Where Ontology Affects Information Systems[1]

Roland Kaschek
Department of Information Systems,
Massey University, Palmerston North, New Zealand
Roland.Kaschek@ieee.org

Sergiy Zlatkin
Department of Information Systems,
Massey University, Wellington, New Zealand
szlatkin@yahoo.com

Abstract: Nicola Guarino has coined the phrase ontology driven information systems. This phrase appears to offer ontology based help and guidance for information systems development. In this paper we try to understand better whether, why and where ontology affects information systems. We briefly discuss what kind of help we would appreciate and identify domains of knowledge in information systems development that according to our point of view could benefit from such help. We briefly discuss the following five domains: universes of discourse, information systems resources, information space use, customers, and human inference making.

1. Introduction

Ontology has turned into a buzzword. Apart from fashion there are reasons to deal with ontology from an information system (IS) perspective. In the present paper we explore some of the respective issues. Unfortunately apart from [Gu98] and [Zu01] we are not aware of related work. Our methodological assumption is that we need to explicate the concepts involved in our study, i.e. information system, ontology and model. However, we do not explicitly discuss whether the quality aspects (see for them, e.g. [Po01]) of explications are met in our discussion since we believe that the concepts we are dealing with are highly debatably and subjective preferences severely impact respective choices.

Paper outline: We first, in section 2, present our conceptual model of IS, then in section 3 we discuss ontologies, introduce models and modeling and show how ontologies can be understood as particular kind of models. In section 4 we discuss the kind of help from ontology we would appreciate. We proceed in section 5 with a discussion of five domains with respect to which we believe such help would be important. We conclude the paper with a resume and the references.

---

[1] We thank Nicola Guarino and his team for a discussion of an earlier version of this paper.

2. A conceptual model of information systems

According to [H*95] representatives of the Scandinavian school of information systems (IS) and *Langefors* in particular defined the artefact part[2] (AP) of an IS from a functional perspective as technically implemented media for recording, storing, disseminating linguistic expressions and deriving linguistic expressions from these. Given an IS then its AP is assumed to interact with customers to whom it, in the best case, replies to their inquiries what they want. This interaction can be perceived as a communication, i.e., an exchange of messages. For more detail on communication theory see, e.g. [CS98]. *Hirschheim* et al in [H*95] additional to the functional perspective definition mention a structural perspective definition of IS due to *Davis* and *Olsen*. According to this definition an IS serves the purpose of individuals in an organization. The purpose is to more efficient and effective do everyday business, solve problems, and manage the organization. The AP of an IS thus helps individuals to be informed about the respective organization. The use an individual makes of the AP of an IS in the widest sense can be characterized as asking questions about the organization and getting answers generated by the AP.

It is thus reasonable to understand the artefact as an intermediator between a customer and an expert. Perceived in this way, information systems are communication systems (see [CS98]). They help exploiting knowledge or abilities of experts. This exploitation, to be optimal with respect to economics, requires several users concurrently benefiting from the experts knowledge or abilities. Therefore the respective knowledge or ability has to be somehow drawn away from the expert. It must be presented in a form in which it easily allows users concurrently and sharing it all the time. Often this can be achieved by modelling a particular universe of discourse (UoD). This model both conserves and helps to operationalize the expert's knowledge and abilities. (In general most likely only a rather small ratio of these we guess.)

We use the information space metaphor (derived from [M*85] et al.) to support reasoning about IS. According to this metaphor an IS creates an information space. Customers are invited to interact with the space. It is a set of locations to which information objects are attached. Several of these locations are connected by links. Customer interaction with the information space consists of entering or leaving it, positioning at a location, interacting with an information object, or traversing inter-location links.

Information objects can be understood as units comprising data as well as operations that can operate on these data. The operations allow to access, filter, project, order, re-shape, create, delete, and import as well as export data and to apply particular business functions on these. For computer supported IS we argue that according to the *von Neumann* principle of computer architecture the distinction between data and operations is not an ontological, but a pragmatic one. Consequently data may be perceived as operation and vice versa. Data as well as operations may be of a primary concern to customers because these need to know the data or process them. Data and operations

[2] This in supposed to be the technology around the linguistic expressions stored within the IS.

additionally might be of a secondary concern for users in that they respectively index data or operations. As a generalizing notion for data, operations, and terms built of operations and data we use the term linguistic expression. More generally, linguistic expressions are considered to be composite signs. They thus are introduced by and underlie convention. See [Gr01] for more detail on signs. Composite signs are generated out of atomic signs according to a set of composition rules, i.e. a grammar. Signs are usually understood such that using a sign is inherently combined with a reference to something apart from them. Properly built and introduced linguistic expressions thus can be used to refer to something. Below we discuss and classify ways of referring to something with linguistic expressions maintained by AP.

