View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Communications of the Association for Information Systems

Volume 14 Article 8

August 2004

Computational Ontologies and Information Systems I: Foundations

Rajiv Kishore
State University of New York at Buffalo, rkishore@buffalo.edu

Raj Sharman
rshrman@buffalo.edu, rshrman@buffalo.edu

Follow this and additional works at: https://aisel.aisnet.org/cais

Recommended Citation
Kishore, Rajiv and Sharman, Raj (2004) "Computational Ontologies and Information Systems I: Foundations," *Communications of the Association for Information Systems*: Vol. 14 , Article 8.
DOI: 10.17705/1CAIS.01408
Available at: https://aisel.aisnet.org/cais/vol14/iss1/8

This material is brought to you by the AIS Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Communications of the Association for Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Communications of the Association for Information Systems

COMPUTATIONAL ONTOLOGIES AND INFORMATION SYSTEMS: I. FOUNDATIONS

Rajiv Kishore
Raj Sharman
Ram Ramesh
School of Management
State University of New York at Buffalo
rkishore@buffalo.edu

ABSTRACT

This paper provides a state-of-the-art review about computational ontologies to raise awareness about this research area in the IS discipline and to explore areas where IS researchers can engage in fruitful research. This paper discusses the basic foundations and definitions pertaining to the field of computational ontologies. It reviews the intersection of computational ontologies with the IS discipline. It also discusses methods and guidelines for developing computational ontologies. The paper concludes with recommendations for important and emerging directions for research.

The technical aspects of ontologies are presented in a companion paper [Sharman et al., 2004]. The companion paper provides a comprehensive review of the formalisms, languages, and tools used for specifying and implementing computational ontologies.

Keywords: ontology, computational ontologies, formal ontologies, information systems ontologies, information systems concepts, ontology-driven information systems, is modeling grammars, ontology development tools, ontology representation formalisms, ontology specification languages, ontological engineering, ontology mining, ontology metrics.

I. INTRODUCTION

Ontology as “the metaphysical study of the nature of being and existence” [Princeton University, 1997] is as old as the discipline of philosophy. More recently and more concretely, it was defined by a contemporary philosopher as “the science of what is, of the kinds and structures of objects, properties events, processes, and relations in every area of reality” [Smith, 2003]. While ontology remains a fertile area of research in philosophy, it is also a matter of inquiry, development, and application for quite some time now in disciplines related to computation, information, and knowledge (e.g., artificial intelligence, knowledge representation, information science, library science, and database management) because of the need in these fields to categorize and structure entities and concepts of interest. However, in the last decade or so, the interest and application of ontological principles exploded in a number of disciplines, including chemistry, enterprise management, geography, linguistics, mathematics, medicine, and sociology. Each discipline sought to create domain-specific ontologies that structure concepts and things pertaining to their discipline. As a result, while the philosophy discipline still treats ontology in the singular because it deals with the nature of all reality, other disciplines take a rather narrower view of ontology and use it only in the limited context of domain-specific reality. Consequently, the notion of ontology is now plural and not singular, because each individual ontology in a particular domain/discipline deals with only a limited portion of reality that is pertinent to that domain/discipline.

Further, the rush to create ontologies in diverse disciplines and domains is not driven simply by an academic desire to understand the nature of reality in that domain. While that goal may be true, the interest and the rush rather emanate from another very pragmatic reason. The goal is to structure and codify knowledge about the concepts, relationships, and axioms/constraints pertaining to a domain in a computational format so that it can be manipulated and used by the computer to aid human and machine agents in their performance of tasks within that domain.

Ontology, therefore, no longer remains a subject matter of esoteric inquiry in the philosophy discipline. It is a core subject of inquiry and development in the information systems (IS) discipline. We draw this conclusion because information systems are knowledge artifacts that capture and represent knowledge about entities, relationships, constraints, and processes in a particular application area [Kishore et al., 2004b] and IS professionals and researchers deal with issues of identifying, capturing, and representing such domain knowledge within information systems.

The objective of this paper and its companion paper [Sharman et al., 2004] is to provide CAIS readers with a comprehensive review of the basic notions about ontologies in the context of the IS discipline. This paper, Part I: Foundations consists of five sections. Section I is this introduction section which essentially motivates the need for this paper. Section II discusses the foundations and definitions pertaining to computational ontologies. Section III discusses the key topics of creating and applying ontologies in the IS discipline. Section IV discusses topics in ontological engineering, i.e., developing computational ontologies. Finally, section V provides future research directions. Figure 1 is a roadmap highlighting the topics that are covered in this paper.

The companion paper [Sharman et al., 2004] provides a comprehensive review of the formalisms, languages, and tools used for specifying and implementing computational ontologies.

II. FOUNDATIONS AND DEFINTIONS

COMPUTATIONAL ONTOLOGIES VS. PHILOSOPHICAL ONTOLOGY

We use the term computational ontology in this tutorial to represent all kinds of ontologies that are being developed or may be developed in specific domains/disciplines. As discussed in Section I, computational ontologies and philosophical ontology differ in at least two ways:

1. Their goals are different. Philosophical ontology is purely an academic pursuit to know about the nature of reality. Computational ontologies add the goal of being implemented and used computationally in the pursuit of other pragmatic objectives in a specific application.

2. They differ in scope. Philosophical ontology deals with all reality in *the* entire universe of discourse while computational ontology deals with only the “reality of interest” and in only a bounded (limited) universe of discourse.

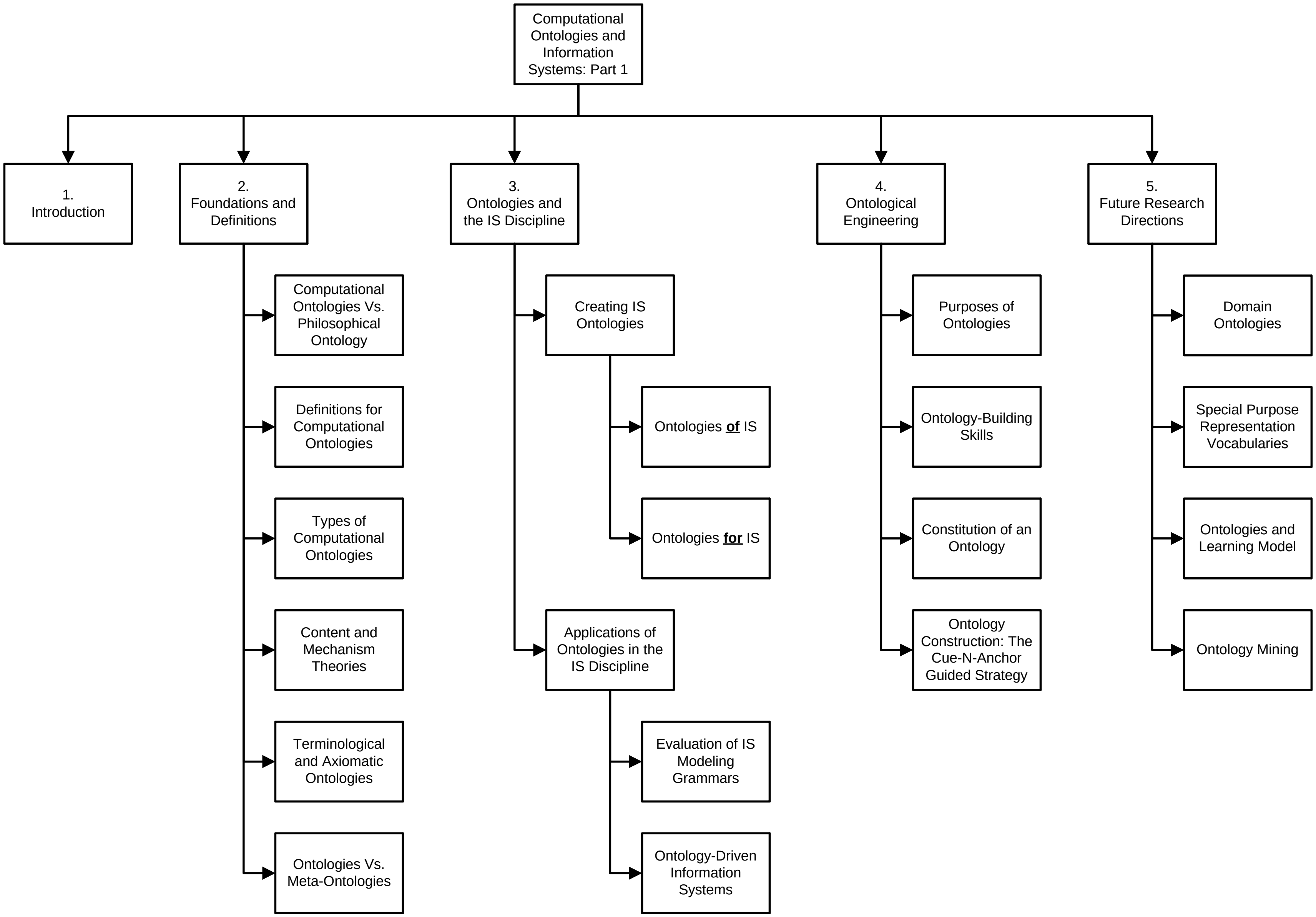

Figure 1. Roadmap for this Paper

We follow the philosophical term “universe of discourse,” first introduced by De Morgan in 1846 [Bergman and Paavola], as it is defined precisely and accurately by philosophers to capture and convey the notions of ontology. Charles S. Peirce, a noted philosopher, defines it as:

> *"The universe of discourse is the aggregate of the individual objects which ‘exist,’ that is, are independently side by side in the collection of experiences to which the deliverer and interpreter of a set of symbols have agreed to refer and to consider"* [Peirce as quoted in Robin, 1967].

Furthermore, the philosophical ontology pertains to an unbounded universe of discourse while the computational ontology pertains to only a bounded universe of discourse. Again, we follow the long-standing philosophical tradition in making this distinction.

> *"De Morgan and his followers frequently speak of a ‘limited universe of discourse’ in logic. An unlimited universe would comprise the whole realm of the logically possible. In such a universe, every universal proposition that is not tautologous is false; every particular proposition that is not absurd is true. Our discourse seldom relates to this universe: we are either thinking of the physically possible, or of the historically existent, or of the world of some romance, or of some other limited universe.” [Peirce, 1893].*

Our notion of a bounded universe is also very similar to Sowa’s notion of a microworld [Sowa, 1999, p. 52].

