Ontological Evaluation of Conceptual Models

A Linguistic Interpretivist Approach

Jörg Becker, Björn Niehaves, Daniel Pfeiffer

European Research Center for Information Systems, University of Münster, Germany
{becker, niehaves, pfeiffer}@ercis.uni-muenster.de

Abstract. The objective of this paper is to present a philosophically sound approach to conceptual model evaluation. Accordingly, the ontological evaluation of conceptual models is enriched with a linguistic interpretivist perspective. The need for such an approach to evaluation is justified by the substantial economic importance of conceptual models. The quality of a conceptual model has a significant impact on other IT artefacts and, thus, on the costs of IT projects. However, little research has so far focused on their evaluation. In the course of this paper, we develop a framework which describes the current state of research and recognizes neglected research fields. With the aid of this framework we identify a notable shortcoming in conceptual model evaluation research, especially with respect to philosophically sound evaluation procedures. Based on these findings we address the following research questions: What are the shortcomings in current evaluation research, what are the merits of 'ontological evaluation' in this context, and how can the linguistic interpretivist approach help to form a comprehensive and philosophically sound conceptual model evaluation approach?

Key words: ontology, epistemology, model quality, conceptual model evaluation, linguistic interpretivism.

1 Introduction

Since the mid-1970s, conceptual models have been employed to facilitate and systematise the process of information systems engineering. A remarkable number of modelling languages and methods have been proposed, aiming at a more efficient and effective software development (Mylopoulos 1998; Wand et al. 1995). At the beginning of the 1990s, encouraged by new findings in management science, the positive experiences with conceptual models were transferred from Information Systems (IS) engineering to organisational design. This process established conceptual models as a widely-used means of determining customer requirements and documenting the project progress of a software system. In addition, it enabled the description of the business processes and corporate structures in an organisation (Shanks et al. 2003). The significance of conceptual models is also reflected in the proposal to define them as the core of the IS discipline (Weber 2003b).

The quality of conceptual models has an immense impact on other IT artefacts (Hevner et al. 2004; March and Smith 1995). Software systems are often based on explicit requirement specifications in the form of conceptual models. The adequacy of these specifications regarding the represented application domain, determines the acceptability and usability of software systems (Lauesen and Vinter 2001). An incorrect description of the application domain potentially leads to problems in the implemented software system or to delays in the project progress. Likewise, the success of a reorganisation project is influenced by the adequacy of the underlying organisational models. A problem analysis based on flawed models can lead to wrong and ultimately very cost-intensive decisions. As a result, the quality of conceptual models is of considerable economic importance.

The scientific and practical significance of conceptual models necessitates an evaluation of these artefacts. During the last few years, numerous research efforts have been undertaken to develop criteria catalogues for evaluating the quality of conceptual models. Nonetheless, little empirical work has focused on their evaluation (Hevner et al. 2004; Wand and Weber 2002). This raises the research questions addressed in this paper:

- What are the shortcomings of current research on the evaluation of IT artefacts? (cf. Section 2)
- How can the concept of ‘ontological evaluation’ (cf., for instance Green and Rosemann 2000; Shanks et al. 2003; Wand et al. 1999; Wand and Weber 1993) contribute to closing the identified research gap? (cf., Section 3)

- What additional value could a linguistic interpretivist approach provide in terms of forming a comprehensive and philosophically sound conceptual model evaluation approach? (cf., Section 4)

The paper is structured as follows: We develop a framework for structuring existing literature on the evaluation of IT artefacts (Section 2). With the help of this framework, we classify existing research results, identify their shortcomings and, thus, identify the need for further research. Based on these insights, we briefly describe the concept of 'ontological evaluation' and discuss its potential contribution to evaluating conceptual models (Section 3). At this juncture, we also consider possible shortcomings of this approach in the specific context of model evaluation. We then outline the essence of Kamlah and Lorenzen's (1984) linguistic interpretivist approach to conceptual model evaluation. We seek to point out how both approaches can usefully be combined (Section 4). Thus, the main objective of this paper is to provide a philosophically sound conceptual model evaluation approach. The paper concludes with a summary of the main findings and a discussion of future research opportunities (Section 5).

The selected research method is that of conceptual/philosophical research. Accordingly, we provide philosophical-logical arguments, rather than empirical ones. However, in developing our argumentation, where applicable, we also refer to previous empirical research results. Furthermore, we present additional evidence in the form of examples from IS research.

2 IT artefact evaluation framework and related work

The evaluation of IT artefacts has been an active research field for the last 20 years. During this time, manifold approaches to assessing the quality of artefacts have been proposed. In order to describe the current state of research and identify possible shortcomings, it is helpful to systematise these approaches with the aid of a conceptual framework. The framework we propose in this paper spans two dimensions, the IT artefact (A) and the evaluation element (E).

The first dimension addresses the question of what can be considered an IT artefact (A). Catalogues of IT artefacts have been suggested by many researchers in the IS field (cf., for instance Benbasat and Zmud 2003; Carroll and Kellogg 1989; Hevner et al. 2004; March and Smith 1995; Orlikowski and Iacono 2001; Weber 1987). In this paper, we follow March and Smith (1995) and Hevner et al. (2004) who identify four artefacts in IS research: constructs, methods, models and instantiations (cf., Table 1).

A1	*Constructs*	Constructs provide the language concepts in which the problem is described and the solution communicated.
A2	*Methods*	Methods explicate the problem-solving processes and offer guidance on how to search the solution space.
A3	*Models*	Models utilise the constructs to represent an application domain and express the problem and solution space.
A4	*Instantiations*	Instantiations constitute the realisation of constructs, models and methods within a working system.

Table 1: Artefact dimension of the framework to evaluate IT artefacts

The second dimension deals with the different elements that are required to evaluate an IT artefact (E). This dimension addresses the question of what aspects of IT artefacts quality assessment should be considered (structurally reflecting the review of related work): the structure of the artefact, evaluation criteria and the evaluation approach/procedure (cf., Table 2).

E1	*Structure of the artefact*	The structure of the IT artefact contains all the information that is required to enable its evaluation. This structure can include: information on the intended purpose of an artefact, the requirements on which an artefact was originally based or the formalism applied to describe it. In other words, the structure of the IT artefact represents the information space available for its evaluation.
E2	*Evaluation criteria*	The evaluation criteria define the relevant aspects for assessment. These criteria specify the dimensions of the information space which are important for determining the utility of the artefact. The criteria are situation-specific and can differ depending on the purpose of the evaluation.
E3	*Evaluation approach*	The evaluation approach/procedure defines all the activities needed to achieve the evaluation result. The approach specifies the roles associated with the assessment and the activities they must perform. Thus, the evaluation approach provides guidance on how to use the evaluation criteria to assess the quality of the artefact. The result is a decision as to whether the artefact meets the evaluation criteria based on the information space available.

Table 2: Evaluation dimension of the framework to evaluate IT artefacts

Taking into account these two dimensions, the IT artefact evaluation framework can be applied to systematise related research work (for an overview of results and examples, see Table 3):

Structure of the constructs (A1/E1): The structure of the constructs is defined by a language-based metamodel (cf., for instance Davies et al. 2002; Guizzardi et al. 2002; Hong et al. 1993; Odell 1995; Rosemann and Green 2002; Saeki 1995). A language-based metamodel links the constructs of a modelling language and, thus, defines its syntax and parts of its semantics. In order to evaluate a language, it is necessary to know how the constructs are related. Without a metamodel, the evaluation of a construct would depend significantly on the personal interpretations of individuals involved. There have been several proposals in the literature as how to define a metamodel (cf., for instance Brinkkemper et al. 1999; Greiffenberg 2004; Lemesle 1998; Object Management Group 2006).

