

## Association for Information Systems AIS Electronic Library (AISeL)

---

ECIS 2005 Proceedings

European Conference on Information Systems  
(ECIS)

---

2005

# Evaluation of Conceptual Models - A Structuralist Approach

Daniel Pfeiffer

*European Research Center for Information Systems, [daniel.pfeiffer@ercis.de](mailto:daniel.pfeiffer@ercis.de)*

Bjorn Niehaves

*European Research Center for Information Systems, [bjoern.niehaves@ercis.de](mailto:bjoern.niehaves@ercis.de)*

Follow this and additional works at: <http://aisel.aisnet.org/ecis2005>

---

### Recommended Citation

Pfeiffer, Daniel and Niehaves, Bjorn, "Evaluation of Conceptual Models - A Structuralist Approach" (2005). *ECIS 2005 Proceedings*. 43. <http://aisel.aisnet.org/ecis2005/43>

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# EVALUATION OF CONCEPTUAL MODELS – A STRUCTURALIST APPROACH

Pfeiffer, Daniel, European Research Center for Information Systems (ERCIS), Leonardo-Campus 3, 48149 Münster, Germany, daniel.pfeiffer@ercis.de

Niehaves, Björn, European Research Center for Information Systems (ERCIS), Leonardo-Campus 3, 48149 Münster, Germany, bjoern.niehaves@ercis.de

## Abstract

*The quality and thus the validation of conceptual models are of high economic importance. However, only little empirical work has focused on their evaluation so far. This raises the question whether a holistic approach to determining the quality of conceptual models is available yet. In order to describe the current state of research and to expose the so far neglected research fields we develop a two dimensional framework. With the help of this framework we can identify a notable shortcoming on conceptual model evaluation. Contrary to models on theories a lot of empirical work has been performed. Therefore we apply the structuralist approach from philosophy of science in order to develop an inner structure of conceptual models. Based on these findings we deduce the structural requirements that conceptual models shall meet. We explain the practical implications of our proposal and sketch an outlook to future scientific inquiries.*

*Keywords: Structuralism, conceptual model, model quality, evaluation.*

# 1 INTRODUCTION

Since mid of the 70's, conceptual models have been employed to facilitate and systematize the process of information systems engineering. A remarkable number of modeling languages and methods have been proposed aiming at a more efficient and effective software development (Mylopoulos 1998, Wand et al. 1995). In the beginning of the 90's, accompanied by new findings in management science, the positive experiences with conceptual models were transferred from information systems engineering to organizational design. This established conceptual models as a widely-used mean for eliciting customer requirements and documenting the project progress of a software system as well as for describing the business processes and corporate structures in an organization (Shanks & Tansley & Weber 2003). The significance of conceptual models is embodied by the proposal to define them as the core of the Information Systems (IS) discipline (Weber 2003).

The quality of conceptual models has gained an immense impact on other IT artifacts. Software systems are often based on explicit requirement specifications in form of conceptual models. The adequacy of these specifications with regard to the represented application domain determines the acceptability and usability of software systems (Lauesen & Vinter 2000). An incorrect description of the application domain will lead to problems in the implemented software system and to delays in the project progress. Likewise, the success of a reorganization project is influenced by the adequacy of the underlying organizational models. A problem analysis based on faulty models can lead to wrong and in the end very cost intensive decisions. By this means, the quality of conceptual models has reached a high economic importance.

The scientific and practical significance of conceptual models encouraged many IS researchers to engage in the evaluation of these artifacts. During the last years numerous research efforts have been undertaken in order to develop criteria catalogs to evaluate the quality of conceptual models. Nonetheless, only little empirical work has focused on their evaluation (Wand & Weber 2002, p. 370). This raises the first two research questions we want to address in the course of this paper: What are the elements of a holistic approach to evaluating IT artifacts (Q1)? Is a holistic approach to determining the quality of a conceptual model available yet (Q2)?

In this paper we approach these issues from a theoretical perspective. Theories are the artifacts of scientific inquiries. In the realm of science, theories are subject to an empirical validation. Depending on the results of this assessment, they are rejected (falsified) or temporary accepted. Contrary to models on theories a lot of empirical work has been performed. This motivates the idea of transferring the insights gained by philosophy of science also to the evaluation of models. Thereof, the third research question derives: How can structuralism contribute to a holistic approach to evaluating conceptual models (Q3)?

The paper proceeds as follows: In the next section we will develop a framework comprising a holistic approach to evaluating IT artifacts. With the help of this framework, we will classify existing research results on this subject to identify their shortcomings and to expose the neglected research fields so far. In the following section we will introduce the basic ideas of the structuralist program towards theory design. Based on the gained insights we will derive the internal structure of a conceptual model in the subsequent section. In this context we will explain the benefits of such a structure for the evaluation of conceptual models. The paper concludes with a short summary of the results achieved and an outlook to future scientific inquiries.

