ECIS 2010 Proceedings

European Conference on Information Systems (ECIS)

2010

Design Boundary Objects – Development Guidelines for Financial Data Warehouse Projects

Helena Vranesic
Goethe University, vranesic@wiwi.uni-frankfurt.de

Christoph Rosenkranz
Goethe University, rosenkranz@wiwi.uni-frankfurt.de

Marc Rakers
zeb/information.technology, marc@raekers.com

DESIGN BOUNDARY OBJECTS – DEVELOPMENT GUIDELINES FOR FINANCIAL DATA WAREHOUSE PROJECTS

Journal:	*18th European Conference on Information Systems*
Manuscript ID:	ECIS2010-0155.R1
Submission Type:	Research Paper
Keyword:	Business intelligence, Information requirements determination, Information system development, Systems analysis and design/development

DESIGN BOUNDARY OBJECTS – DEVELOPMENT GUIDE-LINES FOR FINANCIAL DATA WAREHOUSE PROJECTS

Vranesic, Helena, Goethe-University, Grüneburgplatz 1, 60326 Frankfurt am Main, Germany, vranesic@wiwi.uni-frankfurt.de

Rosenkranz, Christoph, Goethe-University, Grüneburgplatz 1, 60326 Frankfurt am Main, Germany, rosenkranz@wiwi.uni-frankfurt.de

Räkers, Marc, zeb/information.technology, Schlossstr. 22, 48455 Bad Bentheim, Germany, marc@raekers.com

Abstract

Financial data warehouses are an important tool for most financial services providers today since they have to deal with several business domains and different fields of knowledge. To address complexities in the context of design processes that involve multiple heterogeneous social worlds, Bergman et al. (2007) introduced the concept of design boundary objects with four essential features to improve and promote design conversation. Given the lack of methodology or guidelines that could help both researchers and practitioners understand how the four features are implemented and integrated in specific design boundary objects, we investigate the so-called Data Requirements Tool, an artefact for storing and monitoring negotiated requirements in financial data warehouse projects. We found that the implementation of four essential features into the tool was a repeating process, with shared representation as a basic feature of design boundary objects. As a result of our analysis we propose a set of guidelines for the better development of design boundary objects in complex settings such as financial data warehousing projects.

Keywords: financial data warehouse projects, data warehousing, information systems development, requirements analysis, requirements engineering, design boundary object, communities of practice, language communities.

1 INTRODUCTION

Because data warehouse (DWH) projects often fail or significantly exceed budgets, existing research has concentrated on quantitative or qualitative analyses of success factors as well as contemporary best practices for building DWH (e. g., Weir et al. 2003, Herrmann and Melchert 2004, p. 550, Hwang et al. 2004, Watson et al. 2004, Hwang and Xu 2007). Getting data into a DWH is the most challenging aspect of business intelligence, requiring about 80 percent of the time and effort and generating more than 50 percent of the unexpected project costs (Watson and Wixom 2007, p. 96). The challenge stems from multiple causes, such as poor data quality in the operational source systems, politics around data ownership and legacy technology. In this context March's and Hevner's (2007) thorough literature research warns how links between data warehousing, strategic decision-making and evaluation are under-researched.

The four overarching objectives for DWH support of management decision-making processes identified by March and Hevner (2007) – integration, implementation, intelligence and innovation – assume a high affiliation between the communities of practice participating in a DWH project. However, Rizzi et al (2006) note the absence of effective techniques (1) for collecting information needs and quality-of-service requirements and (2) for translating those requirements into conceptual models based on a common vocabulary between IT experts and decision-makers. Similarly, Jarke et al. (2009) argue that a deeper articulation of individual and groups requirements is missing in distributed projects. As a response to the increased distribution of requirements processes, Hansen and Lyytinen (2009) suggest a model to synthesize propagation between social and structural distributed cognition in contemporary requirements practice. However, their propositions only illustrate the current "state of affairs", but give no guidance for overcoming gaps and disagreements.

In the context of communities of practice, the creation and management of boundary objects has proved to be important in developing and maintaining coherence across intersecting social contexts (Star and Griesemer 1989, p. 393). In the specific context of DWH projects, the artefacts (e. g., shared documents, tools, business processes, objectives, schedules) exchanged between communities of practice can potentially become such boundary objects (Brown and Duguid 2001, p. 209). In order to actively participate in multi-stakeholder product design, such as the design of a DWH, boundary objects need to *represent*, *transform*, *mobilize*, and *legitimize* heterogeneous design knowledge between all participating communities (Bergman et al. 2007). Bergman et al. (2007) define objects embodying these four features as *design boundary objects* (DBO).

To counter weaknesses in requirements engineering, Bergman (2009, p. 405) suggests that future requirements documents need to be developed and deployed that, at minimum, conform to the four rules of an operational DBO. However, Bergman (2009, p. 405) recognizes missing arguments to support improvement of DBOs. Concrete guidelines for both researchers and practitioners for designing, implementing and using DBOs in specific domains and contexts are needed. Research questions for inquiry in this area include (Bergman 2009, p. 401):

- What common models, stories and formats are necessary to enable requirements to serve as DBOs?
- How well do DBOs represent problems-requirements-solutions design candidate combinations?
- How do the boundary objects evolve during DWH projects to be DBOs?
- How can the four features actually be integrated and implemented in DBOs used in DWH projects?