DEFINITIONS FOR COMPUTATIONAL ONTOLOGIES

Most efforts in developing the principles, tools, techniques, and representations about computational ontologies took place in the artificial intelligence (AI) and computer science (CS) communities. As a result, several definitions for computational ontologies are available in the AI and CS literature. We provide below some of the more commonly-cited definitions of computational ontologies from the AI/CS area. For example, Gruber states that

> *“an ontology is a formal explicit specification of a shared conceptualization.”* [Gruber, 1993a]

Conceptualization itself is defined by Genesereth and Nilsson [1987] as

> “*the objects, concepts and other entities that are assumed to exist in some area of interest and their inter-relationships.”* [Genesereth and Nilsson, 1987]

Guarino suggests that

> *“a conceptualization contains many ‘world structures’, one for each world. It has both extensional and intentional components”.*[Guarino, 1998]

The definition of computational ontologies by Gruber is quite succinct and popular and we adopt his definition in this paper.

TYPES OF COMPUTATIONAL ONTOLOGIES

A computational ontology can be of several types. AI researchers take the view that it can either be a representation vocabulary or a body of knowledge (i.e., a knowledge base) [Chandrasekaran et al., 1999]. This distinction is essentially a distinction between intension and extension [Sowa, 1984, p. 11] about the universe of discourse. The representation vocabulary in the discourse universe provides symbols for the concepts in the universe, thus being the intension for the universe. The knowledge base is essentially the set of all referents to which a concept may refer to, thus being an extension for this concept. For example, in the MibML ontology that we are developing for the bounded universe of multiagent-based integrative business information systems [Kishore et al., 2004a, Zhang et al., 2003, Zhang et al., 2004], the notion of Goal provides an extension for this universe because it is a symbol and part of the representation vocabulary for this universe. On the other hand, all possible types of goals in this universe (including such goals as replenish inventory or supply orders) form the extension of this universe and are part of the knowledge base for this universe. However, it is to be clearly understood that the extension in the ontological context does not include specific real instances and each member of the extension set is essentially a class of actual instances that exist in time and space. For example, in the universe of medicine, disease is part of the representation vocabulary, a symbol, whereas flu and typhoid are part of the extension of the ontology. A particular occurrence of the disease flu (e.g., John suffering from flu at a particular time in a particular place) will be an actual instance of the class flu and will not be a part of the medicine ontology.

Ontologies can also be distinguished in terms of the level of knowledge they capture and represent. Top-level ontologies start with very general top-level concepts such as a “Thing” and provide a taxonomy of top-level concepts. There are a number of formal top-level ontologies, some of which are implemented on computers. Examples of formal top level ontologies, i.e., ontologies that have been specified using some formal logics, include Bunge’s ontology [Bunge, 1977], Sowa’s upper level ontology [Sowa, 1999], General Ontological Language (GOL) [Degen et al., 2001], and IEEE Standard Upper Ontology (SUO) [IEEE, 2003]. Some popular top-level implemented ontologies include CYC [Lenat and Guha, 1990], WordNet [Miller, 1990], and Generalized Upper Model [Bateman et al., 1994]. Details about these ontologies are provided in Appendix I. The top-level categories of CYC and Sowa’s ontology are shown in Figures 2 and 3, respectively.

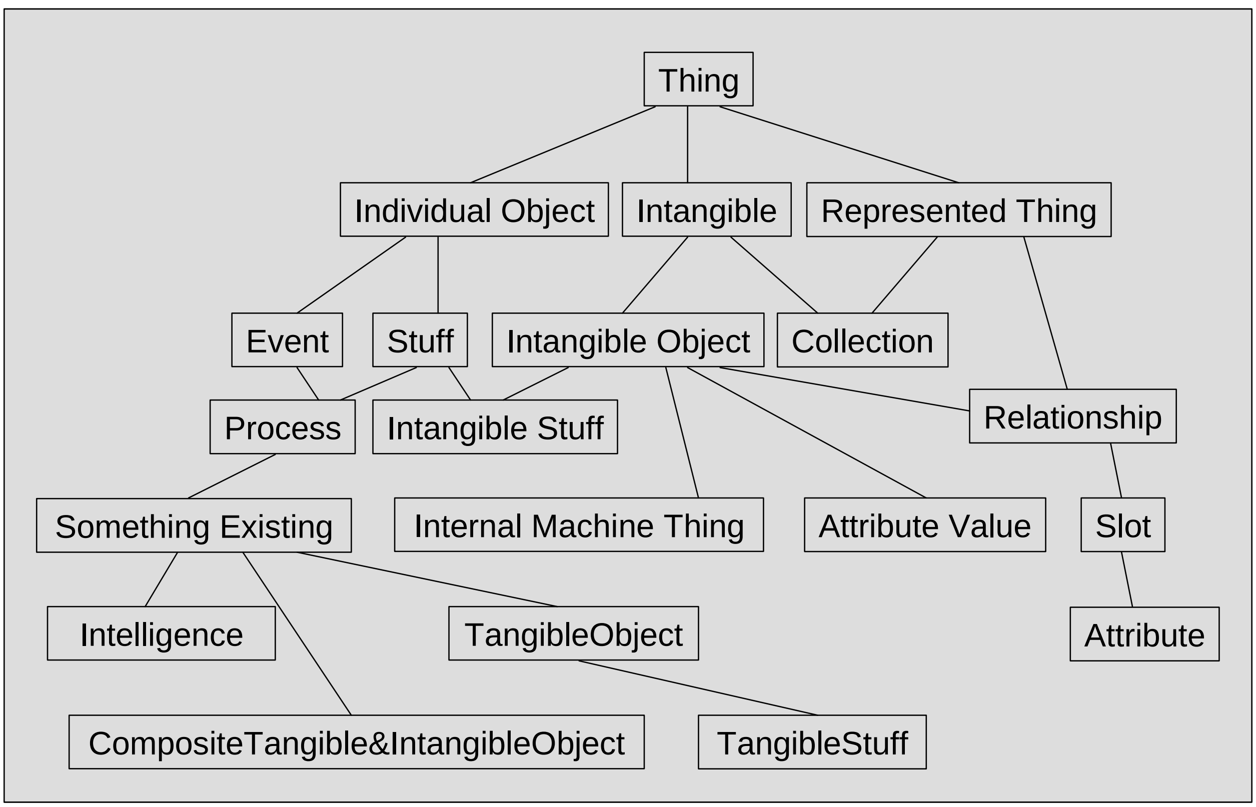

Figure 2: Top-level Categories in CYC (adapted from Lenat and Guha, 1990).

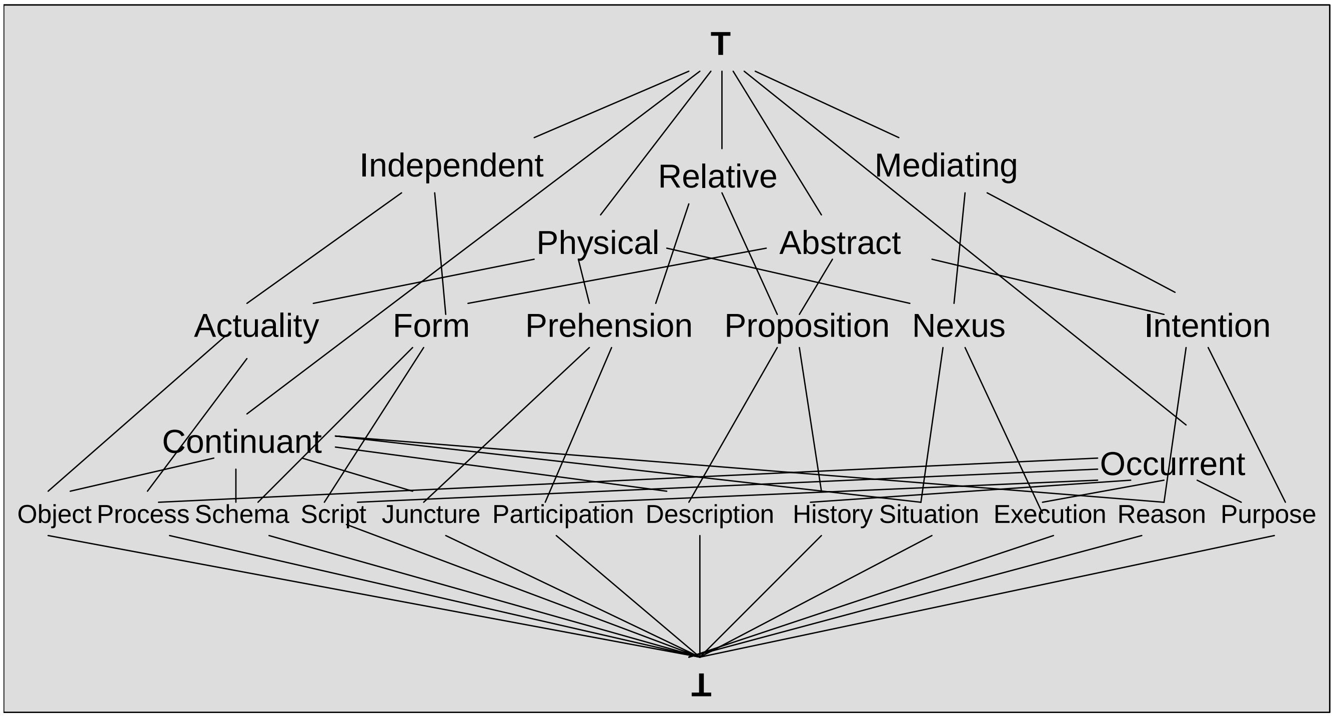

Figure 3: Sowa's Upper-Level Ontology (adapted from Sowa, 2000).

While the top-level implemented ontologies pertain to an unbounded universe of discourse, they are classified as computational ontologies in this paper because they are implemented computationally. Lower-level ontologies pertain to bounded universe of discourses and are referred to as application, domain, and task ontologies in the literature. We do not distinguish among them and simply refer to them as computational ontologies that pertain to a bounded universe of discourse.

CONTENT AND MECHANISM THEORIES

The AI literature has generally distinguished between content and mechanism theories [Chandrasekaran et al., 1999]. Content theory is similar to the notion of declarative knowledge while mechanism theory is essentially procedural knowledge [Smith, 2003]. Some authors use ontology to refer only to content theory or declarative knowledge because it captures the "what" knowledge. Others define the notion of method and task ontologies to capture procedural knowledge about a domain [Studer et al., 1998, Studer et al., 1996].