## 2 A FRAMEWORK TO EVALUATE IT ARTIFACTS

The evaluation of IT artifacts has been an active research field for the last 20 years. During that time, manifold approaches to assessing the quality of artifacts have been proposed. For describing the

current state of research and identifying so far neglected research fields, it is helpful to systematize these approaches with the help of a conceptual framework. The framework proposed spans two dimensions.

The first dimension addresses the IT artifacts. Catalogs of artifacts have been suggested by many authors in the IS field (Brinkkemper & Saeki & Harmsen 1999, Hevner et al. 2004, Wand & Weber 2002). In this paper we follow March and Smith, who identify four different artifacts in IS research (March & Smith 1995). Table 1 explains these artifacts (Hevner et al. 2004, p. 78 f., Wand & Weber 2002, p. 364).

<b>A1</b>	<b>Constructs</b>	Constructs provide the language concepts in which the problem is described and the solution is communicated.
<b>A2</b>	<b>Methods<sup>1</sup></b>	Methods explicate the processes of how to solve a problem and offer guidance how to search the solution space.
<b>A3</b>	<b>Models</b>	Models utilize the constructs to represent an application domain and express the problem and solution space.
<b>A4</b>	<b>Instantiations</b>	Instantiations constitute the realization of constructs, models and methods in a working system.

Table 1. *Artifact dimension of the framework to evaluate IT artifacts.*

The second dimension covers the different elements of a holistic approach to evaluating IT artifacts. It addresses the issue which aspects an instrument to assessing the quality of IT artifacts has to consider (Q1). These three aspects are explicated in table 2.

<b>E1</b>	<b>Structure of the artifact</b>	The structure of the artifact determines the configurational characteristics necessary to enable the evaluation of the IT artifact. Based on this structure all required information about the artifact can be deduced. The structure represents the information space the artifact spans.
<b>E2</b>	<b>Evaluation criteria</b>	The relevant aspects of the assessment are stipulated by the evaluation criteria. These criteria pin down the dimensions of the information space which are relevant for determining the utility of the artifact. These criteria can differ on the purpose of the evaluation.
<b>E3</b>	<b>Evaluation approach</b>	The procedure how to practically test an artifact is described by the evaluation approach. It defines all roles concerned with the assessment and the way of handling the evaluation. The result is a decision whether or not the artifact meets the evaluation criteria based on the available information.

Table 2. *Evaluation dimension of the framework to evaluate IT artifacts.*

The complete framework with its two dimensions and examples of corresponding research results is shown in table 3. The structure of the **constructs** (A1 / E1) is defined by a language based meta model (Guizzardi & Pires & Sinderen 2002). Without a meta model the syntax and the semantics of the constructs would be ambiguous. Every attempt of evaluation would depend significantly on the interpretations of the persons involved, if not even fail. Wand and Weber have proposed the ontology of Bunge as a theoretical anchor for IS constructs (Wand & Weber 1990). Other authors have recommended alternative ontologies, including General Ontological Language (Degen et al. 2001) or Chrisholm Ontology (Milton & Kazmierczak & Keen 1998), as a theoretical foundation of modeling languages. Based on their ontology Wand and Weber have defined the criteria (A1 / E2) construct deficit, construct overload, construct redundancy, and construct excess as evaluation measures (Wand

<sup>1</sup> There is a second notion of a method in Information Systems. In this notion a method is defined as a composite comprising a process as well as the constructs to describe the results of its application (Brinkkemper 1996, Greiffenberg 2003, Nuseibeh & Finkelstein & Kramer 1996).

& Weber 2002). A construct deficit exists, if a modeling language does not provide all constructs available in the ontology. Whereas a construct overload is present, if a concept of a modeling language can be mapped to more than one concept of the ontology. Construct redundancy refers to the case, when two concepts of the modeling language represent the same concept of the ontology. If the language elements have no ontological counterpart, a construct excess exists. Rosemann, Green, and Indulska proposed with the ontological analysis a detailed procedure (A1 / E3) how to perform an ontological evaluation of constructs (Rosemann & Green & Indulska 2004). They also consider the organizational conditions that have to be met in order to assess a modeling language appropriately.