The remainder of the paper is structured as follows. In the next section, we review related work and theoretical foundations. Afterwards, we investigate the so-called "Data Requirements Tool", an IT artefact used for storing and monitoring negotiated requirements in DWH projects in the financial industry. We analyze and examine the tool in order to evaluate and test Bergman's et al. (2007, 2009) DBO design theory in the context of financial DWHs. Afterwards, we summarize our findings and limitations so far, proposing a set of guidelines for the better development of DBOs in financial DWH projects. Finally, we give an outlook on further research.

2 RELATED WORK AND THEORETICAL FOUNDATIONS

2.1 Communities of Practice in Data Warehouse Projects

Communities of practice are characterized as shared histories of learning (Wenger 1998, p. 86, Wenger et al. 2002). Such histories are known to create discontinuities between the ones who participate in the community work and the ones who do not (Wenger 1998, p. 103). For example, two different groups of people are usually involved in specifying, developing, and controlling requirements: users from the business units that employ information technology (IT) and developers from the IT departments that provide IT. One of the key success factors during information systems (IS) development relates to the bridging of the so-called communication gap between those two groups (Bostrom and Kaiser 1981, Al-Rawas and Easterbrook 1996, Peppard 2001). These discontinuities are also revealed in the development of DWHs. We distinguish two groups of participants which confront each other in DWH projects: (1) operative system professionals (OSPs) with knowledge of the legacy and source systems, and (2) business experts in decision-making fields (EDMFs). OSPs are in charge of maintenance and further development of operational source systems. Using reports created of data from the DWH, EDMFs make informed decisions in order to determine the course of action a company needs to take to stay competitive. The connection between these two practices is made by introducing a third practice, (3) DWH professionals, responsible for the development of the DWH.

2.2 Communities of Practice, Language Communities and Boundary Objects

According to several authors (Wenger 1998, p. 105, Brown and Duguid 2001, p. 209), there are two forms of boundary connections: (1) brokering and (2) boundary objects. For example, Pawlowski and Robey (2004) observed the role of IT professionals as knowledge brokers in organizations. By participating in bordering organisational units, they have the responsibilities to design, implement, and maintain shared IS which they then use as *boundary objects* to bridge traditional boundaries separating them (Pawlowski and Robey 2004, p. 662). Star and Griesemer (1989) define boundary objects as "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" detailed. Boundary objects may be abstract or concrete, are weakly structured in common use and become strongly structured in individual-site use. They can have different meanings in different social contexts (i. e., communities of practice).

However, according to Levina and Vaast (2005), for boundary objects to become *boundary objects-in-use*, their common identity is not sufficient to assure the coherence. Their appropriate use must be limited to the context of a joint practice (Levina and Vaast 2005, p. 341). Following Kamlah and Lorenzen (1984), these members form a language community. Kamlah and Lorenzen (1984) argue that language as a system of signs promotes mutual understanding as a "'know-how' held in common, the possession of a 'language community'" (p. 47). A new term is introduced by explicit agreement between language users with respect to its usage and meaning (Kamlah and Lorenzen 1984, p. 57). This agreement leads to a relation of concept and term, and is shared by a language community as the knowledge of using this term.

According to our understanding, boundary objects play a significant role in making language communities explicit: if members of communities of practice have the same concept in mind when being confronted with the same artefact (boundary object), they belong to the same language community. An individual has to revise his or her understanding of the given language signs with respect to the ones of the language community which constructs this language in order to become a member of it, and thus gain the possibility of using the same terminology in discourse with the other community members. In a new product development setting, the impact of new requirements on the common understanding between the participants is hard to determine (Carlile 2004, p. 557). We argue that only the boundary objects which provide effective means of common representation, that facilitate negotiation of common meaning and that thereby, if necessary, help to transform the common knowledge at the boundary, can successfully support a language community creation process.

2.3 The Role of Design Boundary Objects in Development Projects

After a thorough literature research on various adaptations of Star's and Griesemer's (1989) concept of boundary objects, Bergman et al. (2007) found that neither of the suggested objects contain features that actively support the design. By adapting Star's and Griesemeier's (1989) definition of boundary objects to the context of a new product design, Bergman et al. (2007) investigated the role that DBOs have in shaping and influencing dynamics of design ecology.

