Fonseca, F. and J. Martin (2007) "Learning the Differences Between Ontologies and Conceptual Schemas Through Ontology-Driven Information Systems," ***JAIS - Journal of the Association for Information Systems - Special Issue on Ontologies in the Context of IS*** **Volume 8, Issue 2, Article 3, pp. 129–142, February 2007 – preprint version**

LEARNING THE DIFFERENCES BETWEEN ONTOLOGIES AND CONCEPTUAL SCHEMAS THROUGH ONTOLOGY-DRIVEN INFORMATION SYSTEMS

Frederico Fonseca
College of Information Sciences and Technology
The Pennsylvania State University
E-mail: fredfonseca@ist.psu.edu

James Martin, Emeritus
Psychology Department
The Pennsylvania State University
E-mail: jmartin501@gmail.com

ABSTRACT

In the traditional systems modeling approach, the modeler is required to capture a user's view of some domain in a formal conceptual schema. The designer's conceptualization may, or may not match with the user's conceptualization. One of the reasons for these conflicts is the lack of an initial agreement among users and modelers concerning the concepts belonging to the domain. Such an agreement could be facilitated by means of an ontology. If the ontology is previously constructed and formalized so that it can be shared by the modeler and the user in the development process, such conflicts would be less likely to happen. Following up on that, a number of investigators have suggested that those working on information systems should make use of commonly held, formally defined ontologies that would constrain and direct the design, development, and use of information systems – thus avoiding the above mentioned difficulties. Whether ontologies represent a significant advance from the more traditional conceptual schemas has been challenged by some researchers. We review and summarize some major themes of this complex discussion. While recognizing the commonalities and historical continuities between conceptual schemas and ontologies, we think that there is an important emerging distinction which should not be obscured, but should guide future developments. In particular, we propose that the notions of conceptual schemas and ontologies be distinguished so as to play essentially different roles for the developers and users of information systems. We first suggest that ontologies and conceptual schemas belong to two different epistemic levels. They have different objects and are created with different objectives. Our proposal is that ontologies should deal with general assumptions concerning the explanatory invariants of a domain – those that provide a framework enabling understanding and explanation of data across all domains inviting explanation and understanding. Conceptual schemas, on the other hand, should address the relation between such general explanatory categories and the facts that exemplify them in a particular domain (e.g., the contents of the database). In contrast to ontologies, conceptual schemas would involve specification of the meaning of the explanatory categories for a particular domain as well as the consequent dimensions of possible variation among the relevant data of a given domain. Accordingly, the conceptual schema makes possible both the intelligibility and the measurement of those facts of a particular domain. The proposed distinction between ontologies and conceptual schemas makes possible a natural decomposition of information systems in terms of two necessary but complementary epistemic functions: identification of an invariant background and measurement of the object along dimensions of possible variation. Recognition of the suggested distinction represents, we think, a natural evolution in the field of modeling, and significant principled guidance for developers and users of information systems.

Keywords: ontologies, conceptual schemas, modeling

I. INTRODUCTION

In the conventional conceptual modeling activity, ontologies are either bypassed or lack a formal specification [Frank, 1997]. Guarino [1998] considers that every IS has an ontology which many times is not explicit but obscurely embedded in parts of the system. Hirschheim *et al.* [1995] make a similar point for IS saying that all data modeling "presumes either implicitly or explicitly some form of modeling of data meaning". And beyond that, the designer's conceptualization may, or may not match with the user's conceptualization. One of the reasons for these conflicts is the lack of an initial agreement between users and modelers concerning the concepts belonging to the domain. Such an agreement could be facilitated by means of an ontology. If the ontology is previously constructed and formalized so that it can be shared by the modeler and the user in the development process, such conflicts would be less likely to happen.

Guarino [1998] coined the term *ontology-driven information systems* for systems that make use of formally defined ontologies. According to him an explicit ontology plays a central role in this kind of system thus driving all of its aspects and components. Guarino also discussed the prevalent use of the term *ontology* in Artificial Intelligence. He says that "*an ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words. This set of assumptions has usually the form of a first-order logical theory, where vocabulary words appear as unary or binary predicate names, respectively called concepts and relations. In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation.*" [1998 p.4]

Nevertheless some authors consider that the idea of using ontologies as the foundation for IS development is an old idea. Sowa [2000] points out that the problem has been studied since the 1970s. Sowa uses the work of Tsichritzis and Klug [1978] for the American National Standards Institute (ANSI) to demonstrate his point. Tsichritzis and Klug proposed the use of conceptual schemas as a way to represent and store knowledge about an application domain. According to Sowa, in Tsichritzis and Klug's approach a conceptual schema would also be central to the development of information systems in a fashion similar to what is being proposed for ontology-driven IS.

We argue that there is a natural distinction between ontologies and conceptual schemas which arises from the differing roles those models play in the epistemic functioning of the developers and users of the information systems incorporating them. By explicating an epistemological basis for the terminological distinction and proposing that usage follow the epistemic distinction, we hope to clarify and contribute to the field of information systems analysis and design (ISAD).

Using an example from Hirschheim et al. [1995], we can see that a conceptual schema would have propositions such as "*a car is of a particular model*" and "*a car is owned by a person*". Note that such propositions invite specification (e.g., which model or person) and thus direct measurement. Currently, we might find similar propositions in an ontology too. That is one of the reasons why they are often treated as equivalent in the literature. But we want to show how they should be different and thus reflective of differing epistemic motives. In this example, we would expect an ontology not to specify the dimensions of measurement. Instead, an ontology would provide a basis for causally based understanding and explanation and, thus, inference to causally based predictions. For instance, you should expect that people buy their cars according to some rules such as economic constraints, environmental concerns, or purposes of use. By providing a general framework for causal inference, an ontology should enable the conceptual schema to express the rules and expectations unique to this domain. As a consequence, one might expect the conceptual schema to enable an inference such as *the kind of person buying this kind of car has access to its purchase price*. In this way the conceptual schema would specify relevant dimensions of measurement (e.g., cost of purchase and available financial resources). Thus the database could have a relevant fact such as *John Doe, 50, owns a 2005 Cadillac XLR which costs over $75,000*. From this it could be inferred that John has access to over $75,000. In general, John's patterns of economic behavior allow us to measure something about his wealth. The conceptual schema thus tells us what to measure and therefore what the relevant dimensions of the database must be. In this way, operating in conjunction with the ontology, the conceptual schema allows a principled distinction between intelligible (i.e., theory relevant) information, and an almost unlimited mass of irrelevant facts.

It needs to be pointed out that it is precisely the capacity of the information system to integrate information in a causally relevant way that would enable the understanding of what we call the *functional significance* of information. In our example, assuming the intention of purchasing a car, the functional significance for John of over $75,000 in his bank account might be that he can purchase a new Cadillac. But that significance can best become apparent in a context where causal integration of a domain of information is inherent in its organization. Accordingly, if one were interested in IS that enable practice, then the sort of causally relevant integration we have indicated above would be crucial.