	Structure of the artifact (E1)	Evaluation criteria (E2)	Evaluation approaches (E3)
<b>Constructs (A1)</b>	<ul style="list-style-type: none"> <li>language based meta model</li> </ul>	<ul style="list-style-type: none"> <li>construct deficit</li> <li>construct overload</li> <li>construct redundancy</li> <li>construct excess</li> </ul>	<ul style="list-style-type: none"> <li>ontological analysis</li> </ul>
<b>Methods (A2)</b>	<ul style="list-style-type: none"> <li>process based meta model</li> <li>intended applications</li> <li>conditions of applicability</li> <li>products and results of the application</li> <li>reference to constructs</li> </ul>	<ul style="list-style-type: none"> <li>appropriateness</li> <li>completeness</li> <li>consistency</li> </ul>	<ul style="list-style-type: none"> <li>laboratory research</li> <li>field inquiries</li> <li>surveys</li> <li>case studies</li> <li>action research</li> <li>practice descriptions</li> <li>interpretative research</li> </ul>
<b>Models (A3)</b>		<ul style="list-style-type: none"> <li>construction adequacy</li> <li>language adequacy</li> <li>economic efficiency</li> <li>clarity</li> <li>comparability</li> <li>systematic design</li> </ul>	
<b>Instantiations (A4)</b>	<ul style="list-style-type: none"> <li>executable implementation in a programming language</li> <li>reference to a design model</li> <li>reference to a requirement specification</li> <li>reference to the documentation</li> <li>reference to quality management documents</li> <li>reference to configuration management documents</li> <li>reference to project management documents</li> </ul>	<ul style="list-style-type: none"> <li>functionality</li> <li>usability</li> <li>reliability</li> <li>performance</li> <li>supportability</li> </ul>	<ul style="list-style-type: none"> <li>code inspection</li> <li>testing</li> <li>code analysis</li> <li>verification</li> </ul>

Table 3. Framework to evaluate IT artifacts with corresponding research results.

The constitutive part of a **method** is given by a process model, which describes how to reach the objective of the method. Greiffenberg develops a method structure (A2 / E1) which aims at fostering the testability of a method (Greiffenberg 2003). Within this structure a process model has to explicitly state 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. Hence Greiffenberg spans with his structure the required information space to evaluate methods. Based on Brinkkemper, Saeki and Harmsen (1999), Greiffenberg has designed a criteria catalogue (A2 / E2) to evaluate methods. The criterion of appropriateness 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 consistency of a method is satisfied, if all method elements are

mutually compatible. The evaluation of methods (A2 / E3) is often done in form of field inquiries, surveys, case studies and action research (Wynekoop & Russo 1997). Laboratory research, practice descriptions and interpretative research are hardly used.

The Software Engineering community widely agrees upon the elements that an **instantiation** (A4 / E1) must comprise. An executable implementation in a programming language must always come along with a requirements specification and a design model (Sommerville 2001, p. 55 ff.). A good documentation as well as the reference to the configuration management, quality management, and project management files influences the verifiability of the software system. Elaborated software process models facilitate such a structure of instantiations. The FURPS model is a prominent representative of the many criteria catalogues (A4 / E2) concerned with determining software quality (Grady & Caswell 1987). The criteria functionality, usability, reliability, performance, and supportability build the basis for the software quality assessment. Code inspections, testing, code analysis, and verification are established evaluation approaches (A4 / E3) to guarantee a high software quality (Fairley 1985). Software companies engage quality engineers and software testers to conduct these procedures.

Our interest in this paper focuses mainly on the quality of **models**. A lot of efforts were spent in order to develop criteria catalogues (A3 / E2) to determine the model quality (Kesh 1995, Levitin & Redman 1995, Lindland & Sindre & Sølvsberg 1994, Moody & Shanks 1994). The Guidelines of Modeling (GoM) with the measures construction adequacy, language adequacy, economic efficiency, clarity, comparability and systematic design are only one proposal among many others (Schütte & Rotthowe 1998). However, only little research has been performed on the structure of models (A3 / E1) and their evaluation approaches (A3 / E3). So far these research fields have been omitted by the IS community (Q2). The reason why the scientific debate on structure and evaluation approaches has missed out models yet is in our opinion twofold. Models focus on a certain application domain and are mostly less general than constructs and methods. Thus, at first sight it seems to be less worthwhile to deal with their evaluation. However, in contrast they are less concrete than implementations. Therefore, their importance can easily be overseen in a practical project situation.

In the following we will focus on the structure of models, because we consider the information space as more fundamental and thus prior to evaluation criteria and approach. We will propose an inner structure for conceptual models as a first step towards a holistic instrument to evaluating conceptual models. For this purpose we will use the structuralism as a theoretical foundation.

### **3 STRUCTURALISM AS A PROGRAM IN PHILOSOPHY OF SCIENCE**

The idea of structuralism arose in the year 1971 with Joe Sneed's seminal book "Logical Structure of Mathematical Physics". Since then, with more than 700 publications structuralism has become a well established and broadly applied approach in philosophy of science to describing theories (for a detailed bibliography see Diederich & Ibara & Mormann 1989, Diederich & Ibara & Mormann 1994). Structuralism uses a specific frame of concepts to model the inner structure of science. Thereby, it considers science as a huge und complex conceptual network consisting of interdependent theories. Structuralism uses the formal language of elementary set theory as description mechanism. For covering the pragmatic aspects of science it applies informal ways of representation as well. Structuralism has matured to a complex and formal scientific framework. Thus, for the sake of comprehensibility, we will utilize plain English to explain its concepts in a strongly simplified form.