Evaluation criteria for constructs (A1/E2): Wand and Weber (1990) have proposed the ontology of Bunge (1977; 1979) as a theoretical reference point for IS constructs. The resulting conceptualisation is known as the Bunge-Wand-Weber (BWW) ontology. Other researchers recommend alternative ontologies, including General Ontological Language (Degen et al. 2001) or Chisholm Ontology (Milton et al. 1998), as a theoretical foundation of modelling languages. Based on the *BWW ontology*, Wand and Weber (1996; 2002) have defined a set of ontology-based evaluation criteria: construct deficit, construct overload, construct redundancy and construct excess. A *construct deficit* exists if a modelling language does not provide all constructs available in the ontology. A *construct overload* can be observed when a modelling language concept can be mapped to more than one concept within the ontology. *Construct redundancy* refers to a case in which two constructs of the modelling language represent the same concept of the ontology. If the language elements have no ontological counterpart, there is a construct excess. Other evaluation criteria for constructs have been derived from cognition science and the philosophy of language. Evermann (2005) describes a set of cognitive concepts that can be used to evaluate conceptual modelling languages. Larkin and Simon (1987) discuss the differences between a graphical and a textual representation of constructs. Siau (2003) derives evaluation criteria for constructs and modelling languages from findings in human information processing systems. Kim et al. (2000) provide evaluation criteria for modelling languages that support multiple forms of diagrams. Nysetvold and Krogstie (2005) describe a generic quality framework for the evaluation of modelling languages.

Evaluation procedure for constructs (A1/E3): Based on the BWW ontology and its *'ontological analysis',* Rosemann et al. (2004) propose a detailed evaluation procedure for performing an ontological assessment of constructs. The ontological analysis specifies the evaluation as an iterative, incremental process in which multiple researchers must be involved. The evaluation process uses a metamodel of the BWW ontology (Rosemann and Green 2002) as input and compares it with a metamodel of the modelling language. In three different steps, the interpretations of the different team members are studied according to the evaluation criteria and, then, consolidated. This procedure has been applied in many different research projects (for an overview cf., for instance Green et al. 2007). However, the ontological analysis of constructs and modelling languages as well as the entire program of developing an ontological foundation for conceptual modelling, is not undisputed (Wyssusek 2006). While its opponents challenge the validity of the theoretical foundation, its supporters stress the useful insights that have been gained from BWW ontology-based research (cf., for instance Lyytinen 2006). Another stream of research attempts to avoid the theoretical problems of a single "correct" ontology by instead applying multiple modelling languages. These approaches do not evaluate a modelling language against a single ontology, but they compare different modelling languages with each other. The evaluation of the modelling languages and constructs is based on their *metamodels* (Siau and Rossi 1998). Depending on the approach, the languages are either compared directly (cf., for instance Davis et al. 2003; Ledeczi and Paka 2003; Rosemann and zur Mühlen 1998) or a consolidated set of constructs is first created and used subsequently as reference point for the evaluation (cf., for instance Hong et al. 1993; Song and Osterweil 1992; Strahringer 1996). Besides an ontological analysis and a metamodel comparison, evaluations of constructs and languages are often based on *behavioural theories*, such as Cognitive Fit Theory (cf., for instance Agarwal et al. 1996), Technology Acceptance Model (cf., for instance Batra et al. 1990; Burton-Jones and Weber 1999; Recker and Rosemann 2007) or findings from the psychological literature (cf., for instance Gemino and Wand 2005; Siau et al. 1995).

Structure of a method (A2/E1): The constitutive part of a method is given by a process model which describes how to reach the specific objective of the method. In method engineering, many proposals have been made as to what elements constitute parts of a method (cf., for instance Gupta and Prakash 2001; Harmsen 1997; Heym and Österle 1993; Karlsson and Ågerfalk 2004; Ralyté and Rolland 2001). Greiffenberg (2003) has developed a method structure which aims at improving the testability of a method. Within this structure, a process model has to state explicitly the *products and results* of its application as well as the *constructs used in this context*. To be able to appraise the applicability of the method, it must describe its *conditions and intended scope of application.* Greiffenberg (2003) tries to span the required information space with such a structure in order to enable the evaluation of methods.

Evaluation criteria for methods (A2/E2): In (situational) method engineering, a broad stream of research is concerned with contingency factors of methods (cf., for instance Avison and Wood-Harper 1991; Benyon and Skidmore 1987; Davis 1982; Fitzgerald et al. 2003; Topi and Ramesh 2004). These criteria can be used to select or customise an existing method or to assess its performance. Explicit measures for evaluating modelling methods have also been proposed by many researchers (cf., for instance Brinkkemper et al. 1999; Gemino and Wand 2004; Greiffenberg 2003). Greiffenberg (2003) defines, for example, the criterion of *appropriateness* that verifies whether the method is efficient, well structured and easy to apply. A method is *complete* if it describes its in- and output, as well as its process and relations. The requirement of *consistency* of a method is fulfilled if all methodological elements are mutually compatible. Additional criteria for the evaluation of methods can be derived from feature-based comparisons of modelling methods (cf., for instance Boertien et al. 2001; Halpin 1999; Monarchi and Puhr 1992; Túbio et al. 1999). Based on these findings, the different features of a method can be translated into evaluation criteria. In order to operationalise these measures, metrics have been proposed (cf., for instance Bajaj 2000; Rossi and Brinkkemper 1996; Siau and Cao 2001).

Evaluation procedure for methods (A2/E3): Based on the results obtained by Wynekoop and Russo (1997) the evaluation of methods is often conducted in terms of *field inquiries, surveys, case studies and action research*. There are also many publications on method evaluation by means of laboratory experiments (cf., for instance Batra and Davis 1992; Kamsties et al. 2001; Kim and March 1995; Siau et al. 1996). In contrast, in practice, descriptions and interpretative research are hardly used. Moody (2003) provides a general framework for an empirical validation of conceptual models. A comprehensive overview of evaluation approaches for modelling methods can be found in Siau and Rossi (1998).

Structure of an instantiation (A4/E1): The software engineering community widely agrees on the elements that an instantiation must comprise. An executable implementation in a programming language must always be accompanied by a requirements specification and a design model (Sommerville 2001). Good documentation and reference to the configuration management, quality management and project-management files influence the verifiability of the software system. Elaborated software process models facilitate a corresponding structure of instantiations.

Evaluation criteria for instantiations (A4/E2): The FURPS model is a prominent representative of the multitude of criteria catalogs dealing with the evaluation of software quality (Grady and Caswell 1987). In this model, the criteria of *functionality, usability, reliability, performance and supportability* build the basis for software quality assessment. Many alternative criteria catalogs can be found in the software engineering literature (cf., for instance Cavano and McCall 1978; Dromey 1996; Ortega et al. 2003; Preece 1995; Rawashdeh and Matalkah 2006; Wand and

Wang 1996). Some of the evaluation criteria have also been operationalized in terms of software metrics (cf., for instance Basili et al. 1994; Farbey 1990; Fenton and Neil 2000; Hudepohl et al. 1996; Seffah et al. 2006).

Evaluation procedure for instantiations (A4/E3): Code inspections, testing, code analysis and verification are established evaluation approaches for facilitating high software quality (cf., for instance Bassin et al. 1998; Card 1990; Fairley 1985; Grünbacher et al. 2004; Kan et al. 1994; Wallace and Fujii 1989). Software companies engage quality engineers and software testers to conduct these procedures.

Model structure (A3/E1): This paper focuses mainly on the quality of models. Pfeiffer and Niehaves (2005) define a structure for conceptual models which can be considered as necessary for the purposes of their evaluation. Based on a structuralist approach, they deliver a conceptual model together with a *description of its intended applications*, an *explicit reference to a modelling language* (constructs) and a link to a *description of the application domain language*. It is claimed that this information is necessary, to evaluate the conceptual model. This notion for a conceptual model has been formalized in Pfeiffer (2007).

Evaluation criteria for models (A3/E2): Multiple attempts have been undertaken to develop criteria catalogs for determining the quality of models (cf., for instance Kesh 1995; Levitin and Redman 1995; Lindland et al. 1994; Moody and Shanks 1994; Nelson and Monarchi 2007; Siau and Tan 2005). The Guidelines of Modelling (GoM) (Schütte and Rotthowe 1998) constitute one of these approaches. They contain the measures: *construction adequacy, language adequacy, economic efficiency, clarity, comparability and systematic design*. Mayer (1989) defines criteria for a good conceptual model from a psychological perspective. Bajaj (2004) discusses the impact of the number of constructs on the readability of models. Some researchers propose domain ontologies as a relevant means of assessing the quality of a model (cf., for instance Höfferer 2007; Sugumaran and Storey 2006).