The information space metaphor implies a usage model for IS in which customers interaction with the respective AP consists in shipping, processing and receiving linguistic expressions. As is well known (IPO model) AP processes a linguistic expression L shipped to it by a customer in three steps listed below in the order of their execution:

- Transform L into a processable form L`, and
- Obtain a result R matching L` in an optimum way, and
- Transform R in a user-friendly form and ship it to the user.

Achieving the optimality of the result R as well as the form in which R has to be transformed prior to shipping it to the customer implies that AP must have incorporated a representation of a customer model. It may be the case that this model neither is explicitly specified nor differentiates customers from each other.

Looking at IS as systems mediating a user-expert communication from an economical point of view suggests the metaphor of resource management system to characterize an aspect of IS. Resources other than linguistic expressions can be classified according to their capacity to:

- store,
- create, delete, and modify, or
- transport linguistic expressions.

However, concerning these resources IS users in general only have the choice to use them or not, but not to modify them. Rather than all of them in the sequel we only focus on the linguistic expressions. We are going to introduce classes of linguistic expressions that are of particular importance for using IS.

3. A conceptual model of ontologies

Originally ontology was a branch of philosophy dealing with and stating that what exists. For more information about philosophical ontology see, e.g. [RM98]. We believe that humans never can be sure about that what exists independently and outside their mind.

Knowledge about that what exists – as we believe – can only be acquired by experience, which involves perception. Perception in turn is not 'objective'. Evidence for this, e.g. is given by the discussion of perception in [Ma02] as process of conceptualising sense irritation that involves previously acquired knowledge as well as the discussion of perceptual categorization in [ET01] as a perception carving up the signals recognized from sense irritation into categories that are useful for us and thus depend on what we are. Humans thus can generate ideas about the world but never can know whether these are true. Consequently, in a restricted sense, we understand that ontology is about theories on conceptually constituting world parts. Concerning the reception of ontologies for information systems see, e.g. [Zu01].

According to [Gu98] ontologies are classified as either being:

- *Top-level ontologies*, describing general concepts like space, time, matter, object, event, action, etc. that are independent of a particular domain, or
- *Domain ontologies* or *task ontologies*, describing the terminology related to a generic domain (e.g. medicine, automobiles) or a generic task or activity (diagnosing, selling), or
- *Application ontologies*, describing concepts depending both on a particular domain and task, which are often specializations of domain- or task ontologies.

An example for a top-level ontology is given by [So00, p. 72]. Domain or task ontologies as well as application ontologies are what information systems development methods usually recommend (depending on the actual development task) the developers to work out. These then usually are used in the role of the expert model mentioned above. As is well known in information systems development models and the method or procedure of obtaining them in general differ from each other. Ontology papers (see, e.g. [Gu98] and [Zu01]) have discussed this distinction using the terms conceptualisation and specification (of it). From our point of view the relationship between ontologies and models is not an accidental one. We believe that ontologies are particular models, a point of view shared by Guarino, [Gu98].

Our favourite theory of modelling is due to *Stachowiak*. We use [St92], [St83], and [St73] as definitive sources concerning his theory. According to *Stachowiak* a model *S* (substitute) always is related (by the modeller *I*) to something apart from it, its original *T* (thing). We call this relationship *M(T,S,I)* model relationship. In line with our point of view indicated above concerning the limitations of human thinking, knowledge, and communication we presuppose *S* and *T* to be sets of linguistic expressions. We thus do not suppose that human thinking directly refers to real world things but that only our practical, physical activity interacts with them. *M(T,S,I)* is supposed to have a truncation property and a pragmatic property. The truncation property implies that the model in general lacks some of the expressions belonging to the original. (This gives rise to abstraction.) Connected with this property is what we call the plenty property implying that in general models contain expressions not contained in the original. The use one can make from the model in general depends on both of these properties. The pragmatic property implies that the model is valid for modelling only with respect to particular modelling individuals, purpose, applied techniques and tools, and period of time etc.