In this paper we will argue that ontologies and conceptual schemas should be conceived to operate at two distinct epistemic levels. Furthermore, they have different objects and are created with different objectives. Nevertheless it is important to note that there is an interaction between the two and that this interaction may be benefic to the development of both as we are going to show in section V. The relative independence between the measurement function of conceptual schemas, on the one hand, and the presumptive and explanatory functions of ontologies, on the other, enables a potential clash between these two components of an IS. As the above example shows, it is possible to record data in an information system which is inconsistent with the ontology although consistent with the conceptual schema – indicating the possible need to change the ontology. On the other hand, an inconsistency between the ontology and the IS might indicate that the IS should be checked for errors, or even that the conceptual schema should be revised. A degree of independence between ontologies and conceptual schemas creates the potential for the system to record the need for its revision. In this case, the system will promote the possibility of learning by the user and the developer – certainly a benefit of accepting the distinction we are proposing.

II. DIFFERENT SCOPES FOR ONTOLOGIES AND CONCEPTUAL SCHEMAS

There is a continuum from the developer/user conceptualizations, later expressed in informal and formal languages, to the creation of conceptual schemas and the subsequent representation of facts in a database. However, in order to achieve our objectives with IS, we need to separate things in clearly defined levels. We propose that computational ontologies and conceptual schemas definitely should belong to two different epistemic levels. The search for data independence and the creation of more generic and abstract models which started with two-level architecture continued with the three-level architecture in the ANSI/X3/SPARC proposal [Hirschheim et al., 1995]. Techniques and tools for information systems design and analysis are always in constant evolution. Automatic generation of conceptual schemas from ontologies [Fonseca et al., 2003, Shanks et al., 2003, Sugumaran and Storey, 2002, Wand and Weber, 2002] and the creation of conceptual schemas that are more independent from the implementations [Caruso and Umar, 2004, Parsons and Wand, 2000, Scheer, 2003, Zdonik and Maier, 1990] are examples of cases in which the borders of these two epistemic levels sometimes are blurred.

At the same time that the research in conceptual modeling argues for the creation of more generic models, the research on ontologies sometimes (wrongly in our point of view) go to specifics such as having instances of classes within ontologies. As McGuinness [2003] points out, “some classification schemes only include class names while others include ground individual content”. Bodenreider et al. [2004] points to a similar problem. They say that in some medical ontologies names represent classes or universals while in other ontologies names convey how reality is perceived, measured, and understood. We argue that the recording of instances or ground content should be done by the IS itself under the guidance of the conceptual schema. In general, ontologies should not include instances of its concepts. The recording of facts belongs to the domain of measurement (i.e., the level of conceptual schemas).

Regarding the scope of ontologies and conceptual schemas, some researchers suggest that ontological research is the study of instrumentally useful formal models, not of the formal properties of reality. For instance, Smith [2003] argues against what he calls 'instrumental ontologies'. Such ontologies may be interpreted in terms of the 'closed world assumption'. In such cases, what Gruber [1995] asserts for AI, will be true of ontologies: "what 'exists' is that which can be represented" (p. 907). Milton distinguishes between ontologies that are theory-focused and those that are pragmatically-oriented [1998]. For Milton, theory-focused ontologies are the ones created using philosophical theory. Pragmatically-oriented ontologies are very common in the practice of information systems and are targeted to specific domains such as banking or taxation. Ontologies often combine a philosophical approach with pragmatic purposes. This odd combination may result in incompatible philosophical underpinnings being used in the same pragmatically-oriented ontology. The pragmatic approach allows the investigator to deftly avoid users' objection that a given ontology does not coincide with his or her view of reality. The developer of ontologies can reply that what is at issue is not whether the ontology is correct, but whether the models it defines are useful, or adequate, for some (limited) purposes. In this case, an ontology may have relevance to a narrow range of problems without being correct. We strongly disagree with this strategy [Fonseca and Martin, 2005]. Adjustment of an IS to local conditions is the role to be fulfilled by conceptual schemas. An excessively narrow view of ontologies will lead to the same problems in modeling that led to the very emergence of ontologies in ISAD research in the first place. These were the problems uncovered by Guarino [1998] in his introduction of Ontology-driven Information Systems.

From the discussion above we can begin to understand why it is implicit in Sowa's argument that a conceptual schema (the result of the modeling process) is essentially the same as a computational ontology (the result of the ontology engineering process). Other researchers have asked themselves whether ontologies were actually the well-known conceptual data modeling techniques in disguise [Winter, 2001]. But Guarino [1998] himself advised against using ontology as just a "fancy name denoting the result of familiar activities like conceptual analysis and domain modeling" (p. 3). The origin of the confusion lies not only in the similarity of the two concepts but also to the fact that research on the use of ontologies in information systems is a relatively recent effort. Green and Rosemann [2004] say that not only "the type of research work that is conducted under the umbrella term, 'ontologies', varies significantly" but the "understanding of an ontology in terms of its scope, details and purpose varies significantly" (p. i).

The distinction between the two terms that we propose here is needed because a misuse or misunderstanding of the terms will hinder progress both in the research and in the practice of ISAD. Ontologies actually reflect the result of the evolution of research in modeling. For instance, we can trace the origins of the idea of using ontologies for ISAD in the early work of Wand & Weber [1989, 1990]. They introduced the idea that a modeling method could be reality or information-system driven. In our argument, we make a similar comparison on the scope of ontologies (reality driven) and conceptual schemas (IS and measurement driven) to highlight the differences between them. Our position in this paper is to argue for a more restricted view of conceptual schemas in which they would be closer to what is called a logical data model. We propose that the broader versions of conceptual schemas should often be designated as ontologies. Next we follow on with a review of the ontologies and conceptual schemas in the literature.

III. ONTOLOGIES AND CONCEPTUAL MODELING IN THE LITERATURE

GUARINO'S CONCEPT OF ONTOLOGY-DRIVEN INFORMATION SYSTEMS

Guarino [1998] uses a broad definition of information systems to analyze the importance of ontologies for IS. He includes applications such as "enterprise integration, natural language translation, medicine, mechanical engineering, standardization of product knowledge, electronic commerce, geographic information systems, legal information systems, biological information systems" [Guarino, 1998 p.3]. Guarino and Giaretta [1995] favor the meaning of the term ontology as being a theory instead of being a simple specification of particular epistemic states. They say that "an ontological theory differs from an arbitrary logical theory (or knowledge base) by its semantics, since all its axioms must be true in every possible world of the underlying conceptualization" (p. 31).

Guarino sees two important categories in the use of ontologies. First, a methodological one, characterized by highly interdisciplinary approaches using philosophy and linguistics to build the ontologies. Second, an architectural category in which we are interested here. In this category, an ontology plays a central role in IS driving all of its aspects. Guarino proposes that those working on information systems should make use of commonly held, formally defined ontologies in the design, development, and use of information systems. He named such systems ontology-driven information systems (ODIS). Next we summarize the mains aspects of an ODIS as seen by Guarino [1998].