Evaluation procedure for conceptual models (A3/E3): Little research has been conducted in the area of conceptual model evaluation approaches and procedures (Shanks et al. 2003). The literature provides no detailed guidance on how to apply the abovementioned evaluation criteria to specific models. How can an existing model be mapped against a domain ontology? How can conflicts between the model and a domain ontology be resolved? A comprehensive approach that answers these questions does not exist. In our opinion, there are two main reasons why the scientific debate on structure and evaluation approaches has so far not considered models of this kind. Firstly, models focus on a certain application domain and are usually less general than constructs and methods. Thus, at first sight, it does not seem worthwhile to investigate their evaluation. Secondly, in contrast to instantiations, they are less tangible. Therefore, their importance can easily be overlooked in a practical project situation. Consequently, a major shortcoming of IT artefact evaluation research lies in the field of conceptual model evaluation procedures, which, therefore, form the focus of this paper (cf., Table 3).

	Structure of the artefact (E1)	*Evaluation criteria (E2)*	*Evaluation approaches (E3)*
Constructs (A1)	metamodel	construct deficit construct overload construct redundancy construct excess	ontological analysis metamodel-based comparison investigations based on behavioural theories
Methods (A2)	products and results of their application constructs used in this context conditions and intended scope of application	appropriateness completeness consistency	field inquiries survey case studies action research laboratory experiments
Models (A3)	intended applications reference to a modelling language link to a description of the application domain	construction adequacy language adequacy economic efficiency clarity comparability systematic design	→ focus of this paper
Instantiations (A4)	requirements specification design model documentation configuration management, quality management and project management files	functionality usability reliability performance supportability	code inspections testing code analysis verification

Table 3: Framework to evaluate IT artefacts

Accordingly, in our investigation, we are concerned with approaches and procedures for evaluating conceptual models. For constructs, the ontological analysis has been presented as a popular evaluation procedure (see references above). In the next section, we, thus, consider whether an ontology-based approach is also feasible for the evaluation of models.

3 Ontology-based evaluation of conceptual models

3.1 Ontology and language

Ontological thinking has already exerted a significant impact on the IS literature and shaped the discussion on conceptual modelling and evaluation (Ashenhurst 1996; Fox et al. 1998; Green and Rosemann 2000; Green and Rosemann 2002; Rosemann and Green 2000; 2002; Shanks et al. 2003; Wand et al. 1999; Wand and Weber 1993; Weber 2003a). Ontologies have frequently and successfully been applied to evaluate conceptual modelling grammars (constructs) and modelling methods. A top-level ontology developed by Bunge (1977; 1979) which inspired Wand and Weber (1990) to build a framework for evaluating conceptual modelling techniques, was particularly influential. Here, Wand and Weber (1990) use the term 'conceptual modelling grammar', reflecting the fact that only the language aspect (constructs; A1) of a conceptual modelling method is assessed, rather than the method as a whole (Wand and Weber 2002). An application of the Bunge-Wand-Weber-Ontology for evaluating modelling methods (A2) can be found, for instance, in Wand and Weber (1990) and Rosemann and Green (2000). It is used, in particular, for evaluating the languages EPC (Green and Rosemann 2000; Rosemann and Green 2000) and ERM (Wand et al. 1999). Rosemann and Green (2000) provide an overview of papers using the Bunge-Wand-Weber approach for evaluating conceptual modelling grammars.

There are many distinct categories of ontologies, depending on their degree of generalisation or specialisation. In this respect, Guarino (1998) differentiates between four types of ontologies:

1. *Top-level ontologies* describe very general concepts, such as time and space. They are independent of particular domains or problem tasks. The ontology created by Bunge, for instance, is just such a top-level ontology. Typical statements are those about 'things', 'properties' or 'attributes' (Bunge 1977; 1979).

2. *Domain ontologies* describe a basic vocabulary regarding a particular domain/real-world object system. "[A domain] ontology is a shared understanding of some domain of interest." (Uschold and Grüninger 1996). Domain ontology statements are specialisations of those made in top-level ontologies; for instance, 'information system', 'user', 'power user' or 'runtime'.

3. *Task ontologies* describe a basic vocabulary regarding a particular activity or task. This type of ontology is also a specialisation of a top-level ontology.
4. *Application ontologies*, furthermore, focus on a particular domain or task. They are, thus, specialisations of both the former and the latter.

Domain languages are used within conceptual models. When designing a conceptual model, essentially two languages are applied: a modelling language and a domain language (Pfeiffer and Niehaves 2005). Modelling languages provide (formal) procedures (A2) and constructs (A1) to build a conceptual model. They include, for example, Event-driven Process Chains (EPC) (cf., for instance Green and Rosemann 2002; Rosemann and Green 2000) or Entity-Relationship Model (ERM) (cf., for instance Wand et al. 1999). Modelling languages, on the one hand, are addressed by the approaches to 'ontological evaluation' (Wand and Weber 1993). Domain languages, on the other hand, provide terms and concepts relating to the particular problem domain addressed by the conceptual model. A domain language provides an understanding of domain terms that are part of a conceptual model, such as 'user', 'information system' or 'power user'. A domain language might provide more terms and concepts than a specific conceptual model would refer to. For instance, certain (real world) elements described by a domain language could be regarded as irrelevant with respect to the intention of a given conceptual model.

Domain ontologies can be used as an instrument to (partly) represent domain languages. An ontology is "[…] a shared and common understanding of a domain that can be communicated across people and computers." (Studer et al. 1998) They are "[…] a shared understanding of some domain of interest." (Uschold and Grüninger 1996) Since they serve as a communication instrument, (domain) ontologies can be used to explicate a given domain language. In fact, for the majority of cases domain ontologies would fulfill this function. Both domain language and domain ontology would possibly comprise terms such as 'information system', 'user' or 'power user'. Nevertheless, also top-level, task and application ontologies could have a supportive role. In general, a domain ontology does not necessarily comprise all terms and concepts provided by a domain language (cf., for instance Fadel et al. 1994; Fox et al. 1998; Uschold 1998; Zuniga 1999).

3.2 Ontological evaluation of conceptual model statements

Conceptual model statements can be evaluated by comparing them to ontology statements. A domain ontology comprises a large set of terms and concepts and may be applied to represent a domain language (see Section 3.1). However:

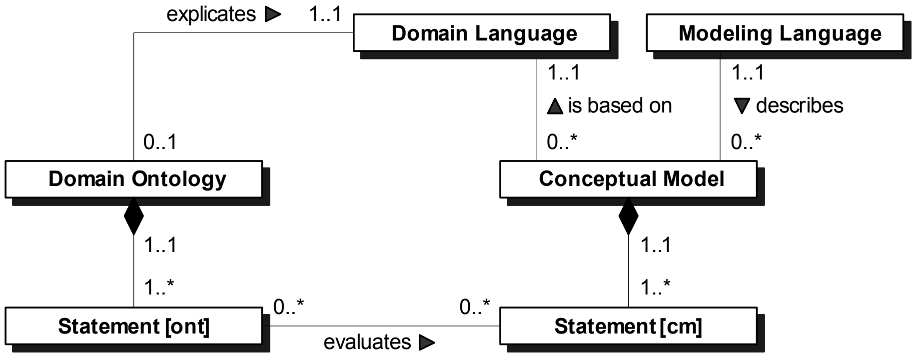

Figure 1: Ontological evaluation of conceptual model statements

1. not all of the domain language terms and concepts can be formalized into a domain ontology and
2. conceptual models may refer only to a small part of existing terms and concepts.

Thus, the terms and concepts that are part of a domain ontology and those referred to by a particular conceptual model are not necessarily congruent. Here, we can differentiate between two general categories that are essential to the ontological evaluation of conceptual models:

Congruence of ontological and conceptual model statements.

A very rare case in practice, is that the conceptual model statement which is to be evaluated, refers only to terms and concepts that are part of a domain ontology. Both statements, i.e., the conceptual model statement and the ontology statement, refer to the same entities (cf., Figure 1). An example of possible congruence of a conceptual model and ontological statement could be that "a 'power user' is a specialisation of a 'user'". Hence, if the conceptual model statement has an equivalent statement in the ontology, the former can be evaluated with the help of the latter.

Incongruence of ontological and conceptual model statements.