Structuralism belongs to the "non-statement view" in philosophy of science (Balzer & Moulines & Sneed 1987). The non-statement view assumes that the choice of the concrete axioms of a theory is a rather insignificant question. So long as the axioms cover the interesting phenomena of the domain they are arbitrary. Structuralism therefore focuses on the inner assembly of a theory and the elements it comprises.

From a structuralist perspective the concept “theory” is polysemic (Moulines 2002, p. 4). Theories exist at different levels of granularity. The simplest form of a theory is a **theory element**, which also constitutes the basic building block of structuralism. A theory element is the smallest unit that colloquially is called a theory. Theories are never completely independent. They **form theory nets** on a medium level and **theory holons** on the large.

Theory elements are described by axioms, preferably expressed in a formal language. There are two kinds of axioms, the **frame conditions** and the **substantial laws**. The first do not state any information about the world but provide the formal vocabulary to describe it. The latter contain at least one assertion about world via the previously defined terminology. The class of structures that only meets the frame conditions is denoted as **potential models**. The class of structures which additionally satisfies the substantial laws is called **actual models**.

The different models of a theory are never isolated from each other. They mutually restrict certain admissible combinations of potential models. These cross-connections within one theory are called **constrains**. Also theory elements themselves are never independent from each other. They form a net out of related, more general or more special theory elements. The connections between them are called **inter-theoretic links**.

The frame conditions of a theory are formulated with two kinds of concepts. The first group consists of theory-immanent concepts which can only be determined if the fundamental laws of the theory T hold. They are called **T-theoretical concepts**. The second group contains the so called **T-non-theoretical concepts** which stem from other, linked theories and do not depend on the correctness of the theory T. The potential models are structures which contain T-theoretical as well as T-non-theoretical concepts. The **partial potential models** consist only out of T-non-theoretical concepts and can therefore be considered as truncated potential models. The distinction between theoretical and non-theoretical concepts is closely connected with the question of how to provide an empirical basis for a theory (Balzer & Moulines & Sneed 1987, p. 48). The empirical basis contains a frame of concepts by means of which the theory can be controlled. It serves as a fundament to determine, whether the theory functions or not. Therefore, the empirical basis should only contain T-non-theoretical concepts to make sure that an empirical validation of a theory does not already presume the correctness of the theory.

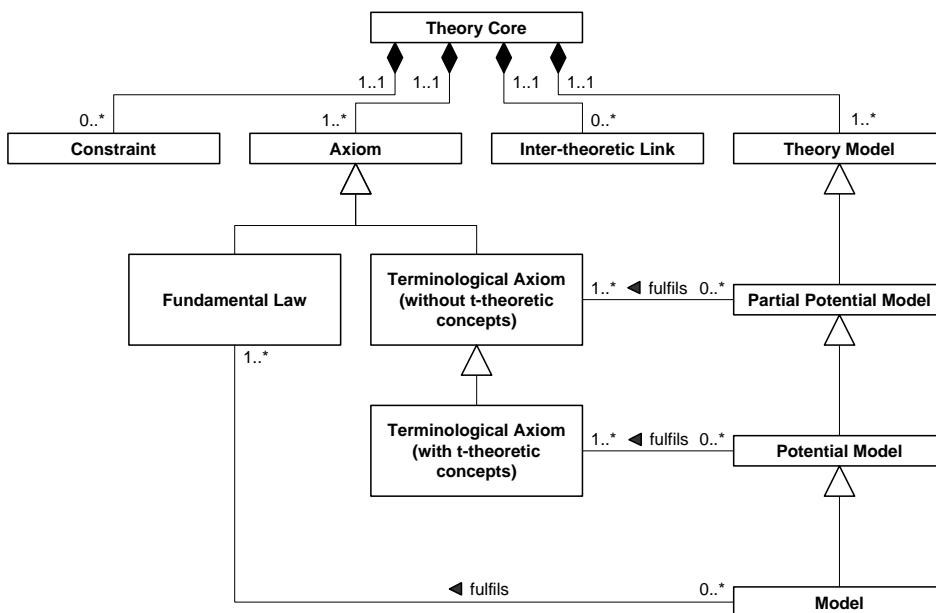


Figure 1. Structure of a theory core (Greiffenberg & Schermann 2003).

Actual models, potential models, partial potential models, constraints, and inter-theoretic links constitute the (formal) **theory core**. Figure 1 summarizes the gained insights about the structure of the theory core in the notation of the Unified Modeling Language (UML) (Object Management Group 2003).

The axioms of a theory refer to a certain real world domain, to the domain of **intended applications**. The empirical claim of a theory can be condensed to the statement that the intended applications can be subsumed under the theory core. Intended applications and theory core together build a theory element. The structure of a theory element is explained in figure 2.

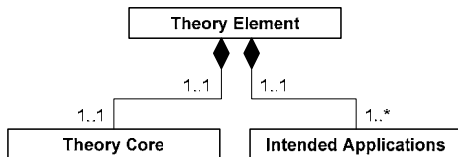


Figure 2. Structure of a theory element.

