Shared understanding
        Shared understanding
        Misunderstanding
                Boundary object with insufficient capacity
                Missing experience
                Deficiency of source data system (technical)
        Developed shared understanding over time
        Did not develop shared understanding over time
```
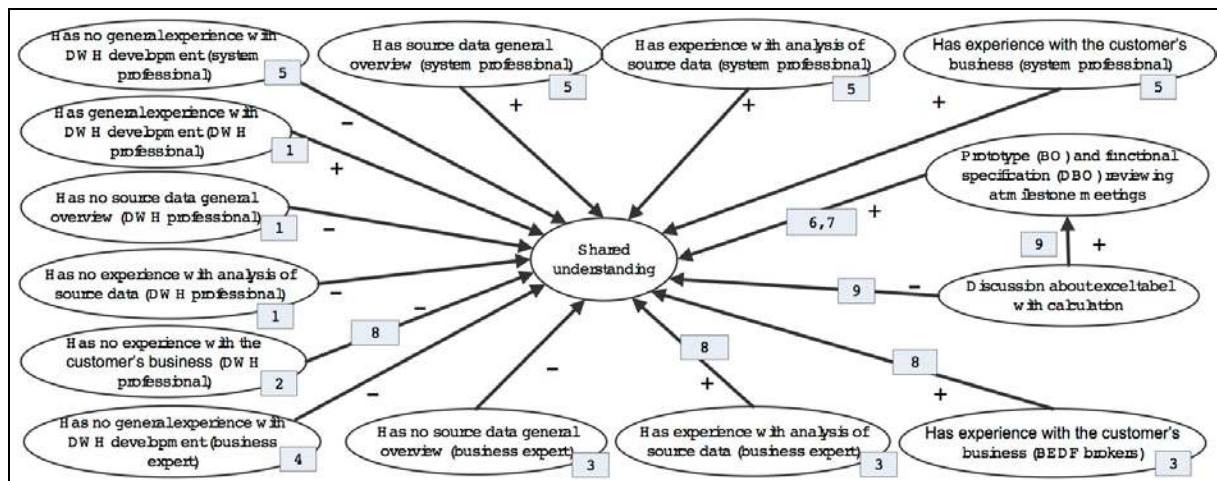

Figure B-1. Sample Consolidated Causal Map for Project Clover

About the Authors

Christoph ROSENKRANZ is Full Professor of Information Systems at the University of Cologne. Previously, he worked as an assistant professor at Goethe University Frankfurt. He holds Diploma and Ph.D. degrees from the University of Münster and Goethe University, respectively. His research focuses on integrated information systems and business process management, information systems development and IT project management, and online communities. He has published articles in such outlets as *ACM Transactions on Management Information Systems, Business & Information Systems Engineering, Information Systems Journal, International Journal of Web Based Communities*, *Journal of Database Management*, *Journal of Information Technology,* and *Supply Chain Management.*

Helena VRANEŠIĆ works as consultant for complex IT projects, mainly in the field of business intelligence systems. She holds a Ph.D. in Information Systems from Goethe University Frankfurt, Germany. Drawing on her ample experience in both fields, she can combine scientific rigor with practical relevance. Her research focuses on the dynamics within multi-discipline teams in complex IT projects, a topic on which she has authored several scientific papers in international conferences.

Roland HOLTEN is a Full Professor of Information Systems and holds the Chair for Information Systems Engineering at Goethe University Frankfurt, Germany. He received his Ph.D. in Information Systems from the University of Münster, Germany. His research interests include IT-mediated communication structures of groups, information systems development, and IS modeling. His work has been published in peer-reviewed journals such as *Journal of Information Technology*, *Information Systems Journal*, *Information Systems*, *Information Systems and e-Business Management*, and *Business & Information Systems Engineering*. He is member of the editorial board of *Information Systems and e-Business Management*.