The usual and more complicated case is an incongruence of conceptual model and ontological statements. This means that the ontology and/or the conceptual model include terms and concepts and express statements not made by the other. Some IS authors analyse the case in which a domain ontology is more comprehensive than a conceptual model with respect to the statements made (Green and Rosemann 2000; Wand and Weber 1993). However, the more critical case in practice is that a domain ontology does not comprise all terms and concepts referred to by a conceptual model statement (cf., for instance Fadel et al. 1994; Fox et al. 1998; Zuniga 1999). Thus, a clear evaluation statement cannot be made. For instance, particular constructs and terms of the statement "All 'power users' using the 'information system XYZ' have 'computer experience of more than 10 years'" are part of a conceptual model but not part of a domain ontology. In such a case, ontologies can only be used to evaluate certain terms or constructs used in the conceptual model, for instance, whether a 'power user' uses an 'information system'. However, evaluating the entire statements with respect to the 'unknown' terms and concepts is not possible. Therefore, in the following section, we elaborate on possible philosophical foundation of approaches and procedures for evaluating conceptual models. We consider the case of when the conceptual model statements cannot be completely evaluated with the aid of a domain ontology.

4 Linguistic interpretivist extension of conceptual model evaluation

4.1 Philosophical foundation of linguistic interpretivism

Linguistic interpretivism closes a major philosophical gap in conceptual model evaluation. Kamlah and Lorenzen (1984) established the 'school of methodical constructivism', which is also regarded as linguistic interpretivism. A major objective of this philosophical school is to mediate between the positivist paradigm—which is perceived as understating the social and individual embeddedness of knowledge—and the hermeneutic philosophy—which appears to become diffuse or arbitrary in terms of practical considerations (Kamlah and Lorenzen 1984). Thus, linguistic interpretivism seeks to provide a philosophical basis for an interpretivist epistemological understanding by focusing on the social artefacts of speech and language. However, the debate on epistemological questions must be considered as an open issue. No theory based on a Philosophy of Science can be regarded as binding for IS researchers (Becker and Niehaves 2007). Nonetheless, linguistic interpretivism will become particularly relevant because it closes a gap in the philosophically sound evaluation of conceptual models.

Linguistic interpretivism assumes language as the basis of an interpretivist epistemological position. The philosophical assumptions underlying to this approach can be assessed with the aid of the epistemological framework of Becker and Niehaves (2007) as follows (cf., also Table 4):

1. *Assumption of a 'real world'.* The first question to answer refers to the ontological aspect: what is the object of true knowledge? Here, linguistic interpretivism assumes the existence of a real world, one which is independent of human thoughts and speech and, for this reason, exists beyond human consciousness (Kamlah and Lorenzen 1984; Weber 2004). Conceptual model statements are, thus, believed to refer to real world issues and to describe elements which are part of this real world. Hence, conceptual models are understood as elements of a design science process which aims at solving real world problem situations.

2. *Assumption of subjective influence on the process of achieving knowledge.* The second question to be addressed is that of the relationship between knowledge achieved and the object of knowledge. According to linguistic interpretivism, specific importance is attached to the influence of subjects in the process of achieving knowledge; knowledge is seen as subject-related. According to Kamlah and Lorenzen (1984), the main factor determining this subjective influence is the particular language a subject applies. This means, on the one hand, that language is applied to reconstruct a real-world experience. On the other hand, language also shapes the way a specific real world situation is perceived. Thus, language provides the categories and concepts which reconstruct cognition. In practical terms, this means, for instance, that a more elaborate vocabulary would allow a more precise perception and a more precise verbal/linguistic reconstruction of what is perceived. It has often been stated that Inuit tribes, for example, have more than 60 terms to differentiate between the various types of 'snow' that they encounter. Consequently, someone using such a conceptual repertoire would have distinctly different perceptions of the nature of 'snow', compared to someone with a less elaborate language. Furthermore, since languages are regularly shared within a linguistic community, linguistic interpretivism entails acquiring knowledge as a social process. In this sense, it follows the tradition of interpretivism (Burrell and Morgan 1979; Klein and Myers 1999; Weber 2004). Against this background, a conceptual model statement can be understood mainly as a linguistic construction of a real-world phenomenon.

3. *Consensual agreement among linguistic community members as underlying concept of truth.* The third relevant aspect is the underlying concept of truth. In this respect, linguistic interpretivism provides a criterion of truth that centers around a notional group of individuals who use the same language (consisting of names, definitions or speech artefacts), the *linguistic community*. A certain (conceptual model) statement is considered as true if every member of the relevant linguistic community could potentially consider the statement as true. However, the process of interpersonal verification must be applied in order to achieve the necessary expertise to evaluate whether a certain statement is true. In this regard, conceptual models contain formalized linguistic statements which are to be tested for truth in combination with additional empirical research methods. This is conducted through members of a linguistic community to obtain a consensus. According to Kamlah and Lorenzen (1984), the main instruments are observations, experiments, interviews and the interpretation of texts. The details of the process of interpersonal verification are discussed in the following Section 4.3.

I	*What is the object of true knowledge? (ontological aspect)*	A world exists independently of human cognition, for instance, independent of thought and speech processes (also referred to as ontological realism).
II	*What is the relationship between knowledge achieved and the object of knowledge?*	The relationship between cognition and the object of cognition is determined by the subject (also referred to as constructivism or subjectivism).
III	*What is the underlying concept of truth?*	A statement is considered as true if every member of the relevant linguistic community could potentially consider this statement as true by conducting an interpersonal verification.

Table 4: Philosophical assumptions of a linguistic interpretivist position

Hence, linguistic interpretivism is characterized by a specific interpretivist epistemology, which is influenced mainly by the language-oriented philosophy of Kamlah and Lorenzen (1984). The question arises of what constitutes the elements and practical conceptual model evaluation procedures when adopting a linguistic interpretivist perspective.

4.2 Terms as the basic elements of interpersonal verification

The basic starting point for linguistic interpretivism is the incremental (re)construction of languages. This process aims at a semantic foundation of languages in order to make speech artefacts and statements intelligible to its addressees (potential linguistic community members). On the basis of natural language and practical acts, linguistic interpretivism (re)introduces words and fixes their meaning. Hence, this approach provides the theoretical basis for analyzing terminological systems, such as domain languages which are referred to and used within conceptual models and conceptual model statements. The (re)construction of terms and concepts is carried out in three steps (Kamlah and Lorenzen 1984):

4. *Exemplary introduction:* In the process of exemplary introduction, the first step is that of linking a term to an extra-linguistic entity. The meaning of the word 'book', for example, can be defined by pointing to an appropriate object and saying, "this is a book". By so doing, the word 'book' is assigned to a non-linguistic representative from the perspective of its linguistic community. However, it is a time consuming process to (re)enact all situations necessary for the compilation of a standardized language. If all parties use the same common natural language, it is considered sufficient to describe these situations in natural language terms to achieve a common understanding. In this case, the natural language serves as an explanatory language that is used for teaching the meaning of words.

5. *Stating the predicate rules:* Predicate rules are defined to further consolidate the correct usage of technical terms from the perspective of a linguistic community. A predicate rule defines the relations between the technical term and other terms. For example, it states which terms are synonymous or represent super-ordinate concepts. Predicate rules can also specify further attributes of a particular term, for instance, "this book is in English".

6. *Stipulated definition:* As the third step, the meaning of words is defined explicitly. A definition is conceived as an explanation of a certain term with the help of other already-known terms. Such a definition can be formulated in a syntactically standardized way (normative grammar) or in natural language. If natural language is applied, it is necessary to ensure that all applied terms have been previously understood by the members of the linguistic community. It is often necessary to include other technical terms in the definition as well. For instance, "A book is a set or collection of written, printed, illus-

trated or blank sheets, made of paper, parchment or other material, usually fastened together to hinge at one side and within protective covers".

Thus, a new term or concept is introduced by *explicit agreement* with respect to its usage and meaning (Kamlah and Lorenzen 1984). This agreement leads to a relationship between concept and term and is shared by a linguistic community as the *knowledge* of using this term (Kamlah and Lorenzen 1984). Since language, as a system of signs, is shared by a *linguistic community as common knowledge*, semantics and pragmatics are directly linked to each other.