Bergman et al. (2002) define the *design ecology* as a context of product design with its functional and political elements, with DBOs as: "…any representational artefact that enables knowledge about a designed system, its design process, or its environment to be transferred between social worlds and that simultaneously facilitates the alignment of stakeholder interests populating these social worlds by reducing design knowledge gaps" (p. 551). In the context of product design, the *political ecology* comprises the actors holding organizational power and/or control over the project's resources (Bergman et al. 2007, p. 548). The *functional ecology* contrives operational parts of the design – it defines operational goals of the designed product and the technological means to meet these goals (Bergman et al. 2007, p. 549). These ecologies form a *duality*: the functional ecology informs the political ecology of the further feasible functional design operations while the political ecology determines which ones are most suited for the given design circumstances (Bergman et al. 2007, p. 548).

Due to the ambiguity within the shared product design and due to the probable stakeholders' disagreements, practical functional solutions and acceptable political solutions are usually not feasible in a single step (Bergman et al. 2007, p. 547). Therefore Bergman's et al. (2007) design path is composed of *routines* – stepwise sets of activities, rules, techniques and norms that support the processes of identifying problems, determining stakeholders' requirements and finding the solutions understandable to all the participants (e. g., requirement analysis, initial solution design, and so forth). After one routine is ended, the project stakeholders need to align both of their ecologies and set a new course of action. To actively support the design, for example by indicating "successful progress" or "impending issues" in the design, Bergman et al. (2007) argue, that DBOs need to possess four features: (1) *promote shared representation,* (2) *transform the design knowledge*, (3) *mobilize for design action*, and (4) *legitimize the design knowledge* (Bergman et al. 2007, p. 551). Ignoring the features could hinder the design or even lead to its collapse (Bergman et al. 2007, p. 547).

In conclusion, the use of DBOs as presented by Bergman et al. (2007) helps (1) to resolve the ambiguity and uncertainty associated with functional requirements and (2) to generate political momentum for choosing a design solution. However, Bergman's et al. (2007) research mainly concentrates on the features that boundary objects need to reflect in order to become DBOs; it gives no concrete guidance on how to implement these features into DBOs. Even in the outlook of his latter work, Bergman (2009) emphasizes the importance of finding ways to improve DBOs' properties, so that they could better reflect the features. Therefore we argue that a better understanding of how the four features can actually be implemented and integrated in specific DBOs should lead to a formation of better boundary objects facilitating a design. In the context of DWH projects, we argue that DBOs – IT artefacts – could help the involved communities of practice in DWH projects to transform their visions of solutions into representations that are understandable to all other involved stakeholders easier and faster. Basically, we suggest that DBOs supporting this process could support faster discovery of misunderstandings between the participants and thereby accelerate the process of creating a joint language community (Kamlah and Lorenzen 1984, p. 57) that afterwards enables the transfer of knowledge across boundaries between communities of practice. We propose that the creation of such a language community between DWH professionals and both EDMFs and OSPs in early stages of DWH development plays an important role for the success of the final product and should be considered as the first and most important step in the development of DWHs. Next, we examine the design of a specific DBO for supporting the development of financial DWHs.

3 A DBO FOR FINANCIAL DATA WAREHOUSE PROJECTS

DWHs play increasingly important roles in the IT landscape of the financial industry. For example, financial institutions increasingly have to fulfil extensive regulatory requirements such as Sarbanes-Oxley Act or Basel II. Banks are also required to conduct calculations that compute the institutions' whole portfolio and rely on group-wide data consolidation and integration. Therefore most of the leading banks implemented DWHs during the last decade to enhance decision-making and internal reporting, especially controlling and risk management. Yet many banks struggle with data integration of heterogeneous, disorganized or even inaccessible data sources. Moreover, financial DWH projects have to deal with several business domains and different fields of knowledge (Behrmann and Räkers 2008). Consequently, they are characterized by high semantic complexity, which increases the difficulties in reaching a shared understanding (Rosenkranz 2009, pp. 161-209).

No common standards for data integration exist until today in the financial industry. The specification of the semantics of a single data field in its context varies from bank to bank, from business unit to business unit and from department to department (Behrmann and Räkers 2008). This is a serious problem for financial DWH development: data integration fundamentally relies on a shared understanding and precise specification of data fields. However, a consistent and unitary description of all types of financial businesses is not a realistic option because financial institutions constantly update their business models by rapidly developing new products and services in order to gain a competitive advantage (Corcho et al. 2005). Consequently, semantic complexity and heterogeneity are likely to stay high in the financial industry. In order to support the negotiation of requirements for DWHs in these complex environments, the consultancy zeb/it, which focuses on IT in the financial industry, developed the so-called "Data Requirements Tool" (DRT). This tool was implemented in order to create "common ground" (Clark 1996) in financial DWH projects. The artefact consists of a data requirements format, a process description (method) and a corresponding software tool to document and support all important process steps (see screenshot of interface in Figure 1). Starting in 2005, the DRT has been implemented to improve and test the effects of changes. Since then, the DRT has successfully been used and iteratively refined in over fifteen large financial DWH projects and has become the standard approach for eliciting requirements in financial DWH projects of the consultancy company.