First, Guarino considers that "every (symbolic) information system (IS) has its own ontology, since it ascribes meaning to the symbols used according to a particular view of the world". The difference in an ODIS is that the ontology should be made explicit before the IS is even designed. Then, Guarino discusses a temporal and a structural dimension of the use of ontologies in IS. In the temporal dimension, an ontology can be used at the development time or at the run time. He says that "an important benefit of using an ontology at development time is that it enables the developer to practice a "*higher" level of reuse than is usually the case in software engineering (i.e. knowledge reuse instead of software reuse). Moreover, it enables the developer to reuse and share application domain knowledge using a common vocabulary across heterogeneous software platforms. It also enables the developer to concentrate on the structure on the domain and the task at hand and protects him from being bothered too much by implementation details*" (p.12). At run time an ontology may enable, for instance, the communication between software agents.

In the structural dimension Guarino considers that an ontology can help build more consistent user interfaces or support the creation of software components. In the conceptual modeling phase an ontology is used to link informal specifications to formal specifications. At run time, ontologies can be used to support information integration. A longer review of Guarino's ODIS concept can be found in Kishore *et al.* [2004].

But before we start our discussion on the differences between ontologies and conceptual schemas, there is some terminology that needs to be clarified. We need to explain how the term ontology is used in a philosophic sense. For philosophy, ontology is the "branch of metaphysics that concerns itself with what exists" [Blackburn, 1996 p.269]. Also, the fact that philosophers have been studying ontology since Aristotle may help us find an answer for how to build good ontologies. But philosophy itself has different branches that have different assumptions about the world and how we may understand it. Although the basic assumptions behind the theories that are used in creating ontologies are an important subject of study, it is not our main concern here. Other works in the literature discuss the subject at length. For instance, Wand & Weber [Wand and Weber, 2004] and Milton & Kazmierczak [Milton, 1998, Milton and Kazmierczak, 1999] look at which kind of philosophical work should become the foundation for the ontologies of IS. The study of how the term ontology is used in philosophy and how it is used in information and computer sciences is discussed in other work [Smith, 2003, Wyssusek, 2004, Zuniga, 2001]. The different philosophic positions behind the different modeling tools and techniques are extensively analyzed in Hirschheim et al. [1995].

Next we will clarify meanings of the terms computational ontology and conceptual schema that will be used in the remainder of the paper. We will also make a distinction between a conceptual model and a conceptual schema.

COMPUTATIONAL ONTOLOGIES

From Gruber's [1992] definition that "an ontology is an explicit specification of a conceptualization", Guarino [1998] created a refined distinction between an ontology and a conceptualization. Guarino starts the discussion saying that a conceptualization is "a set of conceptual relations defined on a domain space" and that it is important to "focus on the meaning of these relations, independently of a state of affairs" [Guarino, 1998 p.3]. He says that in a conceptualization we are interested, for instance, in the meaning of the relation 'above' instead of being concerned that in this particular state of affairs object A is above object B. After clarifying what a conceptualization is, he says that "*an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models".* Smith [2003] says that in the current context of research on information sharing, an ontology is seen as a dictionary of terms expressed in a canonical syntax. In this use it is implied that ontology is a common vocabulary shared by different information systems communities. Smith then gives a definition of an IS ontology: *"an ontology is a formal theory within which not only definitions but also a supporting framework of axioms is included (perhaps the axioms themselves provide implicit definitions of the terms involved)*" [Smith, 2003].

We are interested here in computational ontologies, ontologies that are used to build IS as Guarino [1998] suggested in his concept of ontology-driven IS. From this point on in the paper, we use the term *computational ontologies* as it is suggested in Kishore *et al.* [2004]. It has the same meaning as *IS ontologies* as used by Smith [2003] and by Fonseca & Martin [2005]. Guarino [1998] means the same thing when he uses the term *ontologies* in the plural.

CONCEPTUAL MODELING IN IS

Another topic we need to address is the difference between conceptual *model* and conceptual *schema*. As Hirschheim *et al.* [1995] mention, citing Schmid [1983], there is some confusion about the terms model and schema in the field. A model is "a set of conceptual and notational conventions which help to perceive, organize and specify some data" [Hirschheim et al., 1995 p.26]. Entity-Relationship (ER) [Chen, 1976], Object Modeling Technique (OMT) [Rumbaugh et al., 1991], and Unified Modeling Language (UML) [Rational Software Corporation, 1997] are examples of conceptual models. Conceptual schemas, on the other hand, refer to the result of the modeling, namely a set of diagrams that use a given conceptual model as a language to express the specific data structures for an application that is going to be developed. Wand & Weber [2002] take further the distinction between model and schema. They propose a framework set in what they call a *conceptual-modeling context*. In this context we use a *conceptual-modeling method* that applies a *conceptual-modeling grammar* to model real-world phenomena and create a *conceptual-modeling script*. In this paper we use the term conceptual schema to denote what Wand & Weber are calling conceptual-modeling scripts.

In figure 1 we make a parallel between the creation of computational ontologies and conceptual schemas. Observing a universe of discourse (UoD) we use methods to apply languages or grammars in order to create models of the UoD. We can observe that the processes of construction of both computational ontologies and conceptual schemas are similar. Further in the paper we will notice that one of the main differences should be on the scope of the UoD.

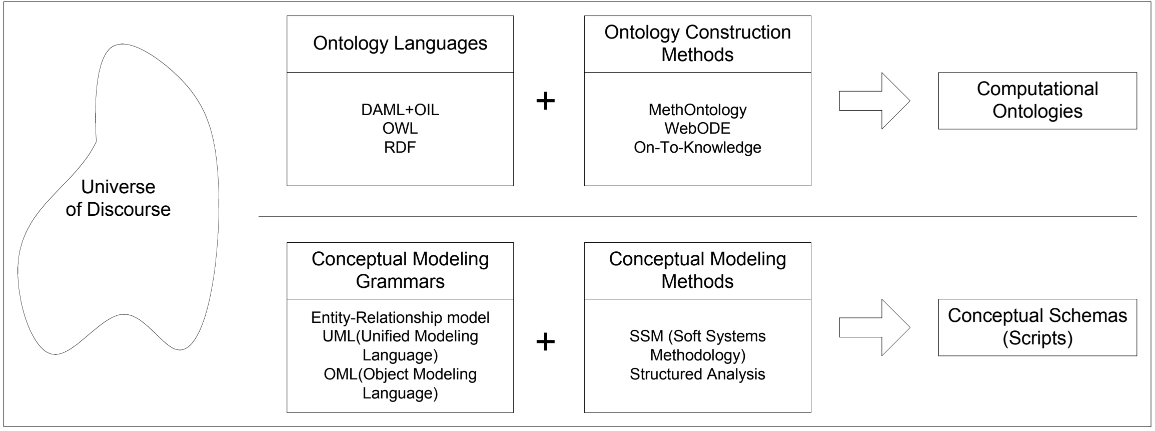
Universe of Discourse
Ontology Languages
DAML+OIL
OWL
RDF
+
Ontology Construction Methods
MethOntology
WebODE
On-To-Knowledge
Computational Ontologies
Conceptual Modeling Grammars
Entity-Relationship model
UML(Unified Modeling Language)
OML(Object Modeling Language)
+
Conceptual Modeling Methods
SSM (Soft Systems Methodology)
Structured Analysis
Conceptual Schemas (Scripts)