Presupposing an individual *I* and a modelling relationship *M(T,S,I)* modelling can be understood as producing, maintaining and using models. More specifically according to *Stachowiak* modelling implies temporarily and for a particular purpose and occasion substituting the thing T by the substitute S. The following multi-step procedure (that might be iterated) makes up modelling:

- Identifying a problem P_T in terms of *T* that shall be solved.
- Transforming P_T into a problem P_S in terms of *S*.
- Obtaining a solution R_S of P_S.
- Transforming R_S into a solution R_T of P_T in terms of *T*.

The mentioned transformation steps according to *Stachowiak* are supposed to be carried out based on functions $F:T\rightarrow S$, and $F^*:S\rightarrow T$. In case the solution R_T obtained in the end helps to solve the problem P_T one tends to believe that the modelling process was successful and that finding the particular solution to a large extent was consequence of performing the modelling steps mentioned above. In case the solution obtained is not satisfactory then one tends to think that either better modelling should take place, and then the multi-step procedure is applied again, or that modelling is not helpful in the particular case of problem solving.

In information systems development models occur that differ from each other in the way original and model are referred to each other by means of the model relationship. The respective modes of reference in IS development are known as **descriptive mode** and **prescriptive mode**. The difference between them can be paraphrased as the difference between the utterances 'it is so!' and 'so shall it be!'. In contrast to this ontologies might be ascribed a **constitutive mode** of reference. The original being substituted by its original characterizes this reference mode. *Wieringa* in [Wi90] used the concept of ‘direction of fit’ to distinguish from each other descriptive and prescriptive mode of reference. According to *Wieringa*, in case the relationship between model and original is considered not to be satisfactory respectively the model and the original have to be changed to yield a satisfactory relationship. Since the model replaces the original in case of constitutive mode of reference ‘direction of fit’ does not apply to it showing that this mode of reference is different than the other two modes.

It is stated in [Zu01] that ontologies need to be formal. We do not share this point of view. For us the formality of the language in which the model is specified gives rise to a quality aspect of the model. We believe that the quality concept employed in assessing the quality of a thing (compare for this the ISO definition of quality in [ISOQ]) should depend on the thing’s purpose. There can be identified some purposes for which formalization is a key factor such as agent communication on the Web as anticipated by [B*01]. For other purposes of ontologies, however, formalization might not be beneficial. When, e.g. we want to discuss with the customer the ontology worked out in course of an information system development project then due to his or her inability to deal with formalized languages formal formulation of the ontology might not be beneficial. Concerning this point we agree with the discussion in [Gu98], according to which an ontology might be specified in a variety of languages some of which might be formal while others might be not and even might be natural languages[3].

For an explication of the term formal language we refer to the table in `Table 1` The classification of languages in it according to [Or97] is due to *Gethmann*. In this table a '+' and a '-' respectively indicate independency or dependency of speaker or topic. Consequently, according to this classification a formal language is a language that is independent of the speaker using it and the topic the speaker is talking about. In that sense formal languages are shared generic languages.

		Speaker	
		+	-
Topic	+	Private language	Technical language
	-	Natural language	Formal language

Table 1: Languages classified after Gethmann

4. The kind of help asked for

Since the domains we are going to discuss below need to be worked out, i.e., instantiated in a concrete information systems development project it would be nice to have a generic domain specification and a list of instantiation procedures at hand together with a rule system governing the use of these procedures. These rules should depend on the system requirements and the business area affected. The specifications and the instantiation procedures could be grouped according to business areas if area specific specifications or procedures could be given. It furthermore would be convenient to have rules available that help to maintain the (conceptual) domain in case of changes. As is well known, in enterprises such changes occur frequently due to mergers, changed legislation or market conditions or similar. Rules on how to adapt a domain specification while its quality level is guaranteed would be nice to have. First attempts to this can be seen in the design primitive approach (compare, e.g. [B*92], [BP98] and [Th00]). Clearly, also facilities to automatically derive parts of system implementations would be helpful.

The artefact part AP of an information system IS shall give satisfactory answers to specified inquiries. These answers need to be given fast and AP often is required to be scalable. Important facets of scalability are:

- Significantly increased numbers of customers interacting with the AP shall only tolerably increase response time to individual inquiries.
- Significantly increased amount of data managed by the AP shall only tolerably increase response time to individual inquiries.

[3] Employing non-formal languages might make it hard to figure out what the model actually is.

- Both of the above with respect to throughput instead of response time.