TERMINOLOGICAL AND AXIOMATIC ONTOLOGIES

Another distinction made in the literature is between terminological and axiomatic ontologies [Sowa, 1999]. Categories in a terminological ontology need not be fully specified by axioms and definitions but can simply be a collection of categories and terms. On the other hand, in axiomatic ontologies, categories are distinguished by axioms and definitions stated in logic or some computer-oriented language that can be automatically translated into logic. A comprehensive formal computational ontology will, therefore, consist of things and concepts that can be primitive or compound, and relationships of a variety of types among them that could include structural, logical, behavioral, spatial, and temporal types of relationships, all bound together and defined in terms of formal axioms and rules.

ONTOLOGIES VS. METAONTOLOGIES

Finally, it is also important to distinguish between ontologies and meta-ontologies. While a computational ontology captures knowledge about a universe of discourse, a meta-ontology is the language that is used to represent the ontology [Uschold and Gruninger, 1996]. This representation language is itself an ontology and it can be informal (e.g., English), semi-formal (e.g., UML), or formal (e.g., First Order Logic).

It is important to distinguish between ontologies and metaontologies in terms of the different levels of representation languages that are used to represent knowledge about a particular universe of discourse. Tarski [1982] developed a theory of stratified metalevels to distinguish between languages and meta-languages and what they can and cannot refer to. In this theory, a language at level 0 (L_0) can refer to entities in some universe of discourse D but L_0 cannot refer to its own symbols or the truth values of its own statements. The metalanguage L_1 can refer to the original D, to the symbols of L_0, to the truth values of statements in L_0, and to the relationships between the language L_0 and D. The universe of discourse for L_1 is therefore the union of L_0 and D. Similarly, the universe of metalanguage L_2 is the union of L_1, L_0, and D. In general, the n^{th} level metalanguage includes all the languages beneath itself, their domains of discourse, and the truth values of statements in those languages. But no language can refer to its own symbols or to the truth values of its statements. For example, any of the three languages – the English language, the UML language, or predicate logic – can be used as a metalanguage for representing the symbols in the domain of discourse of medicine to create a medical ontology. However, it should be noted that all the three metalanguages – English, UML, and predicate logic – themselves refer to their respective universe of discourse and each of them, therefore, represent an ontology in their own right. However, as has been noted, uninterpreted logic languages such as predicate logic, conceptual graphs, or Knowledge Interchange Format (KIF) are ontologically neutral[1] because they impose no constraint on the subject matter to be represented and, thus, any lower-level ontology (i.e., lower-level language) can be represented using them [Sowa, 1999, p. 492]. This approach is preferred to ensure minimal encoding bias [Gruber, 1993b].

III. ONTOLOGIES AND THE INFORMATION SYSTEMS DISCIPLINE

One of the major goals in every discipline is to define and categorize terms, concepts, and phenomenon of interest to that discipline to create a common language and to advance knowledge in the discipline. While all disciplines deal with the philosophical notion of ontology, the role of ontology is much more pronounced and its impact is much more widespread in the IS discipline. Unlike other disciplines, one of the major pursuits in the IS discipline is to create knowledge about the phenomenon of "knowledge" itself and of its raw material – information [Dretske, 1981, Nonaka, 1994]. Further, information systems artifacts are themselves embodiments of human knowledge [Armour, 2000, Kishore et al., 2004b] and they, therefore, explicitly or implicitly encapsulate ontologies about the discourse universes represented by them. Thus, in our opinion, the IS discipline is much closer to both ontology and epistemology than other typical business and management disciplines.

Research in the IS discipline that deals with ontologies in an explicit manner can be classified under two major streams.

1. Creating IS ontologies

---

[1] This statement is not entirely true because any language at any level makes assumptions about the fundamental entities and relationships that exist in its universe of discourse. As aptly stated by Sowa, "the starting primitives cannot be defined in terms of anything more primitive; they can only be specified indirectly by their relationships to other concepts in the system" [Sowa, 1999, p. 76]. Therefore, it is wise to use a meta-ontology that is not only formally specified and rigorous but also exhibits minimal ontological commitment [Gruber, 1995].

2. The application of philosophical ontology and computational ontologies.

These streams are discussed in more detail in the following paragraphs.

CREATING INFORMATION SYSTEMS ONTOLOGIES

Creating ontologies about the IS universe (i.e., the IS discipline) is a fertile area of research pursuit in the IS discipline as it is perhaps an area of inquiry in other disciplines. However, this pursuit goes beyond other disciplines in that the IS ontologies are not created just for the sake of understanding and reasoning about the IS universe but they also serve as meta-ontologies for the IS discipline. Therefore, we classify IS ontologies into two broad types.

1. Ontologies of information systems
2. Ontologies for information systems.

These two types of ontologies are now discussed.

Ontologies of Information Systems

Ontologies of information systems include attempts to capture a comprehensive ontology of all the concepts and relationships among them that are pertinent to understanding and reasoning about the information system and related universes. These ontologies are essentially the extension of the information systems discipline or a knowledge base of all information systems concepts. Such ontologies range from simple taxonomic structures to more elaborate and axiomatic ontologies.

Examples of simple taxonomic ontologies that capture the knowledge structure within the IS and computing disciplines are the IS keyword lists [e.g., Barki et al., 1988, Barki et al., 1993] and the ACM Computing Classification System [ACM]. Closely related to and a part of the IS discipline is organizational knowledge management. Researchers also attempted to create organizational knowledge management ontologies [e.g., Holsapple and Joshi, 2001, http://www.brint.com/].

More elaborate and axiomatic ontologies include the Bunge, Wand and Weber's IS ontology now commonly known as the BWW ontology [Wand and Weber, 1989, Wand and Weber, 1990, Wand and Weber, 1993] which is based on Bunge's philosophical ontology [Bunge, 1977, Bunge, 1979], the Framework for Information Systems Concepts (FRISCO) ontology produced by the FRISCO task group within Working Group 8.1 of the International Federation of Information Processing (IFIP) [Falkenberg et al., 1998], and Alter's work systems and information systems ontologies [Alter, 1999]. These three IS ontologies are similar in that they attempt to identify and formalize the fundamental concepts in the IS discipline but differ greatly in terms of the amount of detail they capture in the ontology about the IS universe.

The FRISCO ontology is perhaps the most detailed ontology. It not only provides an informal ontology with 37 concepts, 15 assumptions, and 41 textual definitions, but is also a formal ontology that is expressed using first order logic and contains 15 primitives, 2 axioms, 11 functions, and 49 formal definitions. Alter's ontology is an informal ontology and provides a total of 106 "fundamental concepts" (FCs) including 10 first layer FCs and 96 second layer FCs. Wand and Weber's ontology is a formal ontology primarily geared towards analyzing the static and dynamic aspects of information systems and to examine the question of what constitutes a good decomposition in information systems.

Ontologies for Information Systems

Ontologies for information systems are similar to the first category of IS ontologies in that they also attempt to identify and capture the fundamental concepts about the IS universe. However, these ontologies differ from ontologies in the first category in that these ontologies are specifically and expressly designed as IS modeling/programming languages. They contain appropriate syntax and semantics to be used for modeling and implementing specific information systems artifacts (e.g., system models, entire systems, and skeletal systems or frameworks) [Kishore et al., 2004b].

These ontologies are the intension of the IS discipline because they provide a representation vocabulary for modeling and representing information systems. This category of ontologies can be further sub-divided into three sub-categories:

- IS modeling grammars,
- general-purpose computer programming languages, and
- special-purpose computer languages/formalisms.

Grammars. IS modeling grammars are used for modeling of information systems artifacts. Some of the commonly-used grammars include the ER, Data Flow Diagram (DFD), PetriNet, State Transition Diagram, StateCharts, and UML modeling grammars.

General-Purpose Computer Programming Languages, such as C, C++, and Java languages, also qualify as ontologies for information systems because they embody the possible concepts and relationships in a universe of discourse, even though this universe is closer to the computing machine and mathematics than to the social and organizational aspects that are an integral part of business information systems.

Special-Purpose Computer Languages/Formalisms, such as KIF, conceptual graphs, OIL/DAML, and Ontolingua are designed specifically for developing and implementing application and domain ontologies and are discussed in more detail in the companion paper [Sharman et al., 2004]. While ontologies in this sub-category can be readily identified as meta-ontologies, IS modeling grammars and general-purpose computer programming languages are also essentially meta-ontologies. For example, the ER formalism can be used to create an ontology of entities and relationships in a particular domain (such as in the human resources area within a firm or more generally within an industry). State Transition Diagrams can be used to create an ontology of states, triggers, and transitions that various entities within a domain undergo. In a manner similar to special-purpose ontology development languages, general-purpose programming languages can also be used to implement ontologies on the computer although these languages may not be as efficient as special-purpose languages because they are designed primarily to implement information systems and not ontologies.

Ontologies for information systems differ in their formality. While all computer programming languages – both general-purpose and special-purpose – provide very formal syntax and semantics for developing and implementing ontologies and information systems, some IS modeling languages are quite formal (such as PetriNets and StateCharts) while others are only semi-formal (such as ER model and DFDs).

APPLICATIONS OF ONTOLOGIES IN THE INFORMATION SYSTEMS DISCIPLINE

A more recent stream of research deals with application of philosophical (top-level) and computational ontologies in the IS discipline. Research is being undertaken in this stream in the two fertile areas discussed next.

Evaluation of IS Modeling Grammars using Philosophical Ontologies

As discussed in the previous subsection, IS modeling grammars are themselves ontologies that provide symbols and language for modeling of the real world. However, the symbols, axioms, and rules these grammars provide and the assumptions they make are not always explicit or obvious. Further, these grammars may provide guidelines that are sometimes confusing or even conflicting. For example, the ER formalism provides two basic constructs – entities and relationships. The question is when something is an entity and when it is a relationship. For example, is marriage between two individuals an entity or a relationship? The basic ER formalism is silent about this aspect and a data modeler is free to model marriage as either an entity (with its own attributes, e.g., data of marriage) or as a relationship. However, the constructs and semantics of much narrower ontologies such as the IS modeling grammars can be evaluated using top-level philosophical ontologies.