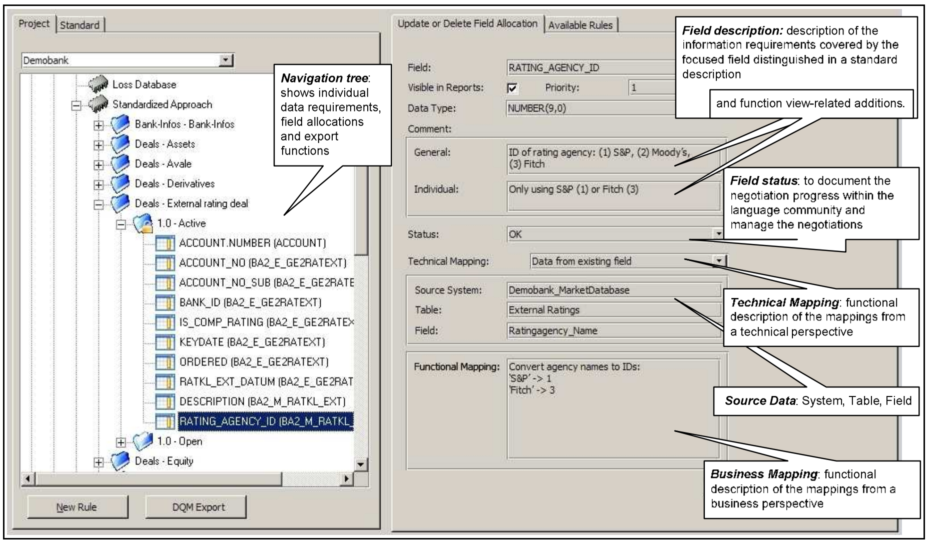

Figure 1. Screenshot of the Data Requirements Tool's Interface

In the next section, we analyze the DRT with regard to the four essential characteristic of a DBO according to Bergman et al. (2007). We show that the DRT exhibits all four characteristics, thereby corroborating Bergman's et al. (2007) findings. Furthermore, we deduce guidelines for the development of comparable DBOs in similarly complex settings.

4 EXAMINING THE DATA REQUIREMENTS TOOL

Bergman's et al. (2007, 2009) conceptualization of DBOs is basically an instantiation of a theory for design (Gregor 2006, p. 620). The corresponding set of characteristics was developed by Bergman et al. (2007) in the context of design processes that involve multiple heterogeneous social worlds (Bergman et al. 2007, p. 550). As argued above, financial DWH projects deal with several business domains and different fields of knowledge, making it therefore a complex design process. In other words, the DRT can be studied within the framework of Bergman's et al. (2007, 2009) theory. If the DRT is to be characterised as a DBO in the sense of Bergman et al. (2007, p. 547), it must posses and display four essential features.

To test the DRT with regard to those features, we started our analysis by examining and interpreting the data collected by the tool. In the absence of any clear "how-to-check-features" guidelines, we tried to find patterns within the raw data that matched the features' definitions. The procedure can be seen as a form of "coding" (Miles and Huberman 1994). This interpretation was done by the first author who subsequently discussed her interpretations with the other authors. Table 1 presents an excerpt of the data requirements together with identified patterns marked in the "Comment" column.

Description	***Datum***	***Status***	***Comment (T)***
IFR_CATEGORY			
IFRS category from the source systems. If this ID can be delivered, all following information is not required for automatic categorisation	15.01.2009	OPEN	"This field has to be discussed with Mr. Kim and Accounting." (M)
	19.01.2009	FUNCTIONAL2	"This information will be available." (L)
	20.01.2009	OPEN	"This field has to be checked by zeb/ for availability." (M)
	21.01.2009	OPEN	"zeb/ will check if a default value can be used." (M)
	23.01.2009	OPEN	"This information will be available by using a special mapping file. This information will not be available directly at each account. zeb/ will check if the information given in the mapping file suffice to do a mapping for each account." (M)
	27.01.2009	OPEN	"This field will be checked by zeb/. Mr. Smith sends a description of current category mappings to zeb/. Without using categorizer of zeb/ifrs only very simple mappings are in scope of the project." (M)
	05.02.2009	FUNCTIONAL2	"This info. will be available using a mapping." (L)
BANK_ID			
Unique identification of the bank. Necessary for multi-client capability. If there is only one credit institution, a constant number has to be delivered	21.08.2008	FUNCTIONAL2	"zeb/ clarifies if the BANK_ID is only one key for all of Bank A (which then would be fix 009)? Bank A is further divided into multiple different books which could be regarded as divisions of Bank A. zeb/ clarifies if the field BANK_ID can be filled simply with fix 009." (SR, M, L)
	10.09.2008	FUNCTIONAL2	"The info. about the book is given for each single customer, so it is available. zeb/ will decide on the design of the data retrieval." (M, L)
	29.09.2008	OK	"It is sufficient to always use the same fixed number 009 for this field." (L, SR)

Table 1. DRT's Data Sample - Test against DBO Features: Shared Representation (SR), Transformation (T), Mobilization (M), and Legalization (L)