Can work on ontologies help in incorporating properties like scalability into the representation of expert models, i.e., domain models? Can they help in targeting the required response time or throughput? The last question appears to be of particular interest because of well known methods to impact these quality aspects of information systems rely on introduction of redundancy in the expert model and considering the trade off between additional maintenance effort and cost, and reduced response time and increased throughput. For a discussion of such issues see, e.g. [B*92]. (For a definition of redundancy in the context of communication theory refer, e.g. to [CS98].) Axiom systems may contain redundancy, see, e.g. [E*92]. However, Mathematics and Logic usually don't deal with redundancy in the sense of how to incorporate it in into a particular model, what purpose it could be used for, and how to deal with the mentioned trade offs.

Patterns since about one decade are considered to be a valuable tool of analysis (, see, e.g. [Fo97]) and design (, see e.g. [B*96]). What concepts of pattern could be defined and usefully applied to ontologies? What pattern taxonomy could be formulated for ontologies in general? How could these patterns be employed for the development of information systems?

5. Domains ontology support is asked for

We identify a number of domains that are important for IS development or use. Referring to *Guarino's* taxonomy of ontologies above they are domain or task ontologies. Results about or experiences with such ontologies could help dealing with these domains. In the sequel we briefly discuss the domains that according to our knowledge play a role in IS development. The point here is not to give latest or deepest knowledge about the respective domain but just to outline it such that readers aware of IS issues can understand it. We hope that ontology aware or specialized readers will reply to it and point out results, methods, and references that in the sense discussed above can help in dealing with our domains.

5.1 Domain 1: UoDs

The IS artefact takes linguistic expressions from customers and ships such expressions to them. Consequently a language being the base of the expression exchange is needed that is understood by the users. This language may be a technical language. Its vocabulary as well as its syntactical rules up to some degree can be derived from the model of the universe of discourse the information system is about. It thus for IS development would be very convenient to have some kind of tool set for suitable UoD constitution. The method of *Abbot* still is in use: Experts somehow are made contributing to the creation of texts specifying the required UoDs and then the text is read against a semantic model. For full detail of the method refer to [Ab83]. Here we understand a **semantic model** as a set of modelling notions and abstraction concepts. See [Ka01] for more detail on these.

Reading a text against a semantic model means firstly, classifying text parts, in particular words, as instances of modelling notions or abstraction concepts, and secondly applying the definitions of these modelling notions and abstraction concepts as prescriptive model of the text part dealt with. This sometimes allows to identify text weaknesses and to better understand or enhance it by answers to questions asked to the experts. Popular semantic models for this purpose are the ER-model ([Ch76]), the relational model ([C70], [C79]), semantic nets such as conceptual graphs (, see, e.g. [So00]), and object models (, see, e.g. [ST93]).

5.2 Domain 2: Resources

The linguistic expressions managed by IS above were said to be the resource of the very IS. Identification of resources affects IS development. Some resources even might give IS completely new characteristics as is the case with inter space transitions, i.e. transitions taking a customer from a location in one space to a location in another space (the links in the Web). Dealt with them properly introduces Web IS as opposed to IS. Furthermore system development shall support that customer behaviour that fits to the system purpose and shall make it difficult for customers to use the system in a way contradicting the system's purpose. We have identified three dimensions of resource classification. The respective dimensions are:

- **Focus**, i.e., the customer refers to the UoD in its totality or only to a part of it, giving rise to the values 'global' and 'local' of the focus.
- **Modus**, i.e., the customer refers to the UoD with respect to a particular state of affairs or with respect to transitions of such states. This gives rise to the values 'static' and 'dynamic' of the modus.
- **Kind**, i.e., the customer refers to something because of his interest in it or because of its relation to something else. This gives raise to the values 'self contained' and 'referential'.

Identifying these dimensions of referring to something and their scale values leads to resources as classified according to `Table 2`.

Resource	Focus	Modus	Kind
Data	Global	Static	Self contained
Schema	Global	Static	Referential
Operation	Global	Dynamic	Self contained
Inter space transition	Global	Dynamic	Referential
Data subset	Local	Static	Self contained
View	Local	Static	Referential
Dialogue	Local	Dynamic	Self contained
Intra space transition	Local	Dynamic	Referential

Table 2: Resources of information systems classified

The idea to classify IS resources like this was derived from *Thalheim's* co-design approach, see [Th00], and extended by the 'kind' dimension. However, *Thalheim* appears not to mention the idea of resource. In his abstraction layer model he mentions 'data', 'view', 'operation', and 'dialogue' as aspects of IS that have to be observed throughout IS development. In our terminology he proposes to have in mind all the time throughout IS development all the resources and design the IS such that an optimum resource management can take place.