Several researchers used such an ontological analysis approach to evaluate the notation and semantics of various conceptual modeling grammars. For example, Wand et al. [2000] evaluated the relationship construct in the ER model based on the BWW ontology. Their ontological analysis not only provides precise definitions for the conceptual modeling constructs, but also derives rules for the use of the relationship construct in the ER conceptual modeling grammar. Milton et al. [2003/4, 2002] have proposed a framework based on Chisholm's ontology to evaluate and compare various data modeling languages including Entity-Relationship (ER) Model, Functional Data Model, Semantic Data Model, NIAM, and OMT's Object Model. Weber and Zhang [1996] examined and indicated the ontological deficiencies of Nijssen Information Analysis Method (NIAM) using the BWW ontology. Green and Rosemann [2000] also used the BWW ontology to analyze the five views – process, data, function, organization and output – provided in the Architecture of Integrated Information Systems (ARIS) [Scheer, 1999]. In addition, the BWW ontology was used to evaluate data flow diagrams [Wand and Weber, 1989], object-oriented modeling [Evermann and Wand, 2001, Parsons and Wand, 1997, Takagaki and Wand, 1991], and reference models in information systems development [Fettke and Loos, 2003].

Further, Wand and Weber [2002] also identified evaluation of the expressiveness of conceptual modeling grammars to assess their strengths and weaknesses as a future research opportunity.

Ontology-Driven Information Systems

There is growing recognition that ontological principles and concepts need not be restricted to the traditional domains of knowledge inquiry, but can be fruitfully applied to and developed further in the broader field of information systems [Guarino, 1998, Studer et al., 1998, Uschold and Gruninger, 1996]. This recognition led to the notion of "ontology driven information systems" [Guarino, 1998], a concept that, although in a preliminary stage of development, opens up new ways of thinking about ontologies and IS in conjunction with each other. Guarino suggests that the idea of ontology-driven IS covers both the temporal and the structural dimensions of IS. A brief overview about the notions of ontology-driven information systems as developed by Guarino [1998] follows; however, the reader is referred to the original paper by Guarino for a complete discussion of this topic.

Guarino's View. From the perspective of the temporal dimension, ontologies can be used in IS both during the development time or run-time. When domain and task ontologies [Van Heijst et al., 1997] are used during development time, the semantic content about the domain contained within those ontologies can be easily transformed and translated into IS components, thereby enabling knowledge reuse, reducing the cost of conceptual analysis, and assuring the ontological adequacy of the IS [Guarino, 1998]. Guarino [1998] also suggests that even a high-level generic ontology consisting of coarse domain-level distinctions among the basic entities of the world and meta-level distinctions about kinds of classes and kinds of relations can also be used as a tool, similar to a CASE tool, during development time to increase quality of the analysis process. Ontologies can also be used during run-time as another component of an ontology-driven information system to enable communication among software agents. It is to be noted that software agents communicate with each other using messages whose meaning can be understood with reference to an ontology that the communicating agents commit to using.

From the perspective of the structural dimension of IS, all the three components of information systems (databases, user interfaces, and application programs) can use ontologies in their own ways [Guarino, 1998]. Here are some examples:

- In the database component, ontologies such as WordNet [Princeton University, 1997] can be used effectively during the development time for requirements analysis and conceptual modeling. Further, the resulting conceptual model for the domain application can be converted into a computer-processable ontology which can support "intensional[2] queries" to provide information about the meta-data content of a particular database.

- A common top-level ontology can be used for information integration because it can assist in developing a common conceptual schema for a large data warehouse by integrating a number of heterogeneous conceptual schemas for underlying databases.

- Domain ontologies can be used to generate user interfaces that check for constraint violations which are encapsulated within the generating ontologies, as was done in the Protégé Project [Guarino, 1998].

- Ontologies can be used effectively in both the static and procedural portions of application programs [Guarino, 1998]. The static portion of application programs encode some knowledge about the domain in terms of class and type declarations whereas the procedural portion of the programs embed knowledge about domain procedures and business rules. If procedural knowledge embedded within a program is explicitly represented outside the program rather than implicitly represented within the program, the system turns into a classical knowledge-based system which can be supported by a core knowledge base and ontologies [Guarino, 1998].

Thus, ontologies can be used in several areas in information systems both during development time and run-time. These applications are covered under the rubric of ontology-driven information systems. Further, as can be seen from the above discussion, the notion of ontology-driven information systems is much broader in scope than the traditional knowledge-based or database-driven systems, which can in fact benefit from an application of computational ontologies.

IV. ONTOLOGICAL ENGINEERING

The discussion in Sections II and III lays out the foundations of Ontology-Driven Information Systems in terms of their basic principles and structure. In this section, we focus on the methodology for conceptualizing and developing ontologies and extending them to information systems design. The methodology for ontology construction can be viewed as the backbone of the emerging discipline of Ontological Engineering.

Ontological engineering is concerned with finding the right answers to the following key questions:

- What is the purpose for which an ontology is needed?
- What skills are needed in conceptualizing and building the ontology?
- What constitutes the proposed ontology?
- What methodology is to be used in ontology development?

In many ways, ontological engineering can be likened to the process of traditional information systems development, where we begin with informal and formal systems analysis, continue with conceptualization of system models, leading to logical and physical designs of systems. Ontological engineering is quite similar, except that the domains of analysis can be very large and the knowledge capture and representation mechanisms can be quite varied. Therefore, ontology

[2] The term 'intension" was defined in Section II.

developers need to bring a wide array of skills to the ontology design endeavor. We synthesize an approach to ontological engineering by drawing from the ontology literature on the various facets of ontologies and their development in this section, keeping our focus on the four key bulleted questions above.

PURPOSES OF ONTOLOGIES

Noy and McGuinness [Noy and McGuinnss, 2002] summarize the following major purposes:

1. enable a shared understanding of the structure of information among people and agents,
2. enable information reuse in applications,
3. make the assumptions underlying an IS implementation explicit and well-understood,
4. specify the knowledge embodied in an ontology at an appropriate level of granularity (universe, bounded universe, domain, operational), and
5. apply the ontological structures at different stages of IS development: analysis, conceptualization and design.

In particular, these broad purposes translate to following specific objectives:

1. Can we create some high-level templates that systems analysts could use to capture data on user requirements and structure them in some standardized manner [Storey, 2002]?
2. Can we enable communication among analysts and interoperability among systems when dealing with diverse system components [Uschold et al., 1998]?
3. Can we create design templates at various levels of granularity that would lead to rapid systems design and development [Kishore et al., 2004b]?
4. Can we enable various systems engineering requirements such as re-usability, search for services in some repository, develop and maintain reliable systems and ensure persistent systems growth [Jasper, 1999]?
5. Can we enable interoperability among heterogeneous systems through a shared understanding at a meta-level ontology [Paton et al., 1999]?

Obviously, ontologies have been developed and continue to be developed for such diverse objectives. In particular, based on a study of various ontologies, Jasper and Uschold [1999] identify four broad categories of ontologies:

- Ontology for knowledge reuse,
- Ontology as specification,
- Ontology as a provider of common access of heterogeneous information and
- Ontology as a search mechanism.

In any case, the fundamental step in ontological engineering is a precise enunciation of the objectives for which an ontology is needed. Clearly, the scope and thrust of ontological engineering will increase as more needs for ontological support in IS development arise.

ONTOLOGY-BUILDING SKILLS

To build meaningful and tractable ontologies, the following skills are needed:

- Conceptual modeling skills
- Domain-specific expertise

- Systems engineering skills.

The conception of an ontology basically begins with an enunciation and modeling of the concepts, their relationships, and behaviors embodied in an ontology. As a result, extensive conceptual modeling constitutes the core of the ontology development process. The scope and thrust of an ontology would determine the extent of domain expertise and systems engineering skills required. When a universe of discourse is vast or when an ontology is used as a purely explanatory device as in the philosophical discipline, the emphasis on ontology-building skills shifts more towards the core conceptual modeling skills. However when an ontology becomes more and more applied, especially as an IS tool for software engineering, the emphasis tends to be more significantly shared among the three skills. Hence the objectives for which an ontology is developed guide the skills needed rather closely.

CONSTITUTION OF AN ONTOLOGY

As described in Section II, the perspectives on what constitutes an ontology vary significantly from the philosophical to the more applied disciplines. We adopt a more applied view in this section, especially in the IS context. Using Gruber's [1993a] definition of an ontology as a specification of conceptualization, we can regard it as an embodiment of:

- concepts,
- relationships, and
- behaviors

These three components constitute the structure of an ontology.

Concepts contain distinctions, are related to one another, and display various behaviors. The distinctions are usually described as in a taxonomy with a view to describe the various concepts constituting the universe of discourse.

A similar taxonomy of distinctions among the various possible relationships could also be envisioned. We term these distinctions of concepts and relationships as basic. Then, applying the different types of relationships among the concept distinctions, various binary and n-ary related concept distinctions could be derived. We term such distinctions as extended.

The behavior construct describes the dynamics of the basic and extended distinctions such as interactions, communication, workflow and states of existence of the distinctions [Eriksson and Penker, 2000]. Together, the basic and extended distinctions with their behaviors yield both axiomatic principles of ontology construction and usage; combining the axioms with the basic assumptions and structure of an ontology, constraints on ontology application can be derived. The ontology structure, its assumptions, axioms and constraints together yield the basic building blocks for the design of a wide array of IS systems and solutions.

ONTOLOGY CONSTRUCTION: THE CUE-N-ANCHOR GUIDED STRATEGY

A fundamental question that always arises when embarking on an ontology project is: How to build an ontology? Several researchers [e.g., Borst et al., 1997, Fernández et al., 1997, Holsapple and Joshi, 2002, Kishore et al., 2004b, Noy and McGuinnss, 2002, Uschold, 1996, Uschold and Gruninger, 1996, Uschold et al., 1998] proposed methodologies for ontology construction based mostly on their personal experiences. Although these approaches are influenced by their personal styles and preferences, certain common themes and directions emerge from a study of their recommended techniques. We synthesize these ideas into an evolving strategy for ontology development in this subsection.

At the outset, we set out with two caveats: (1) no ontology is complete and (2) no methodology is perfect. Consequently, the best one can do is to adopt an evolving strategy for ontology construction in a heuristic sense; the strategy may have to be refined, adjusted and even coursecorrected as the ontology begins to take shape. In this process, the ontology builder may need to revisit some of the earlier developments to refine and strengthen the ontology based on the persistent learning that occurs throughout the development process. This procedure is not a pre-specified and fully structured iterative process; clearly, such a prescription may not work with most ontology builders. Instead, we suggest a non-fully specified, semi-structured strategy of crisscrossing among the various developments that occur during the process. As a result, the proposed strategy is evolutionary, heuristic, but guided throughout. Hence, instead of presenting the proposed strategy as a sequence of steps, we introduce it as a set of guidelines which developers could use as they see appropriate.