4.3 The process of interpersonal verification and statement evaluation

The verification of (conceptual model) statements is based on the procedure of interpersonal verification. Linguistic interpretivism assumes that a consensus on meaning and sense within a group can only be achieved by exchanging speech artefacts (Kamlah and Lorenzen 1984). Hence, language communities are required to ensure a common understanding of terms and statements (which use known terms). One or

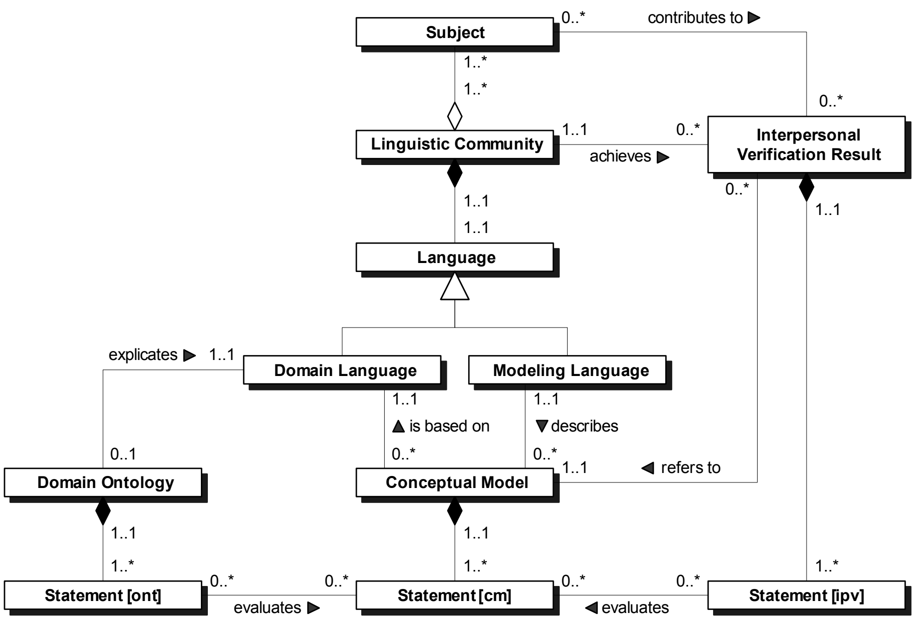

Figure 2: Linguistic community-based evaluation of a conceptual model

more subjects form a linguistic community, while a particular subject may belong to more than one linguistic community (cf., Figure 2). When evaluating conceptual models and conceptual model statements, formalized linguistic statements constituting part of a particular conceptual model are decomposed logically (deduction) until they are accessible as elemental statements for the purposes of interpersonal verification. This takes place within a group of subjects who obtain a consensus on the truth or non-truth of a certain statement. The main instruments are observation, experiments, interviewing and the interpretation of texts (Kamlah and Lorenzen 1984).

Language Construction	Exemplary introduction	Linking a term to an extra-linguistic entity, for instance, by pointing to an appropriate object and stating: "this is a book."
	Predicate rules	A predicate rule defines the relations between the technical term and other terms, for instance, "a book consists of pages."
	Stipulated definition	A definition is conceived as an explanation of a certain term with the help of other already-known terms, for instance, "a written work or composition which has been published; it consists of printed pages bound together."
Ontological Evaluation	Conceptual model decomposition	Decomposing a conceptual model into elemental statements and terms.
	Ontological mappings	Mapping elemental statements (and its terms) with statements contained by the domain ontology.
	(a) Ontology-based statement evaluation	Possibility (a): All statements (and terms) made in a conceptual model are covered by the domain ontology. A conceptual model statement may be assessed in terms of whether it is true in the context of the domain ontology.
Interpersonal Verification	(b1) Identification of semantic deficiencies	Possibility (b): Identify and elaborate the semantic deficiency of the domain ontology. This might result in a statement and/or a term not being covered by the ontology.
	(b2) Interpersonal verification	Logical decomposition of conceptual model statements until they are accessible as elemental statements for the purpose of interpersonal verification. Notional group discussions resulting in a consensus on the truth/or non-truth of a certain statement. Instruments include observations, experiments, interviews or the interpretation of texts.
	(b3) Revising domain ontology	The results of the interpersonal verification are formalized into the domain ontology to enable a closed knowledge-loop.

Table 5: Linguistic interpretivist procedure for conceptual model evaluation

The process of interpersonal verification can be an extension of the ontological evaluation of conceptual models. Against the background of an extensive discussion in the IS literature, we found that (domain) ontologies can be a valuable instrument for evaluating conceptual models (cf., Section 3). However, to conduct this type of ontology-only-based conceptual model statement evaluation, it is necessary that the content of these statements is fully covered by the domain ontology. Since domain ontologies often do not fulfill this criterion, philosophically sound evaluation procedures are required that can fill this gap. Through the process of interpersonal verification, terms, concepts and statements forming a conceptual model can be evaluated that are not fully covered by a domain-language-based ontology. A domain-language-based ontology can serve as a supportive instrument, in the case that it only partly covers conceptual model statements. For instance, having a statement like "if a 'book' is edited by an 'editor', then 'a lot of authors have contributed'", an ontology can specify and evaluate whether an entity such as 'editor' actually 'edits' an entity such as 'book'. The process of interpersonal verification can then be used to evaluate the statements as a whole or to further specify what is meant by 'a lot of authors'. The results of the procedure of interpersonal verification can be formalized into the domain ontology to enable a closed knowledge-loop. The linguistic interpretivist procedure of conceptual model evaluation is described in Table 5 and

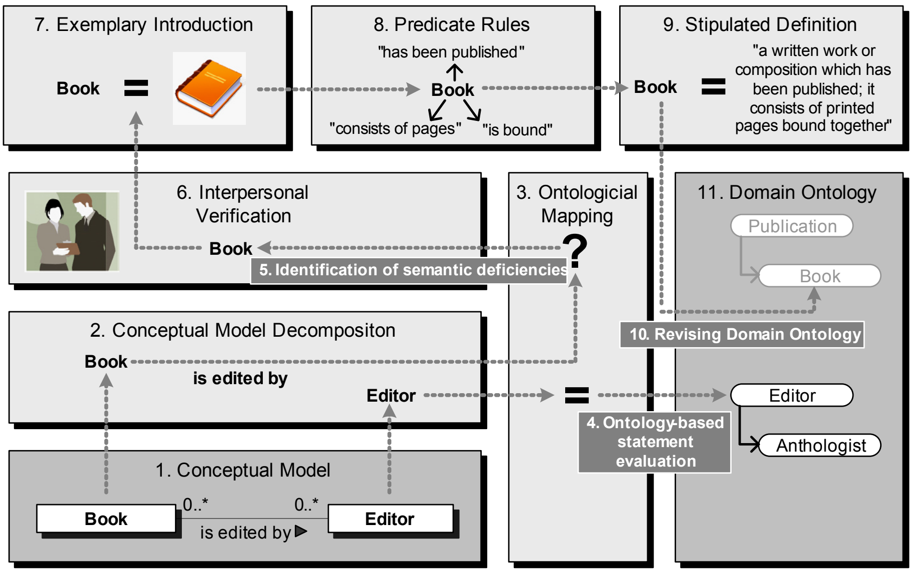

Figure 3: Example of the (re-)construction of the term 'book'

exemplified in Figure 3. In Figure 3 it is assumed that the term ‘book’ is not yet part of the domain ontology and has to be (re-) constructed.

5 Conclusions and future research perspectives

Conceptual model validity and its economic significance constituted the basis of this paper. Against this background, IS research has already intensively investigated, for instance, ontology-based evaluation approaches for conceptual modelling grammars or criteria catalogs for model evaluation. In the present paper, we have developed an evaluation framework which provides new perspectives on the conceptual model evaluation task (cf., Table 3). With the help of this framework, we provided evidence that there is a shortcoming of research on process-related aspects of conceptual model evaluation, especially, with respect to philosophically sound evaluation procedures.

In the recent IS literature, ontologies have gained much attention in the context of *modelling language evaluation*. However, they also provide a valuable contribution to the evaluation of conceptual models. In particular, domain ontologies can serve as a point of reference for conceptual model statement evaluation. However, problems arise in the case that certain statements (or the terms they contain) are not fully covered by the domain ontology. In this context, the IS literature uses the term ‘ontological incompleteness’ or ‘semantic deficiency’. Then, we argued that the ontological evaluation of conceptual models can only be a first step. Thus, we introduced the concept of interpersonal verification, a process adopted by the school of linguistic interpretivism, which can form the basis for a philosophically sound extension of ontology-based approaches to conceptual model evaluation.