4.1 Promote Shared Representation

The capability for *shared representation* assumes a common syntax and semantics between the participating communities of practice (Bergman 2009, p. 399). Bergman et al. (2007) see DBOs as "simple enough to be understandable" (e. g., proto-architectures) or as being defined by a user's standard (e. g., project plans that comply with standards). In such cases, mutual understanding is assumed, and the communication border between the participants is positioned on the syntactical level, where, according to Carlile (2004), a *common lexicon* suffices to specify the differences and dependencies of consequences at the boundary (Carlile 2004, p. 558). On this level, if communities seek better understanding of each other, they only need to communicate more (Carlile 2002, p. 444). However, the rather important point of creating this mutual understanding, having shared semantics and pragmatics (Carlile 2002, Carlile 2004) – "How do we create mutual understanding between participants?" – is not addressed by Bergman et al. (2007). Bergman (2009, p. 400) simply argues that the form and format of most requirements – logical text-based specifications or object-oriented (class or activity) models – tend to fail the four essential features of DBOs, as they are difficult to understand and resistant to change. In the examined DRT, the structured user interface that is shared between the participating communities was developed with much concern regarding its main role – to create a pragmatic "common ground" for all the participants. Firstly, the DRT was developed by DWH professionals with the main intent to help them shape so-called "connection solutions" between the source data models (SDM) and the target data models (TDM). Secondly, the DWH professionals also defined the initial TDM (see Description column in Table 1). Due to the complex environment of financial DWHs, the problem of mutual understanding is not just a question of syntax: to assume that all the stakeholders from the beginning of the DWH project have an aligned understanding of the DRT's interface syntax and TDM definitions is misleading.

Briefly stated, a semantic heterogeneity exists when data is defined differently by different users (March and Hevner 2007), that is often the case in financial DWHs. For example, what is the meaning of terms such as "limit", "bond" and so forth, and which data fields of what source system map to these meanings? Basically, the problem is not syntactically, but one of *pragmatically creating* a mutual understanding and of *semantically* describing the data fields. In this context, "mismatches" occur due to the misalignment between the type of boundary faced and the capacity of the knowledge process occurring on the boundary (Carlile 2004, p. 560). DWH professionals and OSPs will try to use the old syntax to transfer knowledge, but they actually need at first to transform and align the meanings of terms they use. As a solution for this problem, the DRT interface was iteratively developed in tight cooperation with the users, leading to redefinitions of the TDM attributes (i. e. see Table 1, redefinition of BANK_ID in Comment column). Thereby (1) all *the terms within the DRT's interface* as well as (2) *the SDM and TDM attributes managed by the DRT* are part of the same "common-representation space" between participating communities whose meaning needed to be aligned. Only then, when all the bordering communities share the same syntax and interpretations of each term within the DRT's interface and TDM attributes, accurate communication can be established, allowing an undisturbed information flow from the sender to the receiving side (Carlile 2002, p. 443). For example, a status field indicates a reached status of negotiation about certain connections and thereby embodies a shared political representation. Depending on whether the target attribute was successfully matched to the source or not, the connection's status flag changes (or stays the same). Therefore, we suggest as our first guideline that *the first step in the development of a DBO in complex domains such as financial DWH projects should be the creation of a language community between IT professionals and the DBO's users with shared understanding of the DBO interface*. The DRT as a DBO supports the development of a mutual understanding, focusing on *pragmatics in practice*. We assume that with the help of the shared DRT interface and TDM attributes, DWH professionals can expect the generated subset of connections to be a proper solution and not just giving the illusion of a solution. Then the DRT's shared syntax and the political representation promoting knowledge transformation and power alignment at the boundary reflect the tool's capability for common representation.

4.2 Transform Design Knowledge

Transformation defines a DBO's ability to manipulate and converse representations that will propel movement between design routines as to facilitate finding a feasible functional solution and stabilize the political ecology (Bergman 2009, p. 399). This includes the capability to transform domain-specific design knowledge (e. g., by using a common lexicon). This moves knowledge from ambiguous to specific and realigns the operational structure to stabilize the functional ecology, allowing traceability of the actual design and the agreements reached within the design process (Bergman 2009, p. 399). The DRT's interface provides a framework in which all stakeholders fill out the SDM attributes fields (e. g., source systems, data table and field names, and so forth) for every TDM attribute, thereby transforming their domain knowledge into the common syntax and semantics. Beyond that, the DRT provides the possibility for defining configurations of data quality rules. By doing so, the DRT enables mobilisation, transformation and acquisition of OSPs' knowledge as well as constraints of their requirements. Due to the versioning capabilities of the DRT, within each discussion between participants, the transformed knowledge of OSPs' can be further refined and saved (see Comment column in Table 1). With capabilities for ensuring traceability of agreements, supported by comment and flag interface fields, the DRT totally reflects the transformation feature of DBOs. However, as our second guideline, we argue that *a language community between participants has to be created before such a common lexicon can be used*. This includes the common understanding of a DBO's interface as well as SDM and TDM attributes. Their misinterpretations could hinder a transformation of the important domain knowledge. This became apparent in the first applications of the DRT in projects, which let to amendments of the interface.