Guideline 1: Define the area and scope of the ontology as best as you can.

To define the area and scope, we need a fairly clear understanding of the purposes for which the ontology is being built, the skills required in its development, and at least an approximate idea of the ontology constitution and its application. In this regard, a focus on the specific goals of the ontology is essential. While the criteria for evaluating an ontology suggested by Gruber [Gruber, 1995] are more appropriate while actually developing the ontology, we suggest the notion of informal competency questions proposed by Gruninger and Fox [1995] at this stage. Competency questions are essentially queries about the scenarios for which an ontology is designed and which the ontology should be able to answer. The idea behind these questions is to ensure that the ontology being developed is competent enough to respond to queries that may be posed to a future system that uses this ontology as a foundation. These questions can be used to guide the scope definition, such as what ontology is needed, what should be its level of detail, and will it serve our purpose and similar inquiries.

Guideline 2: Perform a baseline analysis.

The baseline analysis consists of two tasks: (1) brainstorming and (2) review of existing ontologies and relevant literature. Uschold and Gruninger [1996] suggest the use of sustained brainstorming sessions[3] to produce all relevant concepts and relationships, eliminating redundancies and ambiguities, and building a tentative structure of the ontology. Activities in the review task may be carried out either concurrently or in some sequence. In this task, other ontologies should be reviewed with the following questions in mind:

- How have they been constructed?
- How do they represent knowledge?
- How are they used?
- What construction approaches, representation structures and applications from the existing ontologies are relevant to our needs?
- In what manner can they be used for our needs? Do they help us in learning about the way (process) we can develop our ontology and associated systems? Or are they helpful in building the content (product) of the ontology and associated systems being developed? In other words, do they help us at the process level or the product level?

Process ➔	Knowledge capture	Knowledge representation	System design and development
Product ➔	Meta knowledge content	Specific knowledge	Application

[3] A group decision support system (GDSS) can be of great help in such a brainstorming session. For a discussion of GDSS and brainstorming, see, for example [Dennis et al., 1999].

Guideline 3: Anchor your ontology well and use cues to guide its development throughout.

An ontology must be well anchored. The anchor points could be domain-specific, context-specific or even literature-specific. Clearly, the brainstorming and review components could cause significant information overload on the developers. The magnitude of information that is both available and could be generated is vast. Therefore, we suggest the following strategy:

> 1. Identify a set of ideas as your anchors. These ideas could come from either brainstorming or the review. The proposed ontology should be adequately grounded in these anchors so that the development effort is both guided and protected from loss of direction.
>
> 2. Identify a set of ideas as your cues. Again these ideas could come from different sources. The cues are ancillaries that could be used to both enrich the ontology as well as guide the development.

The cue-n-anchor notion is quite important to the successful development of an ontology. In this context, we differentiate between the Push and the Pull approaches to ontology development. In the Push approach, all existential evidence tends to drive ontology development and the subsequent population of its knowledge bases; this is roughly the philosopher's approach to an ontology. In the Pull approach, the developer chooses the existential evidence that is appropriate and necessary for the goals of the ontology. This applied strategy and is more closely allied with an engineer's approach. The cue-n-anchor notion is central to the Pull approach and is important to the development of ontology-driven information systems.

Guideline 4: Develop a glossary of terms and refine the competency questions.

The glossary should enumerate the concepts, relationships, behaviors and even rudimentary structures if possible. The competency questions[4] assume more definitive shapes and are specified formally at this stage. These questions will later be used to ensure that the ontology is complete enough to serve the ultimate needs and if is free of any internal and external contradictions. The glossary and the refined and formalized competency questions constitute the baseline ontology document.

Guideline 5: Structure the baseline glossary into a specifiable ontology using a crisscross strategy.

An ontology specification could consist of (a) simple taxonomic structures, (b) specific data modeling structures such as object specifications, or (c) behavior models. A taxonomic structure usually serves as the backbone of an ontology and usually includes the basic and extended distinctions. Internal contradictions are not allowed. Standard relations such as *is-a, has-a, member-of* and many others could be used to structure these distinctions. Extended distinctions are then derived by overlapping the concept and relationship glossary as indicated in Guideline 4. A more formal specification of the ontology is obtained by incorporating class structures and their properties (such as slots and facets within the extended distinctions and by linking the distinctions to different behavior models. A detailed discussion about the class structures and properties is contained in the companion paper [Sharman et al., 2004]. Finally, the axioms and constraints are derived by applying logical and evidential reasoning. The emerging structure should then be tested for soundness using the constraints. By soundness we mean that the ontology is unambiguous and coherent with no internal contradictions[5]. This guideline constitutes the verification and validation steps in ontology construction.

---

[4] The term competency question is introduced in Guideline 1.

[5] Formally, "a deductive argument is *sound* if and only if it is both valid, and all of its premises are *actually true*. Otherwise, a deductive argument is *unsound.* Further, a deductive argument is said to be *valid* if and

Example. The above guideline is a major condensation of numerous steps involved in ontology construction. We next outline a crisscross strategy to weave among these steps in a coherent fashion using the following simple illustration. Consider building a taxonomy of concept distinctions. Assume that we start with a top-down approach where we begin with a highest-level concept, and develop its distinctions using the standard taxonomic relations. We call this the decompositional structure. At some point during this process, we may observe that (a) the decompositional structure is not yet complete, but (b) we are not able to proceed further due to lack of evidential support from below. At this point, we could suspend the top-down enumeration, and begin a bottom-up process by considering individual instances of concepts and grouping them into higher-level aggregates. We call this the compositional structure. At some point during this process, we may observe that: (a) the compositional structure is not yet complete, but (b) we are not able to proceed further due to lack of conceptual support from above. At this stage, we could suspend the bottom up process and resume top-down decompositional process that could be aided by compositional structure built so far. By crisscrossing in this manner between the two processes, the two structures should be combined with tests of verification and validation at the meeting points. We used this strategy quite successfully in our MibML ontology development project.

Guideline 6: Using the Cue-N-Anchor approach, decide on integrating existing ontologies with the one being built and evaluate formal representation mechanisms.

As an ontology is being built, existing ontologies may be found that could well be integrated with the current one in order to achieve the specified purposes. This integration could occur at two levels: (1) knowledge capture level and (2) knowledge level. At the knowledge capture level, the baseline ontology being constructed could benefit from well-understood knowledge capture mechanisms used in existing ontologies. These mechanisms could be used to extend, refine, and even simplify the current ontology. At the knowledge level, existing ontologies could yield representational concepts, tools and other artifacts to enrich and simplify the current ontology. Again, since the world of ontologies is vast, we recommend the Cue-N-Anchor approach. In a similar vein, a representation mechanism and an implementation tool for the ontology being built should be determined based on a survey of existing ontologies.

Guideline 7: Develop the formal representation of the ontology.

The formal specification should include:

1. the foundational conceptual model of the universe being modeled at appropriate levels of granularity, and

2. the full schema of the ontology describing the concept-relation structures, behavior models, assumptions, axioms and constraints, and proofs of bounded completeness and soundness of representation.

The formal representation should be evaluated using the fully specified formal competency questions (Guideline 4) and also Gruber's [1995] criteria for evaluating ontologies. This step could involve both logical and empirical evaluations of the ontology, identification of logical flaws, structural flaws, inadequacies in bounded completeness, and other application deficiencies. The correction of these deficiencies may take the designer to any of the earlier stages in ontology development described in these guidelines.

---

only if it takes a form that makes it impossible for the premises to be true and the conclusion nevertheless to be false. Otherwise, a deductive argument is said to be *invalid"* [Fieser and Dowden].

V. FUTURE RESEARCH DIRECTIONS

Research on computational ontologies, ontological engineering, and ontology-driven information systems are fertile areas of academic pursuit with practical implications. While a complete enumeration of all the research directions in these areas is daunting, we highlight some of the research themes we believe are important and provide guidelines in the following discussion.

DOMAIN ONTOLOGIES

For an ontology to be useful over the long haul, the issue of mapping an ontology to other parts of the system such as databases, user-interfaces, organizational processes needs to be addressed better [Ding and Foo, 2002]. Theoretical and empirical foundations for this kind of mapping need to be established[6]. We also need theoretical, empirical, and best practice papers that provide better guidance to the community, in integrating ontologies into existing information systems. Integration of top-level ontologies with domain level ontologies needs to be established on a much firmer footing.

Ontology needs to be developed in several domains so that information systems can be built upon them. One such area is semantic nets where we see significant development ongoing. Besides semantic nets, research on the ontological foundations of the following domains are highly promising: learning ontologies in application domains such as new product development, immersive learning environments, co-design of business and IT systems, web services, the emerging notion of the semantic web and other organizational process models in systems development. Furthermore, we envisage the arrival of a new field of enquiry – *onto-mining* – that deals with mining ontologies for knowledge discovery and application in several domains.

SPECIAL-PURPOSE REPRESENTATION VOCABULARIES FOR THE INFORMATION SYSTEMS DISCIPLINE

The creation of a complete ontology of a discourse universe (i.e., an extension consisting of ontological instances of the various concepts in the universe) is the ultimate goal from the perspective of knowledge codification and knowledge sharing in any universe. However, the first step towards achieving this goal is the creation of representation vocabularies that can be used to describe and model the ontological instances in an appropriate ontology. As discussed in Section III, conceptual modeling grammars provide representation vocabularies for the IS discipline. However, akin to general-purpose tools most of these grammars and vocabularies, such as DFDs, ER model, UML, and PetriNets are general-purpose vocabularies in the sense that they can be used to model any and all information systems. However, because these grammars are general-purpose vocabularies, they are not geared towards particular types of information systems and, thus, do not provide efficient tool sets for meeting the needs of these bounded universes. Recognizing this limitation of the general-purpose vocabularies, for some time researchers have been developing special-purpose representation vocabularies. Examples of such special-purpose grammars include the Smart Object Model for modeling of complex operations management systems [Vaishnavi et al., 1997], the SEAM model [Bajaj and Ram, 2002], the language/action perspective [Flores et al., 1988, Winograd, 1987-88], the metagraph approach [Basu and Blanning, 2000], and the XRL language [van der Aalst and Akhil, 2003] for modeling of workflow systems, and the IA framework [Pan and Tenenbaum, 1991], the ADEPT framework [Jennings et al., 2000], and the MibML conceptual modeling grammar [Kishore et al., 2004a, Zhang et al., 2003, Zhang et al., 2004] for modeling multiagent systems. While the IS discipline is making progress in grammars, clearly a need exists for more special-purpose vocabularies as the Web and the multiagent systems architectures become more prevalent and widespread.