After this brief introduction into the notion of structuralism we can now transfer these ideas about the structure of theories to conceptual models.

## 4 THE INNER STRUCTURE OF CONCEPTUAL MODELS

A **conceptual model** is a representation of an application domain expressed in a semi-formal, mostly graphical language with the purpose of facilitating information systems development and organizational design (Evermann & Wand 2001, Schütte & Rothowe 1998). In comparison to theories conceptual models (not to be confused with actual, potential or partial potential models in structuralism) are less general in nature. The domain they represent is often restricted to a sole company or department. They do not cover the phenomena of a whole scientific field or parts of it as many empirical theories do. Nonetheless, also conceptual models require a certain inner structure to provide all information about the artifact necessary for its evaluation.

Conceptual models are not described in terms of mathematical axioms but in a semi-formal, graphical way. Their semantics results, on the one hand, from the given graphical arrangement of the model elements and their connections among each other. This is called the **model element structure**. On the other hand, the proposition of the conceptual model is influenced by the meaning of the modeling constructs and the application domain terms which are applied in the model. This second factor is denoted here as **terminological structure**. The model element structure and the terminological structure both characterize **model statements**.

The element structure of a conceptual model contains its actual proposition about the application domain. Thus, the element structure is comparable to the fundamental law of a theory. The element structure uses the terminological structure in order to express the content of a model. The particular arrangement of the model elements and their mutual relations expressed via modeling language constructs and application domain terms represent the empirical claim of the model. By this means, the element structure can describe a certain dynamic or static pattern of the application domain.

The terminological structure of a model is closely related to the framing conditions of a theory. Both of them provide the vocabulary to make assertions about the world possible. The terminological structure of a model can contain references to other conceptual models, including language based meta-models or domain ontologies, which define the applied terms and constructs. Correspondingly, the frame conditions contain T-non-theoretic concepts that stem from other theories. Sometimes the terminological structure also depends on the correctness of the conceptual model itself, for instance when a meta-model of the Entity Relationship Model (ERM) (Chen 1976) is formulated in terms of the ERM. This case is comparable to T-theoretic concepts in a frame condition of a theory.