5.3 Domain 3: Information space use

Customers can enter and leave the information space; they can import and export data, as well as relocate in space and invoke offered functionality on available data. The following areas of customer activity need developers care to enable the IS aiding customers in these areas:

- **Identifying**, i.e., identifying data and operations being required to perform the actual task as well as the locations in space at which they are accessible to the customer. This might involve using a help functionality, or using personal data or functionality introduced into the IS as additional usage aid (personalization).
- **Locating**, i.e., determining the locations at which identified data and operations are situated.
- **Navigating**, i.e., proceeding from a given location to a target location. This might include using space exploration means as well as asking for proceeding related suggestions or simply following a more or less likely path through the space to explore it and get familiar with it and the way to use it.
- **Processing**, i.e., applying operations to data both of which are offered by the AP of the IS.
- **Handling**, i.e., using the IS efficiently to meet the respective goals. This might include using the help functionality or documentation or similar.

5.4 Domain 4: Customers

The IS artefact AP needs to have incorporated a representation of a customer model. It thus is reasonable to care for what customers actually exist. **Customer profiling** is a technique used to support the task of personalization of the IS interface or the whole application. Customer profiling consists in identifying customer types and allocating to each type a profile, i.e. a tuple of dimensions. To each dimension there is then associated a scale, i.e., a totally ordered set. [S*02] has proposed to construct customer types such that in the space of all user dimensions equipped with the scales they give rise to a convex region. Customer types can be identified by means of the method of *Abbot* applied to a text describing the customers of the system as anticipated by the system inventors or by the marketing, sales or human resources people of the company running the IS. These here would function as experts of the domain of system users.

5.5 Domain 5: Human inference making

We like to conceptualise human inference making by presupposing that it is essentially determined by main questions as follows:

1. **What exists?** I.e., what makes up the actual problem and its context?
2. **What should I do?** I.e., what is a solution to the problem at hand? In particular, what is a good solution to the problem? Connected to this, once a solution has been chosen, which solution plan should be implemented? The latter question then asks for the individual solution steps and their relative order.
3. **What can we know?** I.e., which transitions from already obtained statements about to new ones can be justified?

In an attempt to use traditional philosophical terminology we refer to the theories respectively giving answers to the questions listed above as: **ontology**, **ethics**, and **epistemology.** (Concerning these latter notions, see also [RM98].) The interesting point now is that one is free to choose a triplet (O,T,E), i.e. a particular ontology O, ethics T, and epistemology E. Since one is free in this choice one can try to make a reasonable choice. It might be the case that with respect to a purpose P_1 the triplet (O_1,T_1,E_1) is the first choice. However, with respect to a purpose P_2 this triplet might not be preferable.

Though we believe that a choice can be made there are dependencies to be observed. If, e.g. the ontology states that security risks exist in the domain the epistemology, however, does not permit obtaining knowledge about them then these two components of the triplet mentioned above may be not well chosen. If then furthermore the ethics recommends removing security risks then a real problem exists with the particular choice of ontology, ethics and epistemology.

6. Resume

This paper started with the presentation of a simple conceptual model of IS. We then proceeded with our view on ontology. Then we cited a taxonomy of ontologies from the literature. We explained our preferred theory of model and modelling and discussed ontologies as particular models. We elaborated on the kind of help we would appreciate be given from the ontology community and finally discussed five domains: resources, information space use, customers and human inference making. As we believe the IS community wishes to benefit from the ontology community concerning these domains. Apart from the last of these it was roughly explained how one in the area of IS tries to obtain the knowledge required with respect to the domains.

Bibliography

[Ab83] Abbot, R.: Program Design by Informal English Descriptions. Communications of the ACM, 26,11, 1983; pp. 882-894.

[BP98] Blaha, M.; Premerlani, W.: Object-oriented modeling and design for database applications; Prentice Hall; Upper Saddle River, New Jersey; 1998.

[B*01] Berners-Lee, T.; Hendler, J.; Lassila, O.: The semantic Web. Scientific American, May 2001.

[B*92] Batini, C.; Ceri, St.; Navathe, S.: Conceptual database design: an entity-relationship approach. The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1992.

[B*96] Buschmann, F.; Meunier, R.; Rohner, H.; Sommerlad, P.; Stal, M.: Pattern – oriented software architecture: a system of patterns. John Wiley & Sons, Chichester, England, 1999 4th. reprint, 1996.