---

[6] These foundations are discussed further later in this Section.

ONTOLOGIES AND LEARNING MODELS

Learning is central to the ontology development process. Ontologies evolve from a combination of domain knowledge capture, conceptual modeling of knowledge constructs, structural modeling of knowledge representation and inference mechanisms, and populating the knowledge bases. Learning is central in each of these components. Single-loop and double-loop learning mechanisms [Argyris, 1983] which are widely studied in the behavioral sciences literature can be adopted in developing each of these ontology components. The organizational learning literature focuses on four fundamental principles: (1) Organizational learning involves a tension between assimilating new learning (exploration) and using what has been learned (exploitation), (2) Organizational learning is multilevel: individual, group, and organization, (3) These three levels of organizational learning are linked by social and psychological processes: intuiting, interpreting, integrating, and instutionalizing (the 4I's), and (4) Cognition affects action (and vice versa) [Crossan, 1999]. The 4I's are related in feed-forward and feedback processes across the levels. Diagnostic systems are needed in enterprise learning processes to ensure that individuals are following institutionalized methods. Development of enterprise system-wide ontologies incorporating the underlying dynamic learning mechanisms would be valuable in these processes. Research is needed in ontological analysis of the feed-forward mechanisms (representing an innovation for instance) as they relate to the feedback mechanisms (representing what has been learned and adopted from the innovation). Since ontologies are known to increase the potential for knowledge reuse and sharing [Uschold and Gruninger, 1996], ontological bases for organizational learning would considerably improve communication between people with different needs, viewpoints and contexts, and enhance interoperability among organizational processes and systems. In this context, ontologies can be used in knowledge sharing in complex industrial applications [Borst et al., 1997]. Knowledge sharing across domains can be facilitated by using general and abstract ontological super theories such as mereology, topology, graph theory, and systems theory as building blocks underlying the learning and adoption processes involved. Some of these approaches are illustrated in Landes et al. [1999], Gruber [1993a], Motta et al. [2000], and Mulholland, et al. [2001]. Developing ontological approaches to supporting experiential learning processes in organizations is an important area for future research [Shipman and Marshall, 1999].

ONTOLOGY MINING

Mining ontologies for knowledge derivation and development is a new area of research. Mining the underlying ontology structures can renew or extend existing knowledge bases and eventually lead into unexplored territories. Ontology mining can be viewed in terms of two complementary components:

1. mining domain knowledge bases to construct or extend domain ontologies, and
2. mining domain ontologies to derive extended knowledge on the underlying domains.

Some of the major efforts in these research directions include the Gene Ontology (GO) mining project [Ashburner, 2000], ontology mining in text based systems [Maedche and Staab, 2000], grid based mining and knowledge discovery [Cannataro et al., 2003], data mining ontology for grid computing [Cannataro and Comito, 2003], mining web-based data ontologies for intelligent answering processes and multiagent systems [Li and Zhong, 2003], adapting the traditional data mining techniques to ontology mining [Wrobel et al., 2003], ontology-based multimedia data mining for design information retrieval [Simoff and Maheri, 2002], Intelligent Assistance for the Data Mining Process [Bernstein et al., 2002], and many more applications. Several applications of ontology mining appeared in emerging areas such as the semantic web, web services architectures, virtual community design, digital entertainment, and new product development processes in manufacturing systems. Developing new methods of ontology mining in application domains by deriving from and employing foundational theories, tools and techniques of ontology (Section II) is an important research direction as are innovative approaches to the ontology building process by extending the data mining techniques.

VI. CONCLUSION

This paper presented and discussed the foundations and definitions of computational ontologies, their creation and application in the IS domain, and key topics in ontological engineering. We provided a number of definitions and categorizations for ontologies available in the literature and reconciled them (Section II).

We discussed the distinction between ontologies and meta-ontologies. We also distinguished between ontologies of IS and ontologies for IS. We discussed the creation of ontologies of both these types because they are a fertile research area in computational ontology. Ontologies of IS include IS keyword lists, the ACM Computing Classification System, organizational knowledge management ontologies, the Bunge Wand Weber (BWW) ontology, the FRISCO ontology, and Alter's work systems and information systems ontologies. Ontologies for IS include IS modeling grammars, general-purpose computer programming languages, and special-purpose computer languages/formalisms. Although hundreds of ontologies for IS exist, only some of the more prominent ones in these areas are discussed briefly in the paper.

We discussed the application of computational ontologies in the IS discipline in two particular areas: evaluation of IS modeling grammars and ontology-driven information systems. We outlined some past research and current work being done in Section IV.

In Section V we discussed development of computational ontologies, or ontological engineering, including a discussion on the purposes computational ontologies serve, the ontology building skills, and the constitution of an ontology[7]. We synthesized ontology construction guidelines provided in the literature into a Cue-N-Anchor guided strategy for ontology construction and presented seven guidelines that emerge from this synthesis.

The paper concludes with a number of future research directions, including domain ontologies, special-purpose representation vocabularies for the IS discipline, ontologies and learning models, and ontology mining.

A companion paper, Volume 14, Article [Sharman et al., 2004] discusses the technical aspects of ontologies. It provides a comprehensive review of the formalisms, languages, and tools used for specifying and implementing computational ontologies.

Editor's Note: This article was fully peer reviewed. It was received on December 5, 2003. It was with the authors three and a half months for 2 revisions during which the original submission was divided into two parts. The article was published on August 16, 2004.

REFERENCES

Editor's Note: The following reference list contains the address of World Wide Web pages. Readers who have the ability to access the Web directly from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that

1. these links existed as of the date of publication but are not guaranteed to be working thereafter.

2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.

---

[7] This topic is essentially about representation and implementation of ontologies and is discussed in a more elaborate manner in the companion paper [Sharman et al., 2004].

3. the authors of the Web pages, not CAIS, are responsible for the accuracy of their content.

4. the author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

ACM "Computing Classification System," Association for Computing Machinery, http://www.acm.org/class/ (Consulted November 12, 2003).

Alter, S. (1999) "Are the fundamental concepts of information systems mostly about work systems?" *Communications of the Association for Information Systems*, (5) 11, pp. 1-68.

Argyris, C. (1983) "Action science and intervention", *Journal of Applied Behavioural Science*, (19), pp. 115-140.

Armour, P. G. (2000) "The case for a new business model: Is software a product or a medium?" *Communications of the ACM*, (43) 8, pp. 19-22.

Ashburner, M. e. a. (2000) "Gene Ontology: tool for the unification of biology", *Nature Genetics*, (25) 1, pp. 25-29.

Bajaj, A. and S. Ram (2002) "SEAM: A State-Entity-Activity-Model for a Well-Defined Workflow Development Methodology", *IEEE Transactions on Knowledge and Data Engineering*, (14) 2, pp. 415-431.

Barki, H., S. Rivard, and J. Talbot (1988) "An information systems keyword classification scheme", *MIS Quarterly*, (1988) June, pp. 299-322.

Barki, H., S. Rivard, and J. Talbot (1993) "A keyword classification scheme for IS research literature: An update", *MIS Quarterly*, (1993) June, pp. 209-226.

Basu, A. and R. W. Blanning (2000) "A formal approach to workflow analysis", *Information Systems Research*, (11) 1, pp. 17-36.

Bateman, J. A., B. Magini, and F. Rinaldi. (1994) "The Generalized Upper Model." *1994 European Conference on Artificial Intelligence (ECAI '94), Workshop on Implemented Ontologies,* pp. 34-45.

Bergman, M. and S. Paavola "The Commens Dictionary of Peirce's Terms: Peirce's Terminology in His Own Words," http://www.helsinki.fi/science/commens/dictionary.html (Consulted April 24, 2003).

Bernstein, A., S. Hill, and F. Provost. (2002) *Intelligent assistance for the data mining process: an ontology-based approach*. Stern School of Business, New York University, New York, NY, IS-02-02.

Borst, P., J. M. Akkermans, and J. Top (1997) "Engineering ontologies", *International Journal of Human-Computer Studies*, (46) 2/3, pp. 365-406.

Bunge, M. A. (1977) *Ontology I: The Furniture of the World*. Vol. 3. Boston, MA: D. Reidel Publishing Company.

Bunge, M. A. (1979) *Ontology II: A World of Systems*. Vol. 4. Boston, MA: D. Reidel Publishing Company.

Cannataro, M. and C. Comito (2003) "A data mining ontology for grid programming," University of Calabria, 2003).

Cannataro, M., A. Conguista, C. Mastroianni, A. Pugliese et al. (2003) "Grid-based data mining and knowledge discovery," in N. Zhong and J. Liu (Eds.) *Handbook of Intelligent Information Technology*. Vol. 90: IOS Press, pp.

Chandrasekaran, B., J. R. Josephson, and V. R. Benjamins (1999) "What are ontologies, and why do we need them?" *IEEE Intelligent Systems*, (14) 1, pp. 20-26.

Crossan, M. W. e. a. (1999) "An organizational learning framework: from tuition to institution", *Academy of Management Review*, (24) 3, pp. 522-537.

Degen, W., B. Heller, H. Herre, and B. Smith. (2001) "GOL: towards an axiomated upper-level ontology." *International Conference on Formal Ontology in Information Systems (FOIS '01), Ogunquit, Maine,* pp. 34-46.

Dennis, A. R., J. E. Aronson, W. G. Heninger, and E. D. Walker, II (1999) "Structuring time and task in electronic brainstroming", *MIS Quarterly*, (23) 1, pp. 95-108.

Ding, Y. and S. Foo (2002) "Ontology Research and Development: Part 1", *Journal of Information Science*, (28) 2.

Dretske, F. (1981) *Knowledge and the Flow of Information*. Cambridge, MA: MIT Press.

Eriksson, H. and M. Penker (2000) *Business Modeling with UML: Business Patterns at Work*. New York: Wiley.

Evermann, J. and Y. Wand. (2001) "Toawards ontologically based semantics for UML constructs." *the 20th International Conference on Conceptual Modeling, Yokohama, Japan*.

Falkenberg, E. D., W. Hesse, P. Lindgreen, B. E. Nilsson et al. (1998) *A framework of information system concepts: the FRISCO report*. Vol. 2003: International Federation of Information Processing (IFIP).