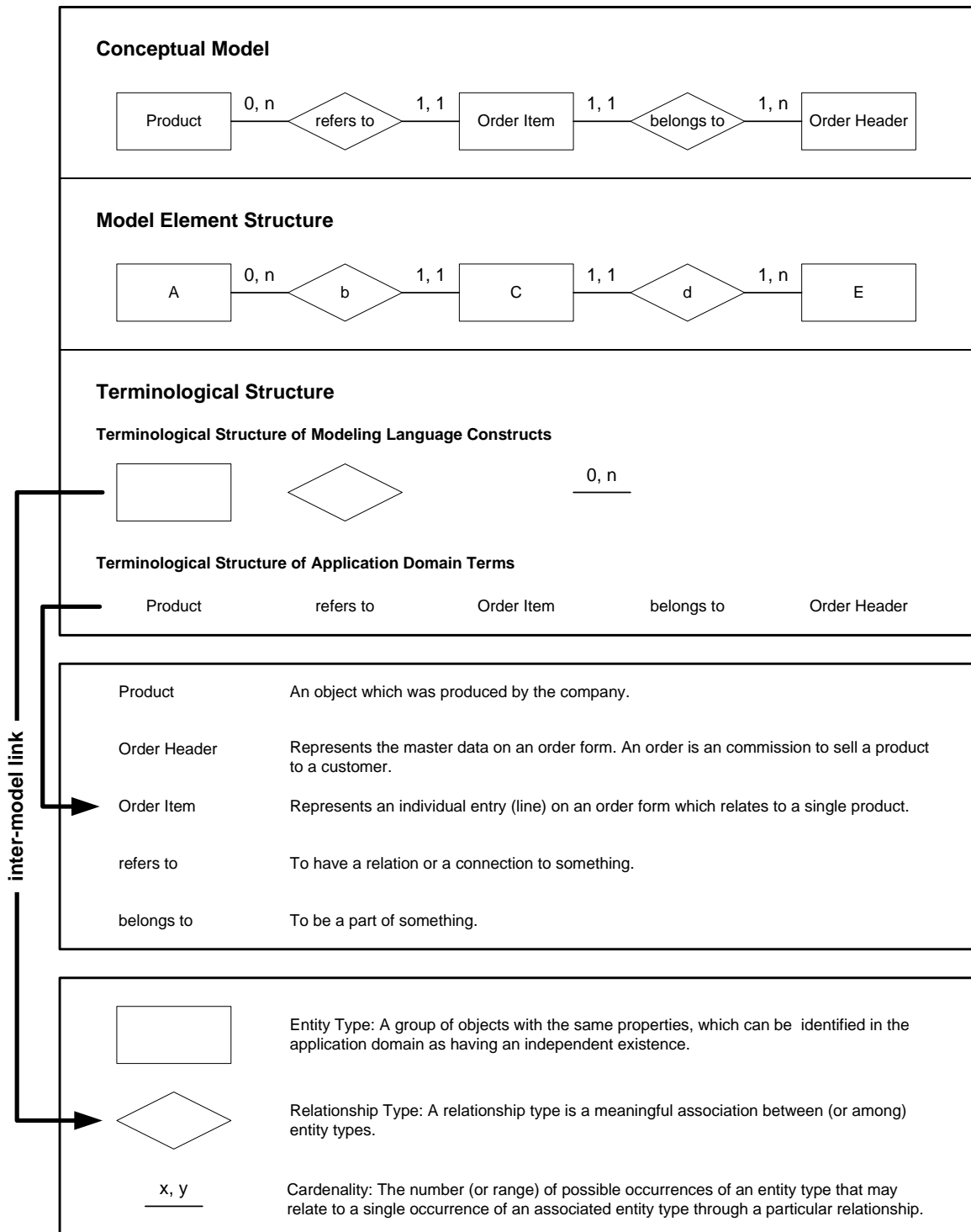


Figure 3. Model structure, terminological structure and inter-model links.

We have already described, that a not unessential part of the semantics of a conceptual model descends from a reference to the language of the application domain and the constructs of a modeling language respectively. If such a reference is missing, the terms in the models are only restrained in their meaning by their connections among each other. Thus, if a conceptual model is not associated with other models, the meaning and existence of its terms originate in the model itself. The correctness of a certain model element is then dependent on the correctness of other model elements and not on its plausibility in terms of an application domain language. In this case, the determination of the meaning

of a model element requires an assumption about the correctness of the model element structure. Since conceptual models only cover certain aspects of finite parts of the world they are not able to sufficiently restrict the meaning of all of the constructs and terms they apply. Isolated models therefore offer a wide range of possible interpretations and provoke misunderstanding, if the meaning of the terms has to be reconstructed from the connections among the model elements.

To facilitate an effective controlling of conceptual models it is crucial, that an evaluation approach does not already presuppose the trueness of the model and the existence of its concepts. Hence, conceptual models have to exhibit **inter-model links** in the style of theories to provide an empirical basis which is independent from an assumption about the correctness of the model. The inter-model links establish a mapping between the model elements and its syntactic and semantic meta-languages. The syntactic meta-language can be represented as a language based meta-model which also forms a conceptual model. The semantic meta-language is given by the application domain language and can be described for example in terms of a technical term model (Rosemann 2003) or a domain ontology (Mena et al. 1998). Beside technical term models also domain ontologies can be considered as conceptual models (Schütte & Zelewski 2002). Model element structure, terminological structure and inter-model links are illustrated in figure 3.

Suppose an application domain, as described in figure 4, where customers can place orders for certain products. An experienced data modeler will represent the “order” in this domain with the two concepts “order header” and “order item”. A domain representative who is not well trained in data modeling and who does not obtain any additional information about the model will hardly be able to understand this distinction. From the model he only gains the insight that an “order header” contains multiple “order items”. However his domain experience tells him, that an order is a coherent form. Without any inter-model links the concepts “order header” and “order item” originate in the model itself. Therefore the correctness of the model element “order header” depends on the correctness of the “order item” and vice versa. The domain representative is not able to confirm that the model is an adequate description of the application domain. Until he gains the information that an “order header” contains the master data of an order form and each “order item” stands for a line on the order form he can not come to a decision about the correctness of the model.

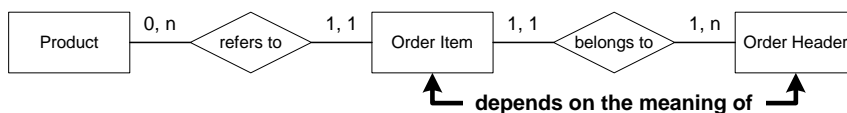


Figure 4. Entity Relationship Model (ERM) of an application domain.

Assume that in another case the term “shower” occurs in a data model. This could indicate that model represents the domain of a sanitary equipment company as well as a meteorological institution. Again, without any additional information the correctness of this model could only be evaluated depending on the correctness of the terms and constructs connected with “shower”. Theories specify their indented applications to restrain the phenomena which are covered by them. Likewise, conceptual models must state which application domain they want to represent. Their so called **intended application domain** makes sure that an evaluation takes place in the environment the conceptual model aims to describe.

The model core comprises models statements and inter-model links. A conceptual model consists out of the model core and the intended application domain. Constraints are not relevant in the context of conceptual models. Figure 5 summarizes the proposed structure.