Figure 1 - A parallel between the creation of computational ontologies and conceptual schemas

Hirschheim *et al.* [1995] define data modeling as “a change process taken with respect to object systems consisting of data and its uses in a set of environments by a development group using a representational form to achieve some objectives” (p.27). They consider it a change process because if the model is accepted by the organization it triggers changes in the knowledge base of the organization. To complement this definition we need to know that object systems “consist of phenomena ‘perceived’ by members of the development group. What is ‘perceived’ is either given as a reality independent of the observer, or socially constructed through sense-making and institutionalised conventions” (p.27). Klein and Lyytinen [1992] consider that data modeling is “the activity of constructing a data specification by applying the generic abstraction concepts of a data model (language) to a particular application domain” (p.204).

We can see an information system as being constituted by an information processor and a Universe of Discourse description (UoDD). The Universe of Discourse (UoD) “is a selected portion of the world and it constitutes the universe made known to the IS and thus to the IS users by the IS” [Hirschheim et al., 1995 p.179]. The UoDD is a formal representation of the UoD. It has two parts, the conceptual schema and the information base. The conceptual schema contains sentences representing general facts, facts which hold for all entities in a UoD. These facts define which entities can exist and also some entities that must exist in the UoD [Hirschheim et al., 1995].

HOW GOOD CAN OUR MODELS BE?

There is a deeper question that, albeit not thoroughly discussed here, needs to be mentioned. The question is “what is our universe of discourse and how can we know about it? Wand and Weber [2004] point to the quality of our knowledge as a potential problem saying that some ontologies created by information systems scholars “are not always rooted in a sound foundation of more fundamental constructs like things and properties”. Smith [2003] speaks of good and bad ontologies and how Science can help us build the former. He says that “our best candidates for good conceptualizations will, however, remain those of the natural sciences – so that we are, in a sense, brought back to Quine, for whom the job of the ontologist coincides with the task of establishing the ontological commitments of scientists, and of scientists alone” (p.163). Milton’s [1998] main question is related to top-level ontologies, which may be “used to provide theoretical underpinning for representation and modeling in information systems” [Milton, 1998 p.85]. Top-level ontologies are created from a philosophical perspective and are suggested as a basis for domain ontologies. The idea is that if we follow the rigid methods of philosophy we will come up with good top-level ontologies. All other ontologies, including computational ontologies, should be derived from the top-level ontologies. Milton [1998] distinguishes between ontologies that are theory focused and those that are pragmatically oriented. For Milton, theory focused ontologies are the ones created using the philosophical theory. The other type of ontologies, according to Milton, are very common in the practice of IS and are targeted to specific domains such as banking or taxation. According to him, computational ontologies often use a mix of (philosophical) ontological tools and pragmatic purposes. This odd combination may result in incompatible philosophical underpinnings being used in the same pragmatically oriented ontology. Both Milton and Wand & Weber are looking at which kind of philosophical work should become the foundation for the ontologies of IS while the main focus of this paper is to distinguish between computational ontologies and conceptual schemas. Thus although recognizing the importance of their research question we leave it out of this work.

Computational ontologies and conceptual schemas have some similarities as well as some differences. Consider, for instance, an ontology of banking and the conceptual schema of Citibank's information system. Are those two the same thing? How similar and different are they? Which criteria can we use to investigate the differences and similarities between the two? In the next section, we propose a framework to understand better the differences and similarities between computational ontologies and conceptual schemas.

IV. SIMILARITIES AND DIFFERENCES BETWEEN ONTOLOGIES AND CONCEPTUAL SCHEMAS

We propose to study the differences and similarities between computational ontologies and conceptual schemas under the light of two concepts: objectives and object. First, we are interested in knowing the objectives of creating computational ontologies or conceptual schemas. Are we trying to explain a domain? Are we trying to integrate information? Are we trying to represent reality? Or in opposition to those, are we trying build a tool that will helps us to measure specific features of reality, following Quine [1953], as cited by Smith [2003], "*to be is to be the value of a bound variable*". Second, concerning the objects of computational ontologies and conceptual schemas, what are they designed to describe, to represent? Is it the real world, reality? Or is it a specific conceptualization of a specific domain as perceived by the users of the IS?

By focusing on objectives and object, we hope to contribute to a better understanding of the differences between ontologies and conceptual schemas. The literature that deals with the problem usually makes the comparison without a clear criterion and also without a clear terminology. By establishing the terminology and looking into the problem using our proposed framework of objectives and object (Figure 2) we will be able to make clear what the differences between computational ontologies and conceptual schemas are.

In this section we are going to argue that regarding the objectives:

Computational ontologies should focus on explanation and information integration grounded in assumptions about invariant conditions that define the domain of interest;

Conceptual schemas should focus on linking the general ontological categories with particular observations to be classified in IS. The objective of conceptual schemas should be thus to enable the measurement and classification of the observed facts.

And regarding the object:

Computational ontologies should focus on the real world, on the reality, instead of focusing on what can be represented. The object of computational ontologies should be the system representing the invariant conditions of the domain of interest – the general, and assumed, categories that are taken to define a domain;

Conceptual schemas should act as links, connecting the ontologies with the data, or "facts". Accordingly, the object of a conceptual schema should be the relationship between ontological categories and the permissible range of variation among the facts that must be brought into relation with those categories. Conceptual schemas have a two-sided aspect. On the one hand they are defined in relation to categories that may be contained in the ontologies (e.g., "automobile"), and on the other hand they provide machinery for linking the general class of automobiles with a particular auto owned by a particular person of a particular age. In this way, measurement – the appropriate linking of general categories with particulars – is made possible.

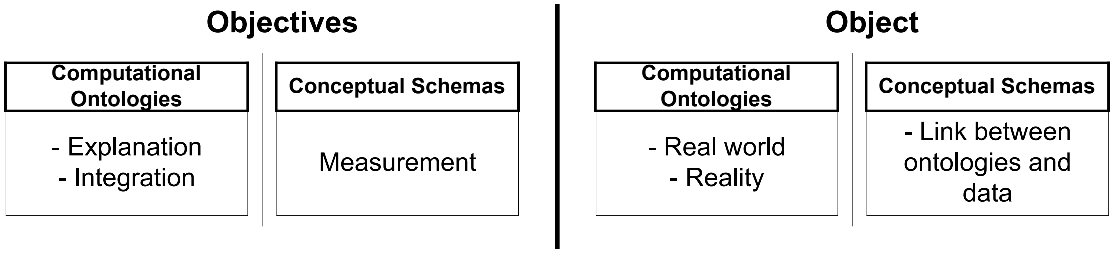

Figure 2 - Differences between the objectives and the object of computational ontologies and conceptual schemas

THE OBJECTIVES

What are the objectives which ontology engineers and conceptual modelers have in mind when they are building their artifacts? We think that when we look at this question we will be able to establish a difference between computational ontologies and conceptual schemas.