Fernández, M., A. Gómez-Pérez, and N. Juristo. (1997) "METHONTOLOGY: from ontological art towards ontological engineering." *Workshop on Ontological Engineering: AAAI-97 Spring Symposium Series, Stanford, USA*.

Fettke, P. and P. Loos. (2003) "Ontological evaluation of reference models using the Bunge-Wand-Weber model." *Ninth Americas Conference on Information Systems, Tampa, FL,* pp. 2944-2955.

Fieser, J. and B. Dowden "The Internet Encyclopedia of Philosophy," http://www.iep.utm.edu/ (Consulted August 3, 2004).

Flores, F., M. Graves, B. Hartfield, and T. Winograd (1988) "Computer systems and the design of organizational interactions", *ACM Transactions on Office Information Systems*, (6) 2, pp. 153-172.

Genesereth, M. R. and N. J. Nilsson (1987) *Logical Foundation of Artifical Intelligence*. Los Altos, California: Morgan Kaufmann.

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

Gruber, T. (1993a) "A translational approach to portable ontologies", *Knowledge Acquisition*, (5) 2, pp. 199-220.

Gruber, T. R. (1993b) "Toward principles for the design of ontologies used for knowledge sharing," in N. Guarino and R. Poli (Eds.) *Formal Ontology in Conceptual Analysis and Knowledge Representation*: Kluwer Academic Publishers, pp.

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.

Gruninger, M. and M. S. Fox. (1995) "Methodology for the design and evaluation of ontologies." *Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal, Canada*.

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

Holsapple, C. W. and K. D. Joshi (2001) "Organizational knowledge resources", *Decision Support Systems*, (31) 1, pp. 39-54.

Holsapple, C. W. and K. D. Joshi (2002) "A collaborative approach to ontology design", *Communications of the ACM*, (45) 2, pp. 42-47.

IEEE (2003) "P1600.1 Standard Upper Ontology (SUO) Working Group," IEEE, http://suo.ieee.org/ (Consulted November 14, 2003, 2003).

Jasper, R. a. M. U. (1999) "A Framework for Understanding and Classifying Ontology Applications." *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden*.

Jennings, N. R., T. J. Norman, P. Faratin, P. O'Brien et al. (2000) "Autonomous agents for business process management", *Journal of Applied Artificial Intelligence*, (14) 2, pp. 145--189.

Kishore, R., R. Sharman, H. Zhang, and R. Ramesh. (2004a) *MibML: a conceptual modeling grammar for integrative business information systems using the agent metaphor*. State University of New York at Buffalo, Buffalo, NY, Working Paper #801.

Kishore, R., H. Zhang, and R. Ramesh (2004b) "A Helix-Spindle model for ontological engineering", *Communications of the ACM*, (47) 2, pp. 69-75.

Landes, D., K. Schneider, and F. Houdek (1999) "Organizational learning and experience documentation in industrial software projects", *International Journal of Human-Computer Studies*, (51) 3, pp. 643-661.

Lenat, D. B. and R. V. Guha (1990) *Building Large Knowledge-Based Systems: Representation and Inference in the CYC project*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.

Li, Y. and N. Zhong (2003) "Web Mining Model and its Applications for Information Gathering", *forthcoming in Knowledge-Based Systems, special issue on Web Intelligence*.

Maedche, A. and S. Staab. (2000) "Mining ontologies from text." *Proceedings of 12th International Workshop on Knowledge Engineering and Knowledge Management (EKAW 2000), Juan-Les-Pins, France*.

Miller, G. A. (1990) "Wordnet: an online lexical database", *International Journal of Lexicography*, (3) 4, pp. 235-312.

Milton, S. K. and E. Kazmierczak (2003/4) "An ontology of data modelling languages: a study using a common-sense realistic ontology", *Journal of Database Management*, Special Issue on Ontological Analysis, Evaluation, and Engineering Of Business Systems Analysis Methods, pp. 18 pages.

Milton, S. K., E. Kazmierczak, and C. D. Keen (2002) "On the study of data modelling languages using Chisholm's ontology," in H. Kangassalo, W. T., H. Jaakkola, I. Rozman et al. (Eds.) *Information Modelling and Knowledge Bases XIII*, Amsterdam: IOS Press, pp. 19-36.

Motta, E., S. Buckingham Shum, and J. Domingue (2000) "Ontology-driven document enrichment: principles, tools and applications", *International Journal of Human-Computer Studies*, (52) 6, pp. 1071-1109.

Mullholland, P., Z. Zdrahal, J. Domingue, M. Hatala et al. (2001) "A methodological approach to supporting organizational learning." *International Journal of Human-Computer Studies*, (55) 3, pp. 337-367.

Nonaka, I. (1994) "A Dynamic Theory of Organizational Knowledge Creation", *Organization Science*, (5) 1, pp. 14-37.

Noy, N. F. and D. L. McGuinnss. (2002) *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA, Stanford Medical Informatics Report SMI-2001-0880.

Pan, J. Y. C. and J. M. Tenenbaum (1991) "An Intelligent Agent Framework for Enterprise Integration", *IEEE Transactions on Systems, man, and cybernetics*, (21) 6, pp. 1391-1991.

Parsons, J. and Y. Wand (1997) "Using objects in system analysis", *Communications of the ACM*, (40) 12, pp. 104-110.

Paton, N. W., R. Stevens, P. Baker, C. A. Goble et al. (1999) "Query Processing in the TAMBIS Bioinformatics Source Integration System." *Proceedings of the 11th International Conference on Scientific and Statistical Database Management, New York,* pp. 138-147.

Peirce, C. S. "The Principles of Logical Graphics (MS 493)," http://www.helsinki.fi/science/commens/dictionary.html (Consulted April 24, 2003).

Peirce, C. S. (1893) "Extension of the Aristotelian syllogistic", *Grand Logic*, (2), pp. 517-518.

Princeton University (1997) "WordNet ® 1.6," Princeton University, ftp://clarity.princeton.edu/pub/wordnet/wn1.6unix.tar.gz (Consulted November 14, 2003).

Robin, R. S. (1967) "Annotated Catalogue of the Papers of Charles S. Peirce,", Amherst, MA: The University Of Massachusetts Press http://www.iupui.edu/~peirce/robin/robin.htm (Consulted November 14, 2003).

Scheer, A.-W. (1999) *ARIS: Business Process Frameworks*, 2nd edition. New York: Springer.

Sharman, R., R. Kishore, and R. Ramesh (2004) "Computational Ontologies and Information Systems: II. Formal Specification", *Communications of the AIS*, (14) 184-205

Shipman, F. M., III and C. C. Marshall (1999) "Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems." *Computer Supported Cooperative Work*, (8) 4, pp. 333-352.

Simoff, S. J. and M. L. Maheri (2002) "Ontology-based Multimedia Data Mining for Design Information," Key Centre of Design Computing, University of Sydney, 2002).

Smith, B. (2003) *Ontology and information systems*. SUNY at Buffalo, Buffalo, NY.

Sowa, J. F. (1984) *Conceptual Structures: Information Processing in Mind and Machine*. New York: Addison-Wesley.

Sowa, J. F. (1999) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks/Cole.

Storey, V. C., Goldstein, R.C., and Ullrich, H. (2002) "Naive Semantics to Support Automated Database Design", *IEEE Transactions on Knowledge and Data Engineering*, (14) 1, pp. 1-12.

Studer, R., V. R. Benjamins, and D. Fensel (1998) "Knowledge Engineering: Principles and methods", *Data & Knowledge Engineering*, (25), pp. 161-197.

Studer, R., H. Eriksson, S. W. T. Gennari, D. Fensel et al. (1996) "Ontologies and the configuration of problem-solving methods." *10th Knowledge Acquisition for Knowledge-based Systems Workshop, Banff, Canada*.

Takagaki, K. and Y. Wand (1991) "An object-oriented information systems model based on ontology," in F. V. Assche, B. Moulin, and C. Rolland (Eds.) *Object Oriented Approach in Information Systems*, North, Holland: Elsevier Science, pp. 275-296.

Tarski, A. (1982) "The concept of truth in formalized languages," in *Logic, Semantics, Metamathematics*, 2nd edition, Indianapolis, IN: Hackett Publishing Company, pp.

Uschold, M. (1996) "Building ontologies: towards a unified methodology." *16th Annual Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, UK*.

Uschold, M. and M. Gruninger (1996) "Ontologies: principles, methods, and applications", *Knowledge Engineering Review*, (11) 2, pp. 93-155.

Uschold, M., M. King, S. Moralee, and Y. Zorgios (1998) "The Enterprise ontology", *The Knowledge Engineering Review*, (13) 1, pp. 31-89.

Vaishnavi, V. K., G. C. Buchanan, and W. L. Kuechler, Jr. (1997) "A data/knowledge paradigm for the modeling and design of operations support systems", *IEEE Transactions on Knowledge and Data Engineering*, (9) 2, pp. 275-291.

van der Aalst, W. M. P. and K. Akhil (2003) "XML-based schema definition for support of interorganizational workflow", *Information Systems Research*, (14) 1, pp. 23-46.

Van Heijst, G., A. T. Schreiber, and B. J. Wielinga (1997) "Using explicit ontologies in KBS development", *International Journal of Human-Computer Studies*, (46) 2/3, pp. 183-292.

Wand, Y., V. C. Storey, and R. Weber (2000) "An ontological analysis of the relationship construct in conceptual modeling", *ACM Transactions on Database Systems*, (24) 4, pp. 494-528.

Wand, Y. and R. Weber (1989) "An ontological evaluation of systems analysis and design methods," in E. D. Falkenberg and P. Lindgreen (Eds.) *Information System Concepts: An In-depth Analysis*, Amsterdam: North-Holland, pp. 79-107.

Wand, Y. and R. Weber (1990) "An ontological model of an information system", *IEEE Transactions on Software Engineering*, (16) 11, pp. 1282-1292.

Wand, Y. and R. Weber (1993) "On the ontological expressiveness of information systems analysis and design grammars", *Journal of Information Systems*, (3) 4, pp. 217-237.

Wand, Y. and R. Weber (2002) "Research commentary: information systems and conceptual modeling - a research agenda", *Information Systems Research*, (13) 4, pp. 363-376.

Weber, R. and Y. Zhang (1996) "An ontological evaluation NIAM's grammar for conceptual schema diagrams", *European Journal of Information Systems*, (6) 2, pp. 147-170.

Winograd, T. (1987-88) "A language/action perspective on the design of cooperative work", *Journal of Human-Computer Interaction*, (3) 1, pp. 3-30.