4.3 Mobilize for Design Action

Mobilization describes the capability of a DBO to mobilize for action (Bergman 2009, p. 399). This sources and wields resources and power to propel the project progress along a design path. The main issue here is how well the DBO promotes the discovery of discrepancies across social worlds and provides mechanisms to address them (Bergman et al. 2007, p. 561). However, again we argue that if a lexicon is no longer sufficient to represent the differences and dependences at the boundary, the development of a common meaning becomes essential (Carlile 2004, p. 558, Bergman et al. 2007). The DRT offers several mechanisms within its shared interface (e. g., detailed description fields) as a framework for the discovery of discrepancies. For example, the first descriptions of attributes in the pre-defined TDM are later aligned with the users' definitions during meetings (see Description column in Table 1). The DWH professionals also include a flagging-based status system within the DRT's interface to facilitate the negotiation process between OSPs and DWH professionals, and to narrow down the number of unconnected attributes in the pre-defined TDM by mobilizing resources and power. For example, the status flag "OPEN" is a typical example of how the DRT actively supports the mobilization process: as a result of an undecided or unknown connection between the SDM and the TDM, the status of the disputed attribute is set to "OPEN", indicating that either party needs to mobilize its own resources and power (i. e. organize internal meetings) to resolve the connection issue (see Status column in Table 1). Our third guideline therefore states: *the status of the shared understanding, its negotiation process, needs to be actively controlled by project participants within a DBO*.

4.4 Legitimization of Design Knowledge

Legitimization is the capability to legitimize design knowledge across social worlds (Bergman 2009, p. 399). According to Bergman et al. (2007), DBOs need to be granted a legitimate status through the validation of their content. This aligns the stakeholders' intents by certifying, verifying and validating the truthfulness and correctness of the design. The DRT is used in every iteration of a meeting, where a subset of attributes is being discussed and the status of affected attributes changes accordingly in relation to the agreement between all participating parties. After each and every attribute has been tested for feasibility of source data (i. e., the status flag is set to “Ok” or "Functional2", depending who will be responsible for delivering the data, DWH professionals or users respectively; see Comment column in Table 1), the data requirements analysis enters a new phase, *design and implementation*. The discovery of the incorrect connections revives discussions between participants (i. e., IFR_CATEGORY changes several times its status, from “OPEN” to “Functional2”; see Status column in Table 1). As long as the mutual understanding of the common ground is not established, the participants of the discussion are stating their views, but do not get closer to a solution (although they might have an illusion of it). To prevent *illusion of evidence*, during this phase, the TDM attributes as well as the DRT interface can be newly redefined in order to gather missing information or enhance user’s understanding, leading to repeated legitimization. The last phase of the requirements analysis process, the *sample data analysis*, includes data tests to reproof defined connections. We observe that the set of connections goes through several circles of legalization until their status is legitimised by all participants. Therefore we argue that DBOs need to possess sufficient clarity of representation for each group to be able to identify mistakes and redefine the connections as well as mechanisms such as flagging in order to track their current status. As a consequence, we suggest as our fourth guideline that *DBOs should support the testing of design objects through repeated instantiations* (Simon 1996, Hevner and March 2003, Gregor 2006).

5 DISCUSSION

While the research of Bergman et al. (2007, 2009) tackles the issue how DBO may help project teams to effectively handle design process involving multiple heterogeneous social worlds, they have largely overlooked the complexity of the “birth” of such a DBO. Nevertheless, understanding how the four DBO features can actually be implemented and integrated in specific DBOs is an important first step in addressing these challenging problems. Thus far, no methodology or guidelines were available that can help both researchers and practitioners. Our approach to create a set of guidelines for the better development of a DBO in financial DWH projects is less focused on detecting the features within the boundary objects; rather, it emphasizes implementing them into the boundary objects.

Our analysis of the DRT with regard to the essential characteristics of a DBO according to Bergman et al. (2007) suggests that the DRT’s capability to promote the shared representation effectively addresses the challenges of creating mutual understanding in project work. Most of the problems encountered during the development of the tool were perceived as misunderstanding-driven. Thus, iterative refinement of the DRT interface context and attributes’ descriptions within the TDM was a contributing factor. Only then the DRT became successful in terms of its ability to support participants in interaction about relevant design issues. These findings may suggest that the creation of a language community between participants has a positive influence on DBOs’ shared representation features.

Moreover, assuring shared representation, the DRT fostered better cooperation, collaboration and dialog among participants so that the participants’ knowledge has been transformed in a more emergent manner. Clearly, the amended DRT’s interface also improved the tool’s capability to transform OSPs’ and EDMFs’ domain knowledge. When the DWH professionals introduced the flagging system within the DRT’s shared interface, they promoted and accelerated a dialogue in which they called for a certain level of consensus. In turn, this dialogue required from participants to compare and contrast their assumptions and interpretations. However, mechanism such as this could enforce a DBO’s capability to mobilize for action only between the participants of the same language community. Otherwise, reached level of consensus could be an illusion, which in turn would trigger errors later in the project. In the case of the DRT, by introducing repeated instantiations and tests, the DWH professionals prevented false solutions to be legitimized (e. g., definition of a connection between SDM and TDM) and enforced the DRT’s legitimization capability. We summarize our discussion by proposing a set of guidelines for the better development of a DBO in financial DWH projects in Table 2.