[Ch76] Chen, P. P.: The Entity-Relationship model: toward a unified view of data. ACM Transactions on Database Systems, 1(1), 1976; pp. 9-37.

[Co70] Codd, E. F.: A relational model for large shared data banks. Communications of the ACM, 13(6), 1970; pp. 377-387.

[Co79] Codd, E. F.: Extending the database relational model to capture more meaning. ACM Transactions on Database Systems, 4(4), 1979; pp. 397-434.

[CS98] Cragan, J. F.; Shields, D. C.: Understanding communication theory: the communicative forces of human action. Allyn & Bacon, Needham Heights, Massachusetts, 1998.

[ET01] Edelmam, G.: Tononi, G.: Consciousness: how matter becomes imagination. Penguin Books, London, 2001.

[E*92] Ebbinghaus, H.-D.; Flum, J.; Thomas, W.: Einführung in die mathematische Logik, 3rd. ed., BI Wissenschaftsverlag, Mannheim et al., 1992.

[Gu98] Guarino, N.: Formal ontology and information systems: Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998.

[G*01] O'Grady, W.; Archibald, J.; Aronoff, M.; Rees-Miller, J.: Contemporary linguistics. Bedford/St. Martin's, Boston & New York, 2001.

[H*95] Hirschheim, R.; Klein, H. K.; Lyytinen, K.: Information systems development and data modelling: conceptual and philosophical foundations, Cambridge University Press, Cambridge, 1995.

[ISOQ] International Standardization Organization. Information technology – Software product evaluation – Quality characteristics and guidelines for their use. International Standard 9126, 1991.

[Ka01] Kaschek, R.: Modellbestandteile: Modellierungsbegriffe und Abstraktions-konzepte. In: Engels, G.; Oberweis, A.; Zündorf, A. (Eds.): Modellierung 2001, CI-Edition LNCS P1, Bonn, 2001, pp. 138 – 147.

[Ma02] Matlin, M.: Cognition, 5th. ed. John Wiley & Sons New York et al., 2002.

[M*85] Mayr, H. C.; Lockemann, P. C.; Bever, M.: A Framework for Application Systems Engineering. Information Systems, 10(1), 1985; pp. 97-111.

[Or97] Ortner, E.: Methodenneutraler Fachentwurf. B. G. Teubner Verlagsgesellschaft, Leipzig, 1997.

[Po01] Poser, H.: Wissenschaftstheorie: Eine philopsophische Einführung. Philipp Reclam jun. GmbH & Co., Stuttgart, 2001.

[RM98] Regenbogen, A.; Meyer, U.: Wörterbuch der philosophischen Begriffe. Felix Meiner Verlag. Hamburg. 1998.

[So00] Sowa, J.: Knowledge representation: logical, philosophical, and computational foundations. Brooks/Cole, Pacific Grove, California, 2000.

[St73] Stachowiak, H.: Allgemeine Modelltheorie, Springer Verlag, Wien, New York, 1973.

[St83] Stachowiak, H.: Erkenntnisstufen zum Systematischen Neopragmatismus und zur Allgemeinen Modelltheorie. In: Stachowiak, H. (Ed.): Modelle – Konstruktionen der Wirklichkeit, Wilhelm Fink Verlag, München, 1983, pp. 87 – 146.

[St92] Stachowiak, H.: Modell. In Seiffert, H.; Radnitzky, G., (Ed.), Handlexikon Zur Wissenschaftstheorie, pp. 219-222. Deutscher Taschebuch Verlag GmbH & Co. KG, München, 1992.

[ST93] Schewe, K.-D.; Thalheim, B.: Fundamental concepts of object oriented databases. Acta Cybernetica, vol. 11, no. 1 – 2, pp. 49 – 84.

[S*02] Schewe, K.-D.; Kaschek, R.; Wallace, C.; Matthews, C.: Modelling Web-based banking systems: Story boarding and user profiling. In: Mayr, H. C.; Van den Heuvel, W.-J. (Eds.): Proceedings of the eCoMo workshop 2002, Springer Verlag, 2002.

[Th00] Thalheim, B.: Entity-Relationship modelling. Springer-Verlag, Berlin, Heidelberg, 2000.

[Wi90] Wieringa, R.: Algebraic Foundations for dynamic conceptual models. Ph.D. dissertation, Free University of Amsterdam, 1990.

[Zu01] Zúñiga, G.L.: Ontology: its transformation from philosophy to information systems: Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001.