Wrobel, A., O. Wuermli, J. Joller, and S. C. Hui (2003) "Data Mining for Ontology Building", *forthcoming in IEEE Intelligent Systems*.

Zhang, H., R. Kishore, R. Ramesh, and R. Sharman. (2003) "The GRITICKA ontology for modeling multiagent-based integrative business information systems." *2003 Americas Conference on Information Systems (AMCIS 2003), Tampa, FL,* pp. 1885-1891.

Zhang, H., R. Kishore, R. Sharman, and R. Ramesh. (2004) "The GRITIKA ontology for modeling e-service applications: formal specification and illustration." *37th Hawaii International Conference on System Sciences (HICSS-37), Big Island, Hawaii*.

APPENDIX I. SOME COMMON TOP-LEVEL[8] ONTOLOGIES

Bunge's Ontology [Bunge, 1977]: Bunge's ontology, termed "The Furniture of the World," is a formal axiomated top-level ontology. The basic notion in this ontology is that the world is made up of things that possess properties. The ontology provides for a number of other concepts including states, classes, possibilities real and conceptual, events, processes, actions and reactions, space, duration, and space time that are linked together in a complete ontology through postulates and axioms.

CYC [Lenat and Guha, 1990]: CYC was developed as a general top-level ontology for commonsense knowledge to facilitate reasoning in the AI field. The root of this ontology is also called a thing but thing in this ontology does not possess properties of its own. Thing is divided into individual objects, intangibles, and represented things. CYC contains more than 10,000 concept types used in the rules and facts encoded in the knowledge base. All concepts in the CYC ontology are connected to each other through axioms and rules.

WordNet [Miller, 1990]: Wordnet is one of the most well-developed lexical ontologies and provides an online lexical reference system. Version v2 of WordNet is used by some dictionaries on the web (e.g., www.dictionary.com). WordNet distinguishes between nouns, verbs, adjectives, and adverbs and organizes lexical objects (categories) in the ontology semantically in these top-level distinctions (categories). WordNet also organizes lexical objects in *synnets* which are essentially networks of synonyms. There are a total of 70,000 synnets in WordNet.

Generalized Upper Model [Bateman et al., 1994]: Generalized Upper Model is also a top-level ontology that was designed in the AI field to support natural language processing in English, German, and Italian. The top-level concept in this ontology is called an *um-thing* which is divided into three categories: configuration, element, and sequence. This ontology is purely a taxonomic ontology and does not connect the concepts in the taxonomy through axioms and rules. It was developed with the assumption that all axiomatic information about linguistic concepts will be encoded in the natural language processing programs.

Sowa's Upper-Level Ontology [Sowa, 1999]: Sowa's ontology is an upper-level ontology that is strongly rooted in philosophy. Sowa synthesizes the ontologies and ontological notions proposed by Heraclitus, Peirce, and Whitehead into an ontology at the root of which is the symbol T for universal type. T is split along three distinctions or dimensions. The first dimension divides T into physical and abstract categories which are further divided into three sub-categories each based on Peirce's distinction of firstness, secondness, and thirdness. T is also divided along another dimension into three other categories, namely, independent, relative, and mediating. Finally, it is also divided into continuants and occurrents. Thus, the entire ontology is represented as a lattice of 12 categories that are derived by forming combinations of the three basic distinctions.

General Ontological Language (GOL) [Degen et al., 2001]: GOL is an upper-level ontology that was developed to overcome limitations of the ontology of sets. In GOL, the entities of the real world are either *sets* or *urelements*. Urelements are further divided into *individuals* and *universals*. Individuals belong to the realm of concrete entities, which means that they exist within the confines of space and time. Universals are entities that can be instantiated simultaneously by a multiplicity of different individuals that are similar in given respects. Individuals are further classified into *substances*, *moments*, *chronoids*, *topoids*, and *situoids*. A substance is that which can exist by itself. A moment is an entity which can exist only in another entity. A situoid is a part of the world that can be comprehended as a coherent whole and does not need other entities in order to exist. *Situations* are situoids at a time to represent a snap-shot view of some part of the world. Chronoids and topoids are instances of the universals *Time* and *Space*, respectively. Relations are entities which glue together the things of the real world. GOL is a fully-axiomated top-level ontology.

[8] Top level ontologies are also known as upper level ontologies.

IEEE Standard Upper Ontology (SUO) [IEEE, 2003]: SUO is one of the latest ontology projects. It was undertaken by the IEEE Standard Upper Ontology Working Group (SUO WG) to specify an upper-level ontology that will support computer applications such as data interoperability, information search and retrieval, automated inferencing, and natural language processing. This ontology is currently under development. IEEE SUO WG defined an information flow framework that comprises three metalevels – top, upper, and lower – at which upper-level ontologies exist. SUO project is an attempt to reconcile a number of existing upper level ontologies including the Knowledge Interchange Format [Genesereth, 1992] and to create a comprehensive and fully-axiomated upper-level ontology that contains between 1000 and 2000 terms which provide a common structure for low-level domain ontologies of much larger size and more specific scope.

ABOUT THE AUTHORS

Rajiv Kishore is assistant professor in the School of Management at the State University of New York at Buffalo. His research interests are in IT outsourcing, adoption of IT innovations, and conceptual modeling. His papers are published or accepted for publication in *Communications of the ACM, Decision Support Systems, Information Systems Frontiers, Journal of Healthcare Information Management, and Journal of Management Information Systems.* Rajiv presented his research at ICIS, HICSS, AMCIS, SIM, and other international conferences and workshops, and received a best paper award at AMCIS 2001. He is the recipient of a multi-year National Science Foundation research grant as a co-principal investigator in the area of IT outsourcing. Rajiv is also a founding board member of AIS SIG on Ontology-Driven Information Systems (SIG-ODIS). He is currently guest-editing a special issue of the *Journal of AIS* on Ontologies in the Context of Information Systems.

Raj Sharman is assistant professor in the School of Management at State University of New York at Buffalo. He received his Bachelors degree in Engineering and Masters Degree in Management from the Indian Institute of Technology, Bombay, India. He also received a Masters in Industrial Engineering, and a Doctorate in Computer Science from Louisiana State University. His research interests are in Conceptual Modeling and Knowledge Representation, Information Security, Web Caching and Medical Informatics. He is the recipient of several grants, both internal and external, to support his research. His publications appear in refereed journals and conferences in both the Information Systems and the Computer Science disciplines. Prior to his appointment at The State University of New York at Buffalo, Raj Sharman was active in the technology sector both as an entrepreneur and as a developer of Enterprise Integration Software.

Ram Ramesh is professor in the School of Management at the State University of New York at Buffalo. His research streams include conceptual modeling (ontologies, connectionist modeling and nonmonotonic reasoning, economics and technologies of internet capacity provision networks (CPN), and database systems and distributed computing frameworks. He is a founding board member of AIS SIG on Ontology-driven Information Systems (SIG-ODIS) and a board member of AIS SIG on Semantic Web and Information Systems (SIG-SEMIS). He serves as an Associate Editor for *INFORMS Journal on Computing, Communications of the AIS* and several more. He is co-Editor-in-Chief of *Information Systems Frontiers* and edited volumes in *Annals of OR* and *CACM.* He is currently guest-editing an issue of the *Journal of AIS* on Ontologies in the context of Information Systems. He publishes extensively in journals such as *INFORMS Journal on Computing, Information Systems Research, IEEE/TKDE, ACM/TODS* and *CACM* among others

Copyright © 2004 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints or via e-mail from ais@gsu.edu

Communications of the Association for Information Systems

ISSN: 1529-3181

EDITOR-IN-CHIEF
Paul Gray
Claremont Graduate University

AIS SENIOR EDITORIAL BOARD

Detmar Straub Vice President Publications Georgia State University	Paul Gray Editor, CAIS Claremont Graduate University	Sirkka Jarvenpaa Editor, JAIS University of Texas at Austin
Edward A. Stohr Editor-at-Large Stevens Inst. of Technology	Blake Ives Editor, Electronic Publications University of Houston	Reagan Ramsower Editor, ISWorld Net Baylor University

CAIS ADVISORY BOARD

Gordon Davis University of Minnesota	Ken Kraemer Univ. of Calif. at Irvine	M.Lynne Markus Bentley College	Richard Mason Southern Methodist Univ.
Jay Nunamaker University of Arizona	Henk Sol Delft University	Ralph Sprague University of Hawaii	Hugh J. Watson University of Georgia

CAIS SENIOR EDITORS

Steve Alter U. of San Francisco	Chris Holland Manchester Bus. School	Jaak Jurison Fordham University	Jerry Luftman Stevens Inst.of Technology

CAIS EDITORIAL BOARD

Tung Bui University of Hawaii	Fred Davis U.ofArkansas, Fayetteville	Candace Deans University of Richmond	Donna Dufner U.of Nebraska -Omaha
Omar El Sawy Univ. of Southern Calif.	Ali Farhoomand University of Hong Kong	Jane Fedorowicz Bentley College	Brent Gallupe Queens University
Robert L. Glass Computing Trends	Sy Goodman Ga. Inst. of Technology	Joze Gricar University of Maribor	Ake Gronlund University of Umea,
Ruth Guthrie California State Univ.	Alan Hevner Univ. of South Florida	Juhani Iivari Univ. of Oulu	Munir Mandviwalla Temple University
Sal March Vanderbilt University	Don McCubbrey University of Denver	Emannuel Monod University of Nantes	John Mooney Pepperdine University
Michael Myers University of Auckland	Seev Neumann Tel Aviv University	Dan Power University of No. Iowa	Ram Ramesh SUNY-Bufallo
Maung Sein Agder University College,	Carol Saunders Univ. of Central Florida	Peter Seddon University of Melbourne	Thompson Teo National U. of Singapore
Doug Vogel City Univ. of Hong Kong	Rolf Wigand U. of Arkansas, Little Rock	Upkar Varshney Georgia State Univ.	Vance Wilson U.Wisconsin,Milwaukee
Peter Wolcott Univ. of Nebraska-Omaha			

DEPARTMENTS

Global Diffusion of the Internet. Editors: Peter Wolcott and Sy Goodman	Information Technology and Systems. Editors: Alan Hevner and Sal March
Papers in French Editor: Emmanuel Monod	Information Systems and Healthcare Editor: Vance Wilson

ADMINISTRATIVE PERSONNEL

Eph McLean AIS, Executive Director Georgia State University	Samantha Spears Subscriptions Manager Georgia State University	Reagan Ramsower Publisher, CAIS Baylor University