Wand & Weber [2002] remind us, drawing from Kung and Solvberg [1986], that the purposes of a conceptual schema are "(1) supporting communication between developers and users, (2) helping analysts understand a domain, (3) providing input for the design process, and (4) documenting the original requirements for future reference" (p. 363). It is important to notice here the mention to original requirements. The question is then how much the IS requirements drive the construction of a conceptual schema?

Hirschheim *et al.* [1995] point to a possible answer describing what is in a universe of discourse (UoD) from the point of view of what they call the fact-based school: "*the entities (the basic entities of the UoD) have several characteristics. First, they can be separately distinguished, they have independent existence, and they are selected based on how interesting they are for IS users. Thus, prominent criteria for something to be counted as an entity are: it has a unique identifier, its existence does not depend on the existence of other entities (its existence is caused by a specific event that creates the entity), and there is a need to know what facts are associated with the entity (i.e. what IS users want to know about the UoD)*" (p. 180). They also state that the target of modeling is data, i.e., "*all invariances that are stored and manipulated in the computer system and their potential meanings for different users of the IS*" (p.27).

Uschold and Gruninger [2004] also write of purposes as being the difference. They consider that computational ontologies have multiple purposes including interoperability, search, and software specification. They think that the most common use of a schema is to structure a set of instances for querying an IS. Their emphasis on software specification highlights one of the proposed aspects of ontology-driven information systems, the automatic generation of software components.

Computational ontologies enable explanation and information integration. They achieve this by linking the users of the IS with a common background of assumptions concerning the whole within which the facts recorded within the IS arise. Fundamental to these computational ontologies is a common perspective that makes communication via the IS a meaningful enterprise. Thus, the computational ontologies contain some of what is taken to be invariant for a domain. On the other hand, conceptual schemas provide a framework for measurement within the context of assumptions described in their corresponding computational ontologies. Thus, conceptual schemas define the dimensions in question, and the range of those questions. In contrast with the computational ontologies' specification of what is assumed to be invariant, conceptual schemas specify what is taken to vary and the range of possibilities in view.

We argue that the main objectives of creating computational ontologies and conceptual schemas are different. Conceptual schemas are built with a specific information system in mind. They have a practical purpose of defining, constraining, and limiting what is going to be registered and manipulated by the information system. This fact shapes the objectives of a conceptual schema. Ontologies are theories that explain a domain by revealing it as a coherent whole. They make predictions and they bring expectations. They give meaning to terms through the explicit relationships between concepts. Although ontologies may be used by an IS and also used to support the development of an IS they are not intrinsically tied to a specific IS as conceptual schemas are. It should also be clear that we are not claiming that ontologies are more or less than conceptual schemas. They are different in nature and should be built with different purposes in mind. This leads us to our next discussion on the difference between them: The object, what is being modeled.

THE OBJECT

Many of the works in the literature focus in a way or another on the object of computational ontologies and conceptual schemas. Usually it is accepted that the process of creating both of them is similar but most work points out to a variation in the scope. For instance, Kishore *et al.* [2004] consider that the process of creating an ontology is similar to the process of creating a conceptual schema. They point out that the main difference is that "*the domains of analysis can be very large and the knowledge capture and representation mechanisms can be quite varied*" (p. 168). Olivé [2004] also thinks that the process of conceptual modeling is similar to ontology engineering. He says that "*given that a CS (conceptual schema) is knowledge, conceptual modelling can be considered as a specialization of the field of Knowledge Representation to information systems*" (p. 20). Regarding the object, Olivé stresses that a model has to incorporate knowledge not only about the domain but, most important, also knowledge about the functions the IS has to perform. Olivé then proposes a division of the traditional conceptual schema in two. He says that one component of a schema, the *Domain Conceptual Schema* (DCS), defines the knowledge about the domain. Such facts are true of the domain and independent of the IS. The second component, the *Functionality Specification*, would define "the entity and relationship types of the Domain Conceptual Schema that are represented in the IS, the types of the query events the IS has to respond to, and the generating condition of the generated query events" (p. 21). Seen this way the problem of the difference becomes only a question of terminology. We argue that the problem is deeper than that. Computational ontologies and conceptual schemas are not only different, they need to be different. Both of them are needed in order to address problems in modeling. What Olivé is calling Domain Conceptual Schema should be called a computational ontology. What he calls Functionality Specification should be called a conceptual schema. Using Olivé's terminology (with which we do not agree) some of the differences reviewed here would disappear. Actually, Olivé explicitly states that "*in the information systems field, an ontology corresponds closely to a DCS. The main difference may be that DCSs include also the concept of domain event (and their effects), while most current ontologies are restricted to the structural part of schemas*" (p.24). In a similar fashion, the ANSI/SPARC architecture (the three level architecture) states that the *conceptual level*, which has as its output a conceptual schema, has an emphasis on the data or in the description of the database structure as viewed by the community of users. However, if we look at the *external level* of the ANSI/SPARC architecture and see the definition of an external view, we can find more similarities to an ontology than we find in the previous level. This is part of our argument. The emergence of ontologies in IS research is in fact the result of the development of the field. Computational ontologies are needed as well as conceptual schemas, but this need comes at different levels.

The issue of the differences between computational ontologies and conceptual schemas became a recent topic of interest in the SIGMOD Record. First, Cui *et al.* [2002] consider that there are many similarities between them but there is one main difference in what is being modeled. Spyns *et al.* [2002] contend that a conceptual schema focus on a single specific enterprise application while computational ontologies models a domain explicitly intended to be shared by multiple applications. Finally, Uschold and Gruninger [2004] give a definition of computational ontology that encompasses much more than what a conceptual schema addresses. They say that "an ontology is used by an agent, application, or other information resource, to declare what terms the agent uses, and what the terms mean. By making this information publicly available, it becomes possible for high-fidelity semantic communication to take place - agents can communicate and share meaning with other agents, and agents can understand the meaning of applications, databases and other information resources on the Web"

(p. 60). Their discussion of constraints hints at the object topic. They write that constraints in a conceptual schema exist to ensure the integrity of data, while in computational ontologies constraints would have a more encompassing object, the logical consistency of the theory which the ontology expresses.

It is our position here that computational ontologies have a broader scope than conceptual schemas. Ontologies aim to explain a world that, although limited, is not as limited as the specific world modeled by a conceptual schema. Ontology engineers are free to include extra information in their models, information which will help to establish the meaning of other concepts in the ontology. On the other hand, conceptual modelers are limited to describe concepts and terms that will be in the information system. For a matter of practicality as well as of feasibility, they have to constrain themselves to the main entities and relationships that will be ultimately recorded by the IS.