Understanding and operationalizing language as the major factor for the subjects’ influence on the epistemological process of gaining knowledge, may shed new light on the paradigmatic discussion in IS research, especially between positivist and interpretivist perspectives. A major objective of Kamlah and Lorenzen’s (1984) work is to mediate between the positivist paradigm—which is regarded as understating the social and individual embeddedness of knowledge—and hermeneutic philosophy – which appears to become diffuse or arbitrary when it comes to practical considerations. Whatever the case, the formulation of a linguistic interpretivist position for conceptual model evaluation has certain limitations. For instance, a further specification of the process of interpersonal verification will be necessary. What instruments are compatible and how can they be applied? Future research will have to further operationalise language as an epistemological factor.

Moreover, our literature survey and the associated analysis of the research field identify the need for a holistic approach to determining the quality of conceptual models. Our proposal for an ontology-based linguistic evaluation of conceptual models can be regarded as a first step in this direction. Future research should complement the current findings with other approaches to conceptual model evaluation. In this respect, it may be relevant to analyse the extent to which these distinct approaches are philosophically sound, conflicting or complementary.

References

Agarwal, R., Sinha, A. P., and Tanniru, M. "Cognitive fit in requirements modeling: a study of object and process methodologies," *Journal of Management Information Systems*, (13:2), 1996, pp 137-162.

Ashenhurst, R. L. "Ontological aspects of information modeling," *Minds and Machines*, (6:3), 1996, pp 287-394.

Avison, D. E., and Wood-Harper, A. T. "Information systems development research: an exploration of ideas in practice," *The Computer Journal*, (34:2), 1991, pp 98-112.

Bajaj, A. "SMMM: a metric based framework to evaluate the scalability of multiple model methodologies," 5th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2000) at the 12th International Conference on Advanced Information Systems Engineering (CAiSE 2000), Stockholm, Sweden, 2000, pp. 1-10.

Bajaj, A. "The effect of the number of concepts on the readability of schemas: an empirical study with data models," *Requirements Engineering*, (9:4), 2004, pp 261-270.

Basili, V. R., Caldiera, G., and Rombach, H. D. "The goal question metric approach," in: *Encyclopedia of software engineering,* J. J. Marchiniak (ed.), John Wiley & Sons, New York, NY, 1994, pp. 528-532.

Bassin, K. A., Kratschmer, T., and Santhanam, P. "Evaluating software development objectively," *IEEE Software*, (15:6), 1998, pp 66-74.

Batra, D., and Davis, J. G. "Conceptual data modelling in database design: similarities and differences between expert and novice designers," *International Journal of Man-Machine Studies*, (37:1), 1992, pp 83-101.

Batra, D., Hoffler, J. A., and Bostrom, R. P. "Comparing representations with relational and EER models," *Communications of the ACM*, (33:2), 1990, pp 126-139.

Becker, J., and Niehaves, B. "Epistemological perspectives on IS research: a framework for analysing and systematizing epistemological assumptions," *Information Systems Journal*, (17:2), 2007, pp 197-214.

Benbasat, I., and Zmud, R. W. "The identity crisis within the IS discipline: defining and communicating the discipline's core properties," *MIS Quarterly*, (27:2), 2003, pp 183-194.

Benyon, D., and Skidmore, S. “Towards a tool kit for the systems analyst,” *The Computer Journal*, (30:1), 1987, pp 2-7.
Boertien, N., Steen, M. W. A., and Jonkers, H. “Evaluation of component-based development methods,” 6th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2001) at the 13th International Conference on Advanced Information Systems Engineering (CAiSE 2001), Interlaken, Switzerland, 2001, pp. 1-13.
Brinkkemper, S. “Method engineering: engineering of information systems development methods and tools,” *Information and Software Technology*, (38:4), 1996, pp 275-280.
Brinkkemper, S., Saeki, M., and Harmsen, F. “Meta-modelling based assembly techniques for situational method engineering,” *Information Systems*, (24:3), 1999, pp 209-228.
Bunge, M. *Ontology I: the furniture of the world*, D. Reidel Publishing Company, Dordrecht, 1977.
Bunge, M. *Ontology II: a world of systems*, D. Reidel Publishing Company, Dordrecht, 1979.
Burrell, G., and Morgan, G. *Sociological paradigms and organizational analysis*, Ashgate, London, 1979.
Burton-Jones, A., and Weber, R. “Understanding relationships with attributes in entity-relationship diagrams,” 20th International Conference on Information Systems (ICIS 1999), Association for Information Systems, Charlotte, NC, 1999, pp. 214-228.
Card, D. N. “Software quality engineering,” *Information and Software Technology*, (32:1), 1990, pp 3-10.
Carroll, J., and Kellogg, W. “Artifact as theory nexus: hermeneutics meets theory-based design,” 7th SIGCHI Conference on Human Factors in Computing (CHI 1989), ACM Press, Austin, TX, 1989, pp. 7-14.
Cavano, J. P., and McCall, J. A. “A framework for the measurement of software quality,” *ACM SIGMETRICS Performance Evaluation Review*, (7:3-4), 1978, pp 133-139.
Davies, I., Green, P., and Rosemann, M. “Facilitating an ontological foundation of information systems with meta models,” 13th Australasian Conference on Information Systems (ACIS 2002), Melbourne, Australia, 2002, pp. 937-947.
Davis, G. B. “Strategies for information requirements determination,” *IBM Systems Journal*, (21:1), 1982, pp 4-30.
Davis, I., Green, P., Milton, S., and Rosemann, M. “Using meta models for the comparison of ontologies,” 8th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2003) at the 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003), Velden, Austria, 2003, pp. 1-10.
Degen, W., Heller, B., Herre, H., and Smith, B. “GOL: toward an axiomatized upper-level ontology,” 2nd International Conference on Formal Ontology in Information Systems (FOIS 2001), Ogunquit, ME, 2001, pp. 34-46.
Dromey, R. G. “Cornering the chimera,” *IEEE Software*, (13:1), 1996, pp 33-43.
Evermann, J. “Towards cognitive foundation for knowledge representation,” *Information Systems Journal*, (15:2), 2005, pp 147-178.

Fadel, F. G., Fox, M. S., and Grüninger, M. "A generic enterprise resource ontology," 3rd International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WET ICE 2003), Morgantown, WV, 1994, pp. 1-12.

Fairley, R. E. *Software engineering concepts*, McGraw-Hill Book Company, New York, NY, 1985.

Farbey, B. "Software quality metrics: considerations about requirements and requirement specifications," *Information and Software Technology*, (32:1), 1990, pp 60-64.

Fenton, N. E., and Neil, M. "Software metrics: roadmap," Conference of the Future of Software Engineering at the 22nd International Conference on Software engineering (ICSE 2000), ACM Press, Limerick, Ireland, 2000, pp. 357-370.

Fitzgerald, B., Russo, N. L., and O'Kane, T. "Software development: method tailoring at Motorola," *Communications of the ACM*, (46:4), 2003, pp 65-70.

Fox, M. S., Barbuceanu, M., Grüninger, M., and Lin, J. "An organization ontology for enterprise modelling," in: *Simulating organizations: computational models of institutions and groups,* M. Prietula, K. Carley and L. Gasser (eds.), AAAI/MIT Press, Menlo Park, CA, 1998, pp. 131-152.

Gemino, A., and Wand, Y. "A framework for empirical evaluation of conceptual modeling techniques," *Requirements Engineering*, (9:4), 2004, pp 248-260.

Gemino, A., and Wand, Y. "Complexity and clarity in conceptual modeling: comparison of mandatory and optional properties," *Data & Knowledge Engineering*, (55:3), 2005, pp 301-326.

Grady, R. B., and Caswell, D. L. *Software metrics: enabling a company-wide program*, Prentice Hall, Englewood Cliffs, NJ, 1987.

Green, P., and Rosemann, M. "Integrated process modeling: an ontological evaluation," *Information Systems*, (25:2), 2000, pp 73-87.

Green, P., Rosemann, M., Indulska, M., and Manning, C. "Candidate interoperability standards: an ontological overlap analysis," *Data & Knowledge Engineering*, (62:2), 2007, pp 274-291.

