

December 2003

# Computational Ontologies: Foundations, Representations, and Methods

Rajiv Kishore

*State University of New York, Buffalo*

Ram Ramesh

*State University of New York, Buffalo*

Raj Sharman

*State University of New York, Buffalo*

Follow this and additional works at: <http://aisel.aisnet.org/amcis2003>

## Recommended Citation

Kishore, Rajiv; Ramesh, Ram; and Sharman, Raj, "Computational Ontologies: Foundations, Representations, and Methods" (2003).  
*AMCIS 2003 Proceedings*. 422.

<http://aisel.aisnet.org/amcis2003/422>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2003 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# COMPUTATIONAL ONTOLOGIES: FOUNDATIONS, REPRESENTATIONS, AND METHODS<sup>1</sup>

**Rajiv Kishore**

School of Management  
State University of New York, Buffalo  
[rkishore@buffalo.edu](mailto:rkishore@buffalo.edu)

**R. Ramesh**

School of Management  
State University of New York, Buffalo  
[rramesh@buffalo.edu](mailto:rramesh@buffalo.edu)

**Raj Sharman**

School of Management  
State University of New York, Buffalo  
[rsharman@buffalo.edu](mailto:rsharman@buffalo.edu)

## Abstract

*In this tutorial, we address some of the fundamental concepts and design issues in ontology-driven information systems. We present the essential formalisms, computer languages and implementation issues with a comparison of selected ontologies. We conclude with an overview of the emerging discipline of ontological engineering. Specific guidelines for designing ontologies directions for future research are also presented.*

## Introduction

Ontology as “the metaphysical study of the nature of being and existence” (WordNet 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 the field of philosophy, it is also a subject of inquiry in computation-oriented disciplines (e.g., artificial intelligence, information science, database management,). Domain-specific ontologies were developed in several disciplines, including chemistry, enterprise management, geography, linguistics, medicine, and sociology. As a result, while philosophy still treats ontology in the singular because it deals with the nature of all reality, other disciplines view it only in the limited context of domain-specific reality. Hence, now there are multiple ontologies in disciplines other than philosophy, and each ontology pertains to the reality limited to a particular domain or discipline. The goal of such ontologies 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 applied in practical settings.

We propose that ontology should become a core subject of inquiry in the information systems (IS) field. This is because of a high degree of affinity between the two fields and as a result ontology-driven information systems are emerging (Guarino 1998; Kishore et al. Forthcoming). This tutorial is, therefore, a modest attempt to introduce the IS reader to the basic notions about ontologies. The rest of the paper is organized as follows: First we outline and discuss some fundamental concepts pertaining to computational ontologies. Next we discuss formalisms, languages, and tools used for representing IS ontologies. Then we outline an approach with guidelines for creating computational ontologies. We conclude with future research directions for IS ontologies.

---

<sup>1</sup>Authors are listed alphabetically.

## Computational Ontologies

We use the term computational ontology to denote all domain-specific ontologies in the applied disciplines. While the purpose of a philosophical ontology is to know, a computational ontology is intended for application. As a result, a philosophical ontology addresses an unbounded universe of discourse (Pierce Manuscripts; Robin 1970), and the universe is bounded in a computational ontology. Our notion of a bounded universe is also very similar to Sowa's notion of a micro-world (Sowa 2000a, p. 52).

We adopt the popular definition for computational ontologies provided by Gruber (1993) which states: "An ontology is a formal explicit specification of a shared conceptualization." 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 (Chandrasekaran et al. 1999). This view is essentially a distinction between intension and extension (Sowa 1984, p. 11) about the universe of discourse. The representation vocabulary provides symbols for the concepts in the universe, thus being the intension for the universe, while the knowledge base is essentially the set of all referents to which a concept may refer to, thus being the extension for that concept. For example, in the Entity-relationship (ER) model, the notions of entity and relationships are intensions; whereas specific classes of entities and relationships are extensions. In the ontological context, the extension does not include the specific instances, and each member of the extension set is essentially a class of actual instances. For example, in 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 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. Top-level ontologies start with very general top-level concepts such as a Thing and provide a taxonomy of top-level concepts. Popular top-level ontologies are CYC (Lenat and Guha 1990), WordNet (Miller 1995) Sowa's ontology (Sowa 2000a), and GUM (Bateman et al. 1995). While these ontologies pertain to an unbounded universe of discourse, they are implemented computationally. Lower-level ontologies pertain to bounded universes of discourse and are termed variously as application, domain, and task ontologies in the literature.

The AI literature generally distinguishes 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 "what" knowledge, while others define the notion of method and task ontologies to capture procedural knowledge about a domain. Another distinction made in the literature is between terminological and axiomatic ontologies (Sowa 2000a). Categories in a terminological ontology need not be specified fully by axioms and definitions but can simply be a collection of categories and terms (e.g., the WordNet ontology). On the other hand, in axiomatic ontologies, categories are distinguished by axioms and definitions stated in different logical forms.

Finally, it is quite important to distinguish ontologies in terms of the different levels of representation language that are used to represent knowledge about a particular universe of discourse. Tarski (1935) developed a theory of stratified metalevels to distinguish between languages and meta-languages and what they can and cannot refer to. For example, any of the three languages—English, UML, or predicate logic—can be used as a metalanguage for representing the symbols in a domain of discourse to create an ontology. However, it should be noted that all the three meta-languages themselves refer to their respective universes of discourse and each of them, therefore, represent an ontology in their own right. However, as was noted, uninterpreted logic languages such as predicate logic, conceptual graphs, or Knowledge Interchange Format (KIF) are ontologically neutral 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 2000a, p. 492).