In general terms, we view the implementation of a DBO’s features within the boundary objects as a repeating process or circle. When a change in common representation occurs, the effects are spread onto the other DBO features. For example, if the change in attributes' descriptions within the TDM resulted in a better understanding by participants then the transformation of the domain-specific design knowledge, mobilization for action and legitimization of the design knowledge across social worlds was also positively affected (see Table 2 first column on the left). According to the fourth guideline, repeated instantiations can cause a redefinition of the shared representation, closing the circle of development (see Table 2, the first column on the right). In other words, in case of the DRT, a DBO cannot be observed as a static entity, but is rather under a constant reconstruction and reimplementation. The DRT evolved during the process of DWH development to be a better and more integrated DBO.

	Feature	*Finding from DRT*	*Guideline*	
	Capability to promote shared representation	Iterative refinement of TDM attributes' descriptions and DRT interface context in close cooperation with OSPs	• Assure mutual understanding of interface and DBO's content, focusing on pragmatics in practice (a language community between all participants) • Align the DBO to support this process (e. g., flagging-system).	←
→	Capability to transform design knowledge	Parts of the DRT's interface are amended during development of the tool to catch needed information from users	• Building on the "shared pragmatic representation", assure sufficient *common lexicon* according to Carlile (2004). (Assure a common syntax level by creation of mutual semantic and pragmatic levels in practice.) • Align the DBO to support this process (e. g., add new tool-features).	
→	Capability to mobilize for action	The mechanisms within the DRT's shared interface for the discovering of discrepancies and mobilization need to be understandable to all participants	• Provide information about the status of the common understanding (controlling) • Align the interface of the DBO to support this process (e. g., add alarms, flagging-system).	
→	Capability to legitimize design knowledge	Flagging-based status system within DRT's interface was not sufficient (problem of illusion of evidence)	• Test the design object through repeated instantiations to prevent illusion of evidence (legitimize correctness of design knowledge) • Align the DBO to support this process (e. g., reproof with *sample data analysis* in design and implementation. phase)	

Table 2. Summary of Findings and Guidelines for DBO Development

6 CONCLUSION

In this paper, we have taken Bergman's et al. (2007) DBO design theory as a theoretical framework for the evaluation of a DBO's features. Whether or not Bergman's et al. (2007) original set of features is in fact appropriate must be considered an open question. As part of our future work we plan to investigate whether sets of features different from the suggested ones are (a) possible and (b) more suitable to the context of financial DWHs and complex IS projects in general. For example, it would be interesting to observe DBOs' features in the context of "content vs. process common ground" (Convertino et al. 2009). This suggests that DBOs would need to fulfil Bergman's et al. (2007) four criteria both on the process level (shared understanding of the DBO itself) and on the content level (shared understanding of the content processed by DBOs) to become "good" design boundary objects.

The DRT tool was developed out of a practical need for creating common ground in complex development situations. Our short analysis can be seen as corroborating and supporting Bergman's et al. (2007) proposals. Although it originates from a single and rather limited analysis of an exemplary industrial case, it compensates this to some degree by the longitudinal usage of the DRT by the consultancy and its iterative refinement in more than fifteen projects in a period of more than four years. Our guidelines were deduced from the close examination of the DRT's development history.

However, no rigorous methodological examination or testing has been done yet. We plan to address this in further studies, for example, using comparative in-depth case studies and field experiments. With regard to our future research, we also plan to investigate if the proposed set of guidelines is valuable in its current (or an extended) form. Only empirical research can establish whether the set of guidelines, as captured in our analysis, does in fact help develop a better DBO. By calling attention to the construction of DBOs, we tried to push the discussion more concretely in the direction of identifying steps that could help both researchers and practitioners in the development of better DBOs.

References

Al-Rawas, A. and Easterbrook, S. (1996). Communication problems in requirements engineering: a field study. In First Westminster Conference on Professional Awareness in Software Engineering London, UK.

Behrmann, W. and Räkers, M. (2008). Specifics of Financial Data Warehousing and Implications for Management of Complex ISD Projects. In 16th European Conference on Information Systems (ECIS 2008) Galway, Ireland.

Bergman, M. (2009). Requirements' Role in Mobilizing and Enabling Design Conversation. In Design Requirements Engineering: A Ten-Year Perspective, Vol. 14 (Eds, Lyytinen, K., Loucopoulos, P., Mylopoulos, J. and Robinson, B.) Springer, pp. 394-407.

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

Bergman, M., Lyytinen, K. and Mark, G. (2007). Boundary Objects in Design: An Ecological View of Design Artifacts. Journal of the Association for Information Systems, 8 (11), pp. 546-568.