COMPUTATIONAL ONTOLOGIES AS AN EVOLUTION OF THE FIELD

We can also see that the implicit notion of the concept *ontology* was present early in the literature. One of our arguments here is that computational ontologies represent an evolution in the field of modeling. They are a needed concept that evolved through the work of the many researchers in the field. Even before Guarino [1998] argued for ontologies as the main component of an IS, Wand and Weber [1989] proposed the use of philosophical ontology in order to achieve a better understanding of what an IS is. Computational ontologies represent a step beyond conceptual schemas. The two are different in their objectives and in their object. But they are linked to each other as models with different scopes and objectives. Both are necessary for modeling and for the advance of the research in the field. We will see in the next section that ontologies and conceptual schemas are linked and are complementary to each other.

In summary, as sophistication with the modeling aspects of IS design has grown, the objectives and object of the older conceptual schemas have been decomposed into two sets of objectives and objects. This has motivated a distinction between two categories of IS modeling – computational ontologies and conceptual schemas. In this section, we have been concerned with the distinction between the respective objects of ontologies and conceptual schemas. We have argued that the object of computational ontologies is the interlocking set of invariant structures of a domain which enable explanation, prediction and communication to take place. On the other hand, the object of conceptual schemas has been held to be the conceptual relationships between general categories, such as those that populate ontologies, and the individual events that fall, or might fall, under those categories. It is the specification of these relationships that enables the measurement and classification of such events.

V. *MEASUREMENT* AS THE KEY TO DISENTANGLE THE MEANING OF COMPUTATIONAL ONTOLOGIES AND CONCEPTUAL SCHEMAS

In the following, we will show how ontologies and conceptual systems may interact with a database to make possible an intelligible source of information about a domain. This tripartite scheme – ontology, conceptual schema, database – follows the spirit, if not the letter of Kant's analysis of scientific understanding (i.e, The Critique of Pure Reason) [Kant and Smith, 2003]. In the first place, we propose that an ontology should provide a general framework for explanation. This framework is required if the IS is to do more than record correlations among events – if it is to provide a basis for intelligible integration of the facts, as well as for significant practice in light of those facts. In the second place, we propose that conceptual schemas interpret the categories of the ontology in ways that are relevant to a particular view of a particular domain. In this way, the conceptual schema makes possible the specification of a set of causal constraints specific to the domain in question (as seen in light of a particular conceptualization, theory or paradigm). This allows an IS to inform explanation, prediction and practice. It also makes possible the specification of a set of dimensions that must characterize the relevant facts of that domain. In this way, the conceptual schema provides guidance for structuring the database. Designed in this way, the IS is not a merely passive repository of information but, given the fundamental explanatory categories derived from the ontology, it actively constrains the organization and interpretation of data.

THE ROLE OF EPISTEMIC CONSTRAINTS IN DEFINING THE DISTINCTION BETWEEN ONTOLOGIES AND CONCEPTUAL SCHEMAS

Drawing from our previous work [Fonseca and Martin, 2004], we here argue that the above mentioned distinction between computational ontologies and conceptual schemas is grounded in two necessary and complementary epistemic objectives which any adequate IS must facilitate. In Fonseca and Martin [2004], we identified two fundamental and complementary epistemic functions: identification of an invariant background and measurement of the object along dimensions of possible variation. Here we wish to exemplify this claim in light of the differences we see between ontologies and conceptual schemas.

Measurement, then, involves two aspects. First, it presupposes a background of invariants that constitute the context of interaction between the knower and the to-be-known. These assumptions include what Kant would have called a priori categories. They provide a basis for the interpretation of input data. As invariants, they permit the generation of hypotheses to explain observations of input data in terms of causes of those data in the domain of interest. These causes are thus recognized (abductively inferred) through the information classified in the IS, or they are, after their recognition, directly recorded in the IS. In either case, the presupposition of a background of invariants is necessary in order for the user of the data to assume that the differences in the data are a function of causally significant differences in the domain of interest and not simply accidentally related to it. The set of assumed invariants constitutes a kind of conceptual ground in the context of which the objects in the domain of interest emerge in intelligible form. We will attempt to clarify by working through two examples.

For example, suppose one were to attempt to infer (indirectly measure) the weight of smoke produced from burning a piece of wood by subtracting the weight of the remaining ash from the weight of the original wood. In this case, one would have assumed the conservation of mass – that the amount of mass remains invariant under conditions of burning (i.e., under the transformation from the state of wood to the state of smoke). On the basis of this presupposition one could assume that measured differences between the weights of wood and ash would correspond to a difference in the domain of interest – a difference equivalent to the weight of the smoke. But it is important to recognize that the principle of conservation is not amenable to strictly empirical demonstration. It is an a priori presupposition that enables such inferences as well as explanations of the data on which such inferences are based.

Consider another example based on Kant's proposals – one derived from the history of physics: Galileo's discovery, or invention, of the Law of Inertia. On our view, the ontology of classical explanatory science contains the presupposition that 'every change of state is caused'. This category of causation is not limited to physics but extends to all causally integrated domains of inquiry. But the concept of inertia in the form Galileo gave is specific to classical physics. In this context, we view Galileo's creative achievement to be the stipulation (at the level of a conceptual schema) that for the domain of mechanics, the relevant states are states of motion, and the relevant causes are forces. In this context, one derives the Law of Inertia: Every change of state (motion) is caused (by the application of a force).

Notice, moreover, that at this point one is given direction concerning the process of measurement. One might indirectly measure a gravitational force by observing changes in motion – as was done in the discovery of Uranus. In describing astronomical events so as to give an intelligible picture of the solar system, one would want to record both the states of motion of observable objects and the forces applied to them. This would constrain the dimensions that define the database in ways specified in the conceptual schema. At the same time, by linking the facts in the database to a general scientific ontology, the conceptual schema would render at least some of the regularities in the data causally intelligible, enable predictions and thus provide a basis for various forms of practice (e.g., traveling to the moon centuries later).

To summarize: with respect to the distinction between computational ontologies and conceptual schemas, we suggest that computational ontologies should be constructed so as to specify a priori assumptions concerning invariance which underlie the measurement and explanatory process, while conceptual schemas should relate those categories to dimensions of measurable variation in the domain of interest. A priori categories are thus central to both the explanation and specification of data. But the specification and explanation of data are distinct, if complementary, functions. An information system is a tool through which an experienced user is enabled to measure crucial features of a domain of interest. As such it must support the dual epistemic functions that enable specification and explanation.