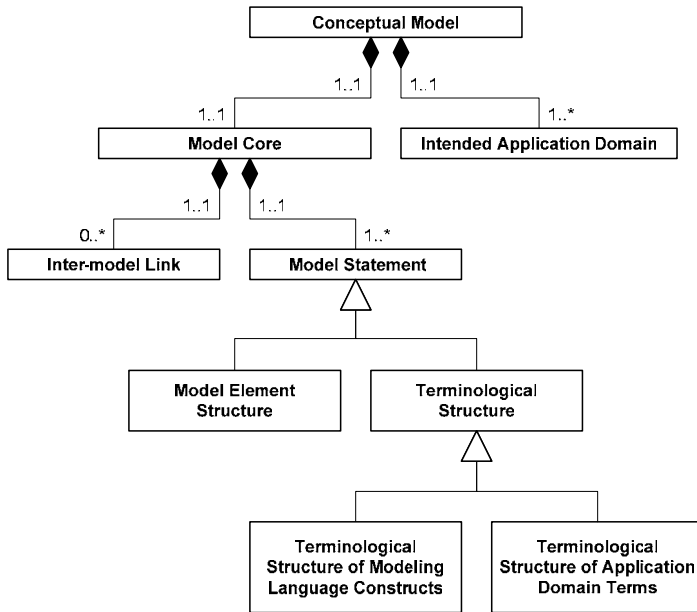


Figure 5. Structure of a conceptual model.

## 5 SUMMARY AND OUTLOOK

The high economic importance that the validation of conceptual models has reached today marked the starting point of this paper. Aiming at a holistic approach towards the evaluation of IT artifacts we developed a framework in order to reflect the current situation in IS research (Q1). With the aid of the framework we could identify two research fields which have been neglected so far (Q2). Motivated by this result we defined the focus of this paper to develop an inner structure of the artifact conceptual model. Then we employed the findings of structuralism in philosophy of science as a theoretical foundation to derive such an inner structure.

In order to meet the requirements deduced from structuralism a conceptual model has to exhibit the following characteristics (Q3):

- A conceptual model must specify the application domain it intends to represent.
- A conceptual model has to refer to the constructs of its modeling language.
- A conceptual model must explicitly state a relationship to the language of the application domain.

A modeling language (syntactic meta-language) and a domain language (semantic meta-language) enable in the role of measures the addressee of a conceptual model to come to a decision about the correctness of the artifact. In addition to the information space spanned by the internal structure, the evaluation criteria and an evaluation approach permit an assessment of the quality of the conceptual model.

These theoretically based findings have at least two practical implications. The first implication is relevant for customers and manufacturers of conceptual models. Before a customer puts an IT artifact into operation he will perform an inspection to determine its quality. A manufacturer of a conceptual model must therefore provide not only the model itself but also information about the application domain, the applied modeling language, and the application domain language in order to enable the customer to evaluate the model. The delivery of a conceptual model on its own prevents an effective verification of the artifact. The second implication addresses the producers of conceptual modeling tools. They should augment their products with the functionality to manage the association between

the conceptual model and the application domain language. Therefore, a modeling tool should be able to access domain ontologies or technical term models and (semi-)automatically establish connections between them and the model elements.

The super ordinate objective of our research was to establish a holistic approach to determining the quality of conceptual models. Our proposal for an internal structure of a conceptual model is only the first step into this direction. In our future research, we will turn towards the design of an evaluation approach for conceptual models. Thus, we will address the second research field which has been neglected so far. Furthermore, we will make the suggested internal structure of conceptual models to the subject of an empirical evaluation. Additionally, we plan to examine, whether the structure of other IT artifacts can be reconstructed in terms of structuralism and how the structure of an IT artifact influences its evaluation criteria.

## References

- Balzer, W., Moulines, C. U. and Sneed, J. D. (1987). *An Architectonic for Science - The Structuralist Program*. D. Reidel Publishing Company. Dordrecht et al.
- Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, 38 (4), p. 275-280.
- Brinkkemper, S., Saeki, M. and Harmsen, F. (1999). Meta-modelling based assembly techniques for Situational Method Engineering. *Information Systems*, 24 (3), p. 209-228.
- Chen, P. P.-S. (1976). The Entity Relationship Model - Toward a Unified View of Data. *ACM Transaction on Database Systems*, 1 (1), p. 9-36.
- Degen, W., Heller, B., Herre, H. and Smith, B. (2001). GOL: toward an axiomatized upper-level ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems*, p. 34-46, Ogunquit, Maine, USA.
- Diederich, W., Ibara, A. and Mormann, T. (1989). Bibliography of Structuralism. *Erkenntnis*, 30, p. 387-407.
- Diederich, W., Ibara, A. and Mormann, T. (1994). Bibliography of Structuralism II. *Erkenntnis*, 41, p. 403-418.
- Evermann, J. and Wand, Y. (2001). Towards Ontologically Based Semantics for UML Constructs. In *Proceedings of the 20th International Conference on Conceptual Modeling (ER 2001)* (H. S. Kunii, S. Jajodia and A. Sølvberg Ed.), p. 354-367, Yokohama, Japan.
- Fairley, R. E. (1985). *Software Engineering Concepts*. McGraw-Hill Book Company. New York et al.
- Grady, R. B. and Caswell, D. L. (1987). *Software Metrics: Enabling a Company-Wide Program*. Prentice Hall. Englewood Cliffs.
- Greiffenberg, S. (2003). Methods as theories in Information Systems, in German. In *Proceedings of the 6th International Conference Wirtschaftsinformatik (WI 2003)* (W. Uhr, W. Esswein and E. Schoop Ed.), p. 947-967, Physica, Dresden.
- Greiffenberg, S. and Schermann, M. (2003). Requirements of Information Systems theories - Part 1, in German. *WiSt*, 10, p. 613-617.
- Guizzardi, G., Pires, L. F. and Sinderen, M. J. v. (2002). On the role of Domain Ontologies in the design of Domain-Specific Visual Modeling Languages. In *Proceedings of the 2nd Workshop on Domain-Specific Visual Languages, 17th ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2002)*, Seattle.
- Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28 (1), p. 75-105.
- Kesh, S. (1995). Evaluating the quality of entity relationship models. *Information and Software Technology*, 37 (12), p. 681-689.
- Lauesen, S. and Vinter, O. (2000). Preventing Requirement Defects. In *Proceedings of the Sixth International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ 2000)*, Stockholm.