Bostrom, R. P. and Kaiser, K. M. (1981). Personality differences within systems project teams: Implications for designing solving centers. In 18th Computer Personnel Research Conference Washington, D.C., United States, pp. 248-285.

Brown, J. S. and Duguid, P. (2001). Knowledge and Organization: A Social-Practice Perspective. Organization Science, 12 (2), pp. 198-213.

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

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

Clark, H. H. (1996). Using Language. Cambridge University Press, Melbourne.

Convertino, G., Mentis, H. M., Rosson, M. B., Slavkovic, A. and Carroll, J. M. (2009). Supporting content and process common ground in computer-supported teamwork. In Proceedings of the 27th International Conference on Human Factors in Computing Systems ACM, Boston, MA, USA, pp. 2339-2348.

Corcho, O., Losada, S., Martínez Montes, M., Bas, J. L. and Bellido, S. (2005). WP10: Case study eBanking. Financial Ontology. DIP deliverable D10.3. http://dip.semanticweb.org/documents/D10.3.pdf.

Gregor, S. (2006). The Nature of Theory in Information Systems. MIS Quarterly, 30 (3), pp. 611-642.

Hansen, S. and Lyytinen, K. (2009). Distributed Cognition in the Management of Design Requirements. In Perspectives Workshop: Science of Design: High-Impact Requirements for Software-Intensive Systems (Eds, Jarke, M., Lyytinen, K. and Mylopoulos, J.) Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany.

Herrmann, C. and Melchert, F. (2004). Sponsorship Models for Data Warehousing: Two Case Studies. In 10th Americas Conference on Information Systems (AMCIS 2004) New York, NY, USA, pp. 1894-1903.

Hevner, A. R. and March, S. T. (2003). The Information Systems Research Cycle. Computer, 36 (11), pp. 111-113.

Hwang, H.-G., Ku, C.-Y., Yen, D. C. and Cheng, C.-C. (2004). Critical factors influencing the adoption of data warehouse technology: a study of the banking industry in Taiwan. Decision Support Systems, 37 (1), pp. 1.

Hwang, M. I. and Xu, H. (2007). The Effect of Implementation Factors on Data Warehousing Success: An Exploratory Study. Journal of Information, Information Technology, and Organizations, 2.

Jarke, M., Lyytinen, K., Mylopoulos, J., Kappel, G., Leite, J., Mark, G., Ramesh, B., Schmitz, D. and Sutcliffe, A. G. (2009). High-Impact Requirements for Software-Intensive Systems: Seminar Outlines and Working Group Summaries. In Perspectives Workshop: Science of Design: High-Impact Requirements for Software-Intensive Systems (Eds, Jarke, M., Lyytinen, K. and Mylopoulos, J.) Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany.

Kamlah, W. and Lorenzen, P. (1984). Logical Propaedeutic. Pre-School of Reasonable Discourse. University Press of America, Lanham, MD, USA.

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

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

Miles, M. B. and Huberman, A. M. (1994). Qualitative Data Analysis: A Sourcebook of New Methods. Sage, Beverly Hills, CA, USA.

Pawlowski, S. and Robey, D. (2004). Bridging User Organizations: Knowledge Brokering and the Work of Information Technology Professionals. MIS Quarterly, 28 (4), pp. 645-672.

Peppard, J. (2001). Bridging the Gap between the IS Organization and the Rest of the Business: Plotting a Route. Information Systems Journal, 11 (3), pp. 249-270.

Rizzi, S., Abell, A., Lechtenbörger, J. and Trujillo, J. (2006). Research in data warehouse modeling and design: dead or alive?, Proceedings of the 9th ACM international workshop on Data warehousing and OLAPACM, Arlington, Virginia, USA.

Rosenkranz, C. (2009). The Quality of Organizations: A Communication-Based Measurement Approach. Economics and Business Administration, Goethe-University, Frankfurt am Main, Germany.

Simon, H. (1996). The Sciences of the Artificial. MIT Press, Cambridge.

Star, S. L. and 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), pp. 387–420.

Watson, H. J., Fuller, C. and Ariyachandra, T. (2004). Data warehouse governance: best practices at Blue Cross and Blue Shield of North Carolina. Decision Support Systems, 38 (3), pp. 435-450.

Watson, H. J. and Wixom, B. H. (2007). The Current State of Business Intelligence. Computer, 40 (9), pp. 96-99.

Weir, R., Peng, T. and Kerridge, J. (2003). Best practice for implementing a data warehouse: A review for strategic alignment. In Proceedings of the 5th Intl. Workshop (DMDW'2003) (Eds, Lenz, H.-J., Vassiliadis, P., Jeusfeld, M. and Staudt, M.) Berlin, Germany.

Wenger, E. (1998). Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press.

Wenger, E., McDermott, R. and Snyder, W. (2002). Cultivating Communities of Practice: A Guide to Managing Knowledge. Harvard Business School Press, Boston, MA, USA.