THE COMPLEMENTARITY OF THEORY AND DATA IN ISAD

As discussed in section IV, we consider that ontologies are theories that explain a domain by revealing it as a coherent whole. They enable an ample understanding of the context in which the IS is used. In analyzing the difficulties of capturing, in an information system, the complete context in which the system will be used, Naur [1992] says "*the dependence of a theory on a grasp of certain kinds of similarity between situations and events of the real world gives the reason why the knowledge held by someone who has the theory could not, in principle, be expressed in terms of rules. In fact, the similarities in question are not, and cannot be, expressed in terms of criteria, no more than the similarities of many other kinds of objects, such as human faces, tunes, or tastes of wine, can be thus expressed*" (p. 40). Guarino [1998] is more explicit when he argues that "*application programs are still an important part of many ISs. They usually contain a lot of domain knowledge, which, for various reasons, is not explicitly stored in the database. Some parts of this knowledge are encoded in the static part of the program in the form of type or class declarations, other parts (like for example business rules) are implicitly stored in the (sometimes obscure) procedural part of the program*." (p.13). Sowa [2000] considers that a programmer trying to solve a problem has the knowledge to implement a solution, but the way of encoding this knowledge can vary from one individual to another. Both the programmer and the modeler have their own ontologies, and they can be either implicit or explicit. This variation is one of the reasons for creating ontology-driven information systems, to build the theories of programs before the system is created [Guarino, 1998].

The use of ontologies as the driver of conceptual schemas was suggested before in the literature. Wand and Weber [1989] suggested some of the differences between an ontology and a conceptual schema. They did not compare ontologies with conceptual schemas, but instead, highlighted how ontologies can improve the process of conceptual modeling from which conceptual schemas result.

It is important to notice that some work is underway in trying to establish mappings or a derivation between ontologies and conceptual schemas. El-Ghalayini et al. [2006] created a Transformation Engine with a mapping algorithm to generate a conceptual schema from a given domain ontology. Hess & Schliedera [2006] suggested a model to verify whether a conceptual schema conforms to an ontology. They do this through the validation of the mappings between the two. Jarrar et al. [2003] study the differences between specific ontologies and conceptual schemas. They show examples of the possible mappings between ontologies and conceptual schemas and introduce DOGMA, an ontology engineering framework, which enables the use of conceptual modeling methods for modeling and representing ontologies. Conesa et al. [2003] suggest that conceptual schemas can be created through a refinement of ontologies or that ontologies may play a supporting role for the designer of conceptual schemas. Fonseca et al. [2003] also suggest a way to establish mappings between ontologies and conceptual schemas in geographic information systems.

In summary, we have pointed to the intimate interaction between the specification of the data and the ontological assumptions to be encoded in the computational ontology. The relation we have noted between the complementary roles of computational ontologies and conceptual schemas is not merely a function of information systems in the modern sense, but is a consequence of the nature of fundamental epistemic constraints on the process of knowing. The reciprocal relation between computational ontologies and conceptual schemas is precisely what is necessary for the experience and measurement of the domain of interest to take place.

VI. CONCLUSIONS

Guarino [1998] coined the term ontology-driven information system for a kind of IS in which an explicit ontology plays a central role driving all of the system aspects and components. Nevertheless some authors consider that the idea of using ontologies as the foundation for IS development is not new. They argue that the early proposal of using conceptual schemas as a way to represent and store knowledge about an application domain [Tsichritzis and Klug, 1978] represents the same idea which Guarino is arguing for. It was implicit in their argument that a conceptual schema (the result of the modeling process) is essentially the same as a computational ontology (the result of the ontology engineering process).

We do not agree that computational ontologies and conceptual schemas are the same. In this paper we have argued that there are differences between ontologies and conceptual schemas. They belong to two different epistemic levels. They have different objects and are created with different objectives. While computational ontologies focus on explanation and information integration grounded in assumptions about invariant conditions that define the domain of interest, conceptual schemas focus on linking the general ontological categories with particular observations to be classified in IS.

We have argued that the use of the term *ontology*, or *computational ontology*, appeared as a result of the evolution of the field. For instance, we can look at the shaping of the idea of computational ontology as early as 1989 when Wand and Weber introduced the idea that a modeling method could be *reality* or *information-system* driven. This kind of principle-based separation, that later appeared in the work of others, is what led, and is still leading, as our work shows, to a progressive and necessary distinction of the two terms.

We have pointed to epistemic principles that support a natural distinction between computational ontologies and conceptual schemas. The distinction we made is grounded in a prior separation between complementary epistemic principles that must be satisfied by IS design.

Our discussion led to a prescription. We argued for the separation of the two terms as part of a principle-based policy for guiding modeling practice. Although they have a common origin in referring to the results of modeling activity, growing insight into the distinct epistemic principles that underlie them shows that they represent distinct and complementary aspects of the modeling process. The clear distinction we tried to show here was intended to clarify issues of information system design and thus promote effective design practices that ontology-driven IS are promising. If we hope for a continuing evolution of modeling, it is necessary to emphasize the distinction between computational ontologies and conceptual schemas. Our argument is that the field needs the two distinct concepts and by using both, IS research will be moved forward.

Much work still needs to be done in elucidating the links between ontologies and conceptual schemas. How can they support, validate and complement each other still remains to be completely clarified.

VIII. REFERENCES

Blackburn, S. (1996) *The Oxford dictionary of philosophy*. Oxford; New York: Oxford University Press.

Bodenreider, O., B. Smith, and A. Burgun. (2004) The Ontology-Epistemology Divide: A Case Study in Medical Terminology. *The 3rd International Conferenceon Formal Ontology in Information Systems (FOIS 2004), Turin, Italy, 2004*, pp. 185-195.

Caruso, F. and A. Umar (2004) "Architectures to Survive Technological and Business Turbulences," *Information Systems Frontiers* (6) 1, pp. 9-21.

Chen, P. S. S. (1976) "The Entity-Relationship Model: Towards a Unified View of Data," *ACM Transactions on Database Systems* (1) 1, pp. 9-36.

Conesa, J., X. d. Palol, and A. Olivé (2003) Building Conceptual Schemas by Refining General Ontologies, in *Database and Expert Systems Applications*, Berlin / Heidelberg, Germany: Springer.

Cui, Z., D. Jones, and P. O'Brien (2002) "Semantic B2B Integration: Issues in Ontology-based Applications," *SIGMOD Record Web Edition* (31) 1.

El-Ghalayini, H., M. Odeh, and R. McClatchey. (2006) Engineering Conceptual Data Models from Domain Ontologies: A Critical Evaluation. *4th International Conference on Computer Science and Information Technology (CSIT'06), Amman, Jordan, 2006*.

Fonseca, F., C. Davis, and G. Camara (2003) "Bridging Ontologies and Conceptual Schemas in Geographic Information Integration," *Geoinformatica* (7) 4, pp. 355-378.

Fonseca, F. and J. Martin (2004) "Space and Time in Eco-Ontologies," *AI Communications - The European Journal on Artificial Intelligence* (17) 4, pp. 259-269.

Fonseca, F. and J. Martin (2005) "Toward an Alternative Notion of Information Systems Ontologies: Information Engineering as a Hermeneutic Enterprise," *Journal of the American Society for Information Science and Technology* (56) 1, pp. 46-57.

Frank, A. (1997) Spatial Ontology, in O. Stock (Ed.) *Spatial and Temporal Reasoning*, Dordrecht, The Netherlands: Kluwer Academic, pp. 135-153.