Green, P. F., and Rosemann, M. "Perceived ontological weakness of process modeling techniques: further evidence," 10th European Conference on Information Systems (ECIS 2002), Uniwersytetu Gdanskiego, Gdânsk, Poland, 2002, pp. 312-321.

Greiffenberg, S. "Methoden als Theorien der Wirtschaftsinformatik," 6th International Conference Wirtschaftsinformatik (WI 2003), Physica, Dresden, Germany, 2003, pp. 947-968.

Greiffenberg, S. *Methodenentwicklung in Wirtschaft und Verwaltung*, Verlag Dr. Kovač, Hamburg, 2004.

Grünbacher, P., Halling, M., Biffl, S., Kitapci, H., and Boehm, B. W. "Integrating collaborative processes and quality assurance techniques: experience from requirements negotiation," *Journal of Management Information Systems*, (20:4), 2004, pp 9-29.

Guarino, N. "Formal ontology and information systems," 1st International Conference on Formal Ontology in Information Systems (FOIS 1998), Trento, Italy, 1998, pp. 3-15.

Guizzardi, G., Pires, L. F., and van Sinderen, M. J. "On the role of domain ontologies in the design of domain-specific visual modeling languages," 2nd Workshop on Domain-Specific Visual Languages at the 17th Annual ACM Conference on Object-Oriented

Programming, Systems, Languages and Applications (OOPSLA 2002), Seattle, WA, 2002, pp. 1-14.

Gupta, D., and Prakash, N. "Engineering methods from method requirements specifications," *Requirements Engineering*, (6:3), 2001, pp 135-160.

Halpin, T. "Data modeling in UML and ORM revisited," 4th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 1999) at the 11th International Conference on Advanced Information Systems Engineering (CAiSE 1999), Heidelberg, Germany, 1999, pp. 1-14.

Harmsen, A. F. "Situational method engineering," University of Twente, Enschede, The Netherlands, 1997.

Hevner, A. R., March, S. T., Park, J., and Ram, S. "Design science in information systems research," *MIS Quarterly*, (28:1), 2004, pp 75-105.

Heym, M., and Österle, H. "Computer-aided methodology engineering," *Information and Software Technology*, (35:6), 1993, pp 345-354.

Höfferer, P. "Achieving business process model interoperability using metamodels and ontologies," 15th European Conference on Information Systems (ECIS 2007), St. Gallen, Switzerland, 2007, pp. 1620-1631.

Hong, S., van den Goor, G., and Brinkkemper, S. "A formal approach to the comparison of object-oriented analysis and design methodologies," 26th Annual Hawaii International Conference on Systems Sciences (HICSS 1993), Maui, HI, 1993, pp. 689-698.

Hudepohl, J. P., Aud, S. J., Khoshgoftaar, T. M., Allen, E. B., and Mayrand, J. "Emerald: software metrics and models on the desktop," *IEEE Software*, (13:5), 1996, pp 56-60.

Kamlah, W., and Lorenzen, P. *Logical propaedeutic: pre-school of reasonable discourse*, University Press of America, Lanham, MD, 1984.

Kamsties, E., von Knethen, A., Philipps, J., and Schätz, B. "An empirical investigation of the defect detection capabilities of requirements specification languages," 6th International Workshop on Evaluation of Modeling Methods in System Analysis and Design (EMMSAD 2001) at the 13th International Conference on Advanced Information Systems Engineering (CAiSE 2001), Interlaken, Switzerland, 2001, pp. 1-12.

Kan, S. H., Basili, V. R., and Shapiro, L. N. "Software quality: an overview from the perspective of total quality management," *IBM Systems Journal*, (33:1), 1994, pp 4-19.

Karlsson, F., and Ågerfalk, P. J. "Method configuration: adapting to situational characteristics while creating reusable assets," *Information and Software Technology*, (46:9), 2004, pp 619-633.

Kesh, S. "Evaluating the quality of entity relationship models," *Information and Software Technology*, (37:12), 1995, pp 681-689.

Kim, J., Hahn, J., and Hahn, H. "How do we understand a system with (so) many diagrams: cognitive integration processes in diagrammatic reasoning," *Information Systems Research*, (11:3), 2000, pp 284-303.

Kim, Y.-G., and March, S. T. "Comparing data modeling formalisms," *Communications of the ACM*, (38:6), 1995, pp 103-115.

Klein, H. K., and Myers, M. D. "A set of principles for conducting and evaluating interpretive field studies in information systems," *MIS Quarterly*, (23:1), 1999, pp 67-93.

Larkin, J. H., and Simon, A. "Why a diagram is (sometimes) worth ten thousand words," *Cognitive Science*, (11:1), 1987, pp 65-99.
Lauesen, S., and Vinter, O. "Preventing requirement defects: an experiment in process improvement," *Requirements Engineering*, (6:1), 2001, pp 37-50.
Ledeczi, A., and Paka, T. "Framework for the rapid evaluation of modeling methodologies," 8th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2003) at the 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003), Velden, Austria, 2003, pp. 1-8.
Lemesle, R. "Meta-modeling and modularity: comparison between MOF, CDIF & sNets formalisms," Workshop on Model Engineering, Methods and Tools Integration with CDIF at the 13th Annual ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA 1998), Vancouver, Canada, 1998, pp. 1-11.
Levitin, A., and Redman, T. "Quality dimensions of a conceptual view," *Information Processing & Management*, (31:1), 1995, pp 81-88.
Lindland, O. I., Sindre, G., and Sølvberg, A. "Understanding quality in conceptual modeling," *IEEE Software*, (11:2), 1994, pp 42-49.
Lyytinen, K. "Ontological foundations of conceptual modeling by Boris Wyssusek: a critical response," *Scandinavian Journal of Information Systems*, (18:1), 2006, pp 81-84.
March, S. T., and Smith, G. F. "Design and natural science research on information technology," *Decision Support Systems*, (15:4), 1995, pp 251-266.
Mayer, R. J. "Models for understanding," *Review of Educational Research*, (59:1), 1989, pp 43-64.
Milton, S. K., Kazmierczak, E., and Keen, C. D. "Comparing data modelling frameworks using Chisholm's ontology," 6th European Conference on Information Systems (ECIS 1998), Aix-en-Provence, France, 1998, pp. 260-272.
Monarchi, D. E., and Puhr, G. I. "A research typology for object-oriented analysis and design," *Communications of the ACM*, (35:9), 1992, pp 35-47.
Moody, D. L. "The method evaluation model: a theoretical model for validating information systems design methods," 11th European Conference on Information Systems (ECIS 2003), Naples, Italy, 2003, pp. 1-17.
Moody, D. L., and Shanks, G. G. "What makes a good data model: evaluating the quality of entity relationship models," 13th International Conference on the Entity Relationship Approach (ER 1994), Springer, Manchester, United Kingdom, 1994, pp. 94-111
Mylopoulos, J. "Information modeling in the time of the revolution," *Information Systems*, (23:3-4), 1998, pp 127-155.
Nelson, H. J., and Monarchi, D. E. "Ensuring the quality of conceptual representations," *Software Quality Journal*, (15:2), 2007, pp 213-233.
Nuseibeh, B., Finkelstein, A., and Kramer, J. "Method engineering for multi-perspective software development," *Information and Software Technology*, (38:4), 1996, pp 267-274.
Nysetvold, A. G., and Krogstie, J. "Assessing business processing modeling languages using a generic quality framework," 10th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2005) at the 17th International