- Levitin, A. and Redman, T. (1995). Quality dimensions of a conceptual view. *Information Processing & Management*, 31 (1), p. 81-88.
- Lindland, O. I., Sindre, G. and Sølvsberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Software*, 11 (2), p. 42-49.
- March, S. T. and Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15 (4), p. 251-266.
- Mena, E., Kashyap, V., Sheth, A. and Illarramendi, A. (1998). Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In *Proceedings of the First International Conference on Formal Ontologies in Information Systems*, Trento, Italy.
- Milton, S. K., Kazmierczak, E. and Keen, C. D. (1998). Comparing Data Modelling Frameworks using Chisholm's Ontology. In *Proceedings of the 6th European Conference on Information Systems*, p. 260-272, Aix-en-Provence.
- Moody, D. L. and Shanks, G. G. (1994). What Makes a Good Data Model? Evaluating the Quality of Entity Relationship Models. In *Proceedings of the 13th International Conference on the Entity Relationship Approach (ER 1994)* (P. Loucopoulos Ed.), Manchester, United Kingdom.
- Moulines, C. U. (2002). Introduction: Structuralism as a program for modelling theoretical science. *Synthese*, 130 (1), p. 1-11.
- Mylopoulos, J. (1998). Information modeling in the time of the revolution. *Information Systems*, 23 (3-4), p. 127-155.
- Nuseibeh, B., Finkelstein, A. and Kramer, J. (1996). Method engineering for multi-perspective software development. *Information and Software Technology*, 38 (4), p. 267-274.
- Object Management Group (2003). Unified Modeling Language Specification Version 1.5. Downloaded from <http://www.omg.org/cgi-bin/doc?formal/03-03-01.pdf> on 2003/11/01.
- Rosemann, M. (2003). Preparation of Process Modeling. In *Process Management - A Guide for the Design of Business Processes* (J. Becker, M. Kugeler and M. Rosemann Ed.), p. 41-78, Springer, Berlin et al.
- Rosemann, M., Green, P. and Indulska, M. (2004). Towards an Enhanced Methodology for Ontological Analyses. In *Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004)*, Riga, Latvia.
- Schütte, R. and Rothowe, T. (1998). The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models. In *Proceedings of the 17th International Conference on Conceptual Modeling (ER 1998)* (T. W. Ling, S. Ram and M. L. Lee Ed.), p. 240-254, Singapore.
- Schütte, R. and Zelewski, S. (2002). Epistemological Problems in Working with Ontologies. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)* (N. Callaos, J. Porter and N. Rische Ed.), p. 161-167, Orlando, USA.
- Shanks, G., Tansley, E. and Weber, R. (2003). Using ontology to validate conceptual models. *Communications of the ACM*, 46 (10), p. 85-89.
- Sommerville, I. (2001). *Software Engineering*. Addison-Wesley. Harlow et al.
- Wand, Y., Monarchi, D. E., Parsons, J. and Woo, C. C. (1995). Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems*, 15 (4), p. 285-304.
- Wand, Y. and Weber, R. (1990). An Ontological Model of an Information System. *IEEE Transactions on Software Engineering*, 16 (11), p. 1282-1292.
- Wand, Y. and Weber, R. (2002). Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research*, 13 (4), p. 363-376.
- Weber, R. (2003). Still Desperately Seeking the IT Artifact. *MIS Quarterly*, 27 (2), p. iii-xi.
- Wynekoop, J. L. and Russo, N. L. (1997). Studying system development methodologies: an examination of research methods. *Information Systems Journal*, 7 (1), p. 47-65.