Gruber, T. (1992) *A Translation Approach to Portable Ontology Specifications*. Knowledge Systems Laboratory, Stanford University KSL 92-71.

Gruber, T. R. (1995) "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human Computer Studies* (43) 5/6, pp. 907-928.

Guarino, N. (1998) Formal Ontology and Information Systems, in N. Guarino (Ed.) *Formal Ontology in Information Systems*, Amsterdam, Netherlands: IOS Press, pp. 3-15.

Guarino, N. and P. Giaretta (1995) Ontologies and Knowledge Bases Towards a Terminological Clarification, in N. J. I. Mars (Ed.) *Towards very large knowledge bases: knowledge building & knowledge sharing*, Amsterdam; Washington D.C.: IOS Press, pp. 25-32.

Hess, C. and C. Schliedera (2006) "Ontology-based verification of core model conformity in conceptual modeling," *Computers, Environment and Urban Systems* (30) 5, pp. 543-561.

Hirschheim, R. A., H.-K. Klein, and K. Lyytinen (1995) *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge; New York: Cambridge University Press.

Jarrar, M., J. Demey, and R. Meersman (2003) "On Using Conceptual Data Modeling for Ontology Engineering," *Journal on Data Semantics* (2800.

Kant, I. and N. K. Smith (2003) *Critique of pure reason*, Rev. 2nd edition. Houndmills, Basingstoke, Hampshire; New York: Palgrave Macmillan.

Kishore, R., R. Sharman, and R. Ramesh (2004) "Computational Ontologies and Information Systems I: Foundations," *Communications of the Association for Information Systems* (14), pp. 158-183.

Klein, H. and K. Lyytinen (1992) Towards a New understanding of Data Modelling, in C. Floyd, H. Zullighoven, R. Budde, and D. Kiel-Slavik (Eds.) *Software Development and Reality Construction*, Berlin: Springer-Verlag, pp. 203-220.

Kung, C.-H. and A. Solvberg (1986) Acitivity Modeling and Behaviour Modeling, in Post-conference edition T. W. Olle, H. G. Sol, and A. A. Verrijn Stuart (Eds.) *Information systems design methodologies: improving the practice: proceedings of the IFIP WG 8.1 Working Conference on Comparative Review of Information Systems Design Methodologies, Improving the Practice, Noordwijkerhout, The Netherlands, 5-7 May, 1986*, Amsterdam; New York: North-Holland Pub. Co., pp. 145-171.

McGuinness, D. L. (2003) Ontologies Come of Age, in D. Fensel, W. Wahlster, H. Lieberman, and J. Hendler (Eds.) *Spinning the semantic Web: bringing the World Wide Web to its full potential*, Cambridge, Mass.: MIT Press, pp. xxiii, 479.

Milton, S. K. (1998) Top-Level Ontology: The problem with Naturalism, in N. Guarino (Ed.) *Formal Ontology in Information Systems*, Amsterdam, Netherlands: IOS Press, pp. 85-94.

Milton, S. K. and E. Kazmierczak. (1999) Enriching the Ontological Foundations of Modelling in Information Systems. *The Information Systems Foundations Workshop - Ontology, Semiotics and Practice, Sydney, Australia, 1999*.

Naur, P. (1992) Programming as Theory Building, in P. Naur (Ed.) *Computing, a human activity*, New York: ACM Press, pp. 37-48.

Olivé, A. (2004) On the Role of Conceptual Schemas in Information Systems Development, in, vol. 3063 A. Llamos and A. Strohmeier (Eds.) *Reliable Software Technologies - Ada-Europe 2004, 9th Ada-Europe International Conference on Reliable Software Technologies, Palma de Mallorca, Spain, June 14-18, 2004*: Springer, pp. 16-34.

Parsons, J. and Y. Wand (2000) "Emancipating Instances from the Tyranny of Classes in Information Modeling," *ACM Transactions on Database Systems* (25) 2, pp. 228-268.

Quine, W. V. (1953) *From a Logical Point of View: 9 Logico-Philosophical Essays*. Cambridge: Harvard University Press.

Rational Software Corporation (1997) "The Unified Language: Notation Guide, Version 1.1," Rational Software Corporation, (July 1997).

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy et al. (1991) *Object-Oriented Modeling and Design*. Englewood Cliffs, NJ: Prentice-Hall.

Scheer, A.-W. (2003) *Business process change management: ARIS in practice*. Berlin; New York: Springer.

Schmid, H. A. (1983) A comparative survey of concepts for conceptual information models, in J. Bubenko (Ed.) *Information Modeling*, Sweden: Chartwell-Bratt Ltd.

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

Smith, B. (2003) Ontology, in L. Floridi (Ed.) *The Blackwell Guide to the Philosophy of Computing and Information*, Malden, MA: Blackwell, pp. 155-166.

Sowa, J. (2000) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brook/Cole.

Spyns, P., R. Meersman, and M. Jarrar (2002) "Data Modelling versus Ontology Engineering," *SIGMOD Record* (31) 4, pp. 12-17.

Sugumaran, V. and V. C. Storey (2002) "Ontologies for Conceptual Modeling: Their Creation, Use, and Management," *Data & Knowledge Engineering* (42) 3, pp. 251-271.

Tsichritzis, D. and A. C. Klug (1978) "The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Dabatase Management Systems," *Information Systems* (3) 3, pp. 173-191.

Uschold, M. and M. Gruninger (2004) "Ontologies and Semantics for Seamless Connectivity," *SIGMOD Record* (33) 4, pp. 58-64.

Wand, Y. and R. Weber (1989) An Ontological Evaluation of Systems Analysis and Design Methods., in E. D. Falkenberg and P. Lindgreen (Eds.) *Information Systems Concepts: An In-Depth Analysis*: Elsevier Science Publishers, pp. 79-107.

Wand, Y. and R. Weber (2002) Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda, in *Information Systems Research*, vol. 13, pp. 363-376.

Wand, Y. and R. Weber (2004) "Reflection: Ontology in Information Systems," *Journal of Database Management* (15) 2, pp. iii-vi.

Wand, Y. and R. A. G. Weber (1990) An Ontological Model of an Information System, in *IEEE Transactions on Software Engineering*, vol. 16, pp. 1282-1292.

Winter, S. (2001) "Ontology: Buzzword or Paradigm Shift in GIScience?" *International Journal of Geographical Information Science* (15) 7, pp. 587-590.

Wyssusek, B. (2004) Ontology and Ontologies in Information Systems Analysis and Design: A Critique. *The Tenth Americas Conference on Information Systems, New York, New York, 2004*, pp. 4303-4308.

Zdonik, S. and D. Maier (1990) Fundamentals of object-oriented databases, in S. Zdonik and D. Maier (Eds.) *Readings in Object-Oriented Databases*: Morgan-Kaufman, pp. 1-32.

Zuniga, G. (2001) Ontology: its transformation from philosophy to information systems. *The 2nd international conference on Formal Ontology in Information Systems (FOIS'01), Ogunquit,ME, 2001*, pp. 187-197.