Conference on Advanced Information Systems Engineering (CAiSE 2005), Porto, Portugal, 2005, pp. 545-556.
Object Management Group "Meta object facility (MOF) core specification, version 2.0," 2006.
Odell, J. "Meta-modeling," Workshop on Metamodeling in OO at the 10th Annual ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA 1995), Austin, TX, 1995, pp. 1-6.
Orlikowski, W., and Iacono, C. "Desperately seeking the 'IT' in IT research: a call to theorizing the IT artifact," *Information Systems Research*, (12:2), 2001, pp 121-134.
Ortega, M., Pérez, M., and Rojas, T. "Construction of a systemic quality model for evaluating a software product," *Software Quality Control*, (11:3), 2003, pp 219-242.
Pfeiffer, D. "Constructing comparable conceptual models with domain specific languages," 15th European Conference on Information Systems (ECIS 2007), St. Gallen, Switzerland, 2007, pp. 876-888.
Pfeiffer, D., and Gehlert, A. "A framework for comparing conceptual models," 1st International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2005) at the 24th International Conference on Conceptual Modeling (ER 2005), Klagenfurt, Austria, 2005, pp. 108-122.
Pfeiffer, D., and Niehaves, B. "Evaluation of conceptual models: a structuralist approach," 13th European Conference on Information Systems (ECIS 2005), Regensburg, Germany, 2005, pp. 1-12.
Preece, A. D. "Toward a quality assessment framework for knowledge-based systems," *Journal of Systems and Software*, (29:3), 1995, pp 219-234.
Ralyté, J., and Rolland, C. "An assembly process model for method engineering," 13th International Conference on Advanced Information Systems Engineering (CAiSE 2001), Interlaken, Switzerland, 2001, pp. 267-283.
Rawashdeh, A., and Matalkah, B. "A new software quality model for evaluating COTS components," *Journal of Computer Science*, (2:4), 2006, pp 373-381.
Recker, J., and Rosemann, M. "Understanding the process of constructing scales inventories in the process modelling domain," 15th European Conference on Information Systems (ECIS 2007), St. Gallen, Switzerland, 2007, pp. 2014-2025.
Rosemann, M., and Green, P. F. "Integrating Multi-Perspective Views into Ontological Analysis," 21th International Conference on Information systems (ICIS 2000), Brisbane, Australia, 2000, pp. 618-627.
Rosemann, M., and Green, P. F. "Developing a meta model for the Bunge-Wand-Weber ontological constructs," *Information Systems*, (27:2), 2002, pp 75-91.
Rosemann, M., Green, P. F., and Indulska, M. "Towards an enhanced methodology for ontological analyses," 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004), Springer, Riga, Latvia, 2004, pp. 112-121.
Rosemann, M., and zur Mühlen, M. "Evaluation of workflow management systems: a meta model approach," *The Australian Journal of Information Systems*, (6:1), 1998, pp 103-116.
Rossi, M., and Brinkkemper, S. "Complexity metrics for systems development methods and techniques," *Information Systems*, (21:2), 1996, pp 209-227.

Saeki, M. “Object-oriented meta modeling,” 14th International Conference on Object-Oriented and Entity-Relationship Modeling (OOER 1995), Gold Coast, Australia, 1995, pp. 250-259.

Schütte, R., and Rotthowe, T. “The guidelines of modeling: an approach to enhance the quality in information models,” 17th International Conference on Conceptual Modeling (ER 1998), Singapore, 1998, pp. 240-254

Seffah, A., Donyaee, M., Kline, R. B., and Padda, H. K. “Usability measurement and metrics: a consolidated model,” *Software Quality Journal*, (14:2), 2006, pp 159-178.

Shanks, G., Tansley, E., and Weber, R. “Using ontology to validate conceptual models,” *Communications of the ACM*, (46:10), 2003, pp 85-89.

Siau, K. “The psychology of information modeling,” in: *Advanced topics in database research,* K. Siau (ed.), Idea Group Publishing, Hershey, PA, 2003, pp. 106-118.

Siau, K., and Cao, Q. “Unified modeling language (UML): a complexity analysis,” *Journal of Database Management*, (12:1), 2001, pp 26-34.

Siau, K., and Rossi, M. “Evaluation of information modeling methods: a review,” 31st Annual Hawaii International Conference on System Sciences (HICSS 1998), Big Island, HI, 1998, pp. 314-322.

Siau, K., and Tan, X. “Improving the quality of conceptual modeling using cognitive mapping techniques,” *Data & Knowledge Engineering*, (55:3), 2005, pp 343-365.

Siau, K., Wand, Y., and Benbasat, I. “A psychological study on the use of relationship concept: some preliminary findings,” 7th International Conference on Advanced Information Systems Engineering (CAiSE 1995), Jyväskylä, Finland, 1995, pp. 341-354.

Siau, K., Wand, Y., and Bensabat, I. “When parents need not have children: cognitive biases in information modeling,” 8th International Conference on Advances Information Systems Engineering (CAiSE 1996), Heraklion, Greece, 1996, pp. 402-420.

Sommerville, I. *Software engineering*, (6th ed.), Addison-Wesley, Harlow, 2001.

Song, X., and Osterweil, L. J. “Toward objective, systematic design-method comparisons,” *IEEE Software*, (9:3), 1992, pp 43-53.

Strahringer, S. *Metamodellierung als Instrument des Methodenvergleichs: Eine Evaluierung am Beispiel objektorientierter Analysemethoden*, Shaker, Aachen, 1996.

Studer, R., Benjamins, V. R., and Fensel, D. “Knowledge engineering: principles and methods,” *Data & Knowledge Engineering*, (25:1), 1998, pp 161-197.

Sugumaran, V., and Storey, V. C. “The role of domain ontologies in database design: an ontology management and conceptual modeling environment,” *ACM Transactions on Database Systems*, (31:3), 2006, pp 1064-1094.

Topi, H., and Ramesh, V. “Toward an extended framework for human factors research on data modeling,” in: *Advanced topics in database research,* K. Siau (ed.), Idea Group Publishing, Hershey, PA, 2004, pp. 188-216.

Túbio, O. D., Fernández, M. L., and Sánchez-Capuchino, A. M. M. “On the capability of analysis techniques in requirements engineering,” 4th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 1999) at the 11th International Conference on Advanced Information Systems Engineering (CAiSE 1999), Heidelberg, Germany, 1999, pp. 1-12.

Uschold, M. "Knowledge level modelling: concepts and terminology," *The Knowledge Engineering Review*, (13:1), 1998, pp 5-29.
Uschold, M., and Grüninger, M. "Ontologies: principles, methods and applications," *Knowledge Engineering Review*, (11:2), 1996, pp 93-155.
Wallace, D. R., and Fujii, R. U. "Software verification and validation: an overview," *IEEE Software*, (6:3), 1989, pp 10-17.
Wand, Y. "Ontology as a foundation for meta-modelling and method engineering," *Information and Software Technology*, (38:4), 1996, pp 281-287.
Wand, Y., Monarchi, D. E., Parsons, J., and Woo, C. C. "Theoretical foundations for conceptual modelling in information systems development," *Decision Support Systems*, (15:4), 1995, pp 285-304.
Wand, Y., Storey, V. C., and Weber, R. "An ontological analysis of the relationship construct in conceptual modeling," *ACM Transactions on Database Systems*, (24:4), 1999, pp 494-528.
Wand, Y., and Wang, R. Y. "Anchoring data quality dimensions in ontological foundations," *Communications of ACM*, (39:11), 1996, pp 86-95.
Wand, Y., and Weber, R. "An ontological model of an information system," *IEEE Transactions on Software Engineering*, (16:11), 1990, pp 1282-1292.
Wand, Y., and Weber, R. "On the ontological expressiveness of information systems analysis and design grammars," *Journal of Information Systems*, (3:4), 1993, pp 217-237.
Wand, Y., and Weber, R. "Information systems and conceptual modeling: a research agenda," *Information Systems Research*, (13:4), 2002, pp 363-376.
Weber, R. "Toward a theory of artifacts: a paradigmatic base for information systems research," *Journal of Information Systems*, (1:1), 1987, pp 3-19.
Weber, R. "Conceptual modelling and ontology: possibilities and pitfalls," *Journal of Database Management*, (14:3), 2003/07//Jul-Sep2003, 2003a, pp 1-20.
Weber, R. "Still desperately seeking the IT artifact," *MIS Quarterly*, (27:2), 2003b, pp iii-xi.
Weber, R. "The rhetoric of positivism versus interpretivism," *MIS Quarterly*, (28:1), 2004, pp iii-xii.
Wynekoop, J. L., and Russo, N. L. "Studying system development methodologies: an examination of research methods," *Information Systems Journal*, (7:1), 1997, pp 47-65.
Wyssusek, B. "Ontological foundations of conceptual modelling reconsidered: a response," *Scandinavian Journal of Information Systems*, (18:1), 2006, pp 139-152.
Zuniga, G. L. "An ontology of economic objects: an application of Carl Menger's ideas," *American Journal of Economics and Sociology*, (58:2), 1999, pp 299-312.