## Representation

This section addresses the issues surrounding the choice of a representation formalism and an implementation language in developing ontologies. For this purpose, we compare some of the popular formalisms and languages that provide excellent directions in ontology development.

### *Issues and Expectations from Formalisms and Languages*

We restrict our discussion to our needs of representing ontologies to make them usable from a computer-based information systems point of view. Regardless of their levels, ontologies in general contain: (a) Concepts, (b) Taxonomies of concepts, (c) Relationships and functions, (d) Axioms, (e) Attributes - Global, conceptual and instance level, (f) Facets - Constraints, Default

slots, and (g) Instances. Most ontologies also support some form of formal semantics and reasoning, besides their mechanisms of basic knowledge representation. Several languages were developed for this purpose and all of them support some or all of the above constructs to different degrees. At a minimum, a knowledge representation mechanism should provide both syntax and logic support. While the syntax is concerned with how knowledge is stored, the logical component deals with its inferential capabilities (Reichgelt 1991). We address these requirements at the implementation, logical, epistemological, and conceptual levels below.

At the implementation level, we are primarily concerned with tractability of the representation mechanism. These mechanisms relate to the ability of the representation language to aid the creation of computer information systems. Some examples of concerns at this level relate to how well the language supports inferencing, indexing, support for a large set of concepts, and relationships in an ontology. At the logical level, the expressive power of the language is the primary concern. This idea refers to the ability to represent logical properties unambiguously and with clarity from both syntactic and inferential points of view. Some examples of these concerns are: (a) can we represent equivalence between concepts or instances? (b) does an 'is-a' relationship between two instances  $x$  and  $y$  imply that every  $x$  is a  $y$  or that some  $x$ 's are  $y$ 's? At the epistemological level, the main concern is with the types of primitive expressions and the types of inference strategies used. For example, these concerns translate to the following questions in a medical ontology: (a) does the formalism support an inferencing strategy to help an expert physician to diagnose a physical ailment, and (b) does the formalism also support a strategy for non-physicians to learn more about the ailment. However, we do not make any decision about which actual primitives and inference strategies are used to represent knowledge about some domain at this stage. At the conceptual level, the actual primitives that are part of the knowledge representation formalism are of concern. Examples of such concerns are whether there is an 'is-a' arc to support inheritance, or whether there is a 'part-of' arc to represent composition in the formalism.

### ***Requirements of Knowledge Representation Languages***

Several criteria can be used to assess the value of a formalism. However, the most important criterion one must consider is adequacy of the language at the implementation, logical, epistemological and conceptual levels. This criterion includes qualifiers such as expressiveness, naturalness, etc. At the implementation level, a language should provide efficient storage, quick inferencing capabilities and consistent encoding of the ontology constructs throughout.

At the logical level, a representation language should allow for precise specification and interpretation of well-formed expressions (as in model-theory). More specifically, this idea deals with the expressive power in terms of flexibility, explicitness, accuracy, and formality. These criteria imply that the meaning of complex expressions should be derivable from simpler expressions and the ability to create sound inference procedures. Note that soundness ensures that statements do not contradict each other. Furthermore, it is important to recognize the trade-offs between expressive power and complexity.

At the epistemological level, the representation language should allow for representations to be constructed or organized in ways that are most natural to the domain. The language should allow for representations to be modular so that changes and evolutions in the domain can be managed by minimal changes to the ontology. At the epistemological level the language should provide flexibility in terms of the granularity of information as well as support to the primitives at the conceptual level. The granularity dictates the chunks of knowledge that form the building blocks for organizing the knowledge.

At the conceptual level, the language or chosen representation should provide the modeler the ability to represent real world concepts, relationships, constraints and axioms in a concise and precise manner (expressiveness).

## **Comparative Analysis of Formalisms**

This section provides a brief overview of the following formalisms: (a) Informal and semi-formal representations, (b) Logic-based languages (First order predicate logic, second-order predicate logic), (c) Production Rules, (d) Semantic Nets (SNePS, Conceptual graphs, KL-One), (e) Frame-based languages, (f) Description Logics, and (g) Hybrid Representations (KL-TWO, KRYPTON).

An Informal ontology is one where the types are either not defined or defined in some natural language (Sowa 1991). There are neither rules nor structures in this ontology. Semi-formal representations express content in a restricted and structured form of natural language or an artificial formally defined language (Sowa 1991). Logic based languages provide a formal way to represent knowledge. A logic-based formalism consists of a set of primitive expressions (constant symbols, function symbols, predicate symbols, variables and connectives, quantifiers – universal and existential) and syntax or set of formation rules to create complex

expressions. Production rules are a knowledge representation language with a pattern-directed inference system (Waterman and Hayes-Roth 1979). Pattern-directed inference systems consist of transition predicates, an optional condition, and an action. Semantic Nets are formalisms based on the notions of associations among concepts and their related properties as the basic artifacts of knowledge (Reichgelt 1991; Sowa 1993). In the frame-based languages, knowledge is stored in larger chunks with usually many connections between the chunks (Minsky 1975). Frames are common in intensional knowledge representations and are described as slots. Description Logics (Borgadia 1995) provide a language for capturing declarative knowledge about a domain and a classifier that allows reasoning about that knowledge. Information captured using description logics is classified in a hierarchical lattice of concepts (comparable to classes, or frames), their inter-relationships or roles (comparable to slots in frame systems) and individual objects (instances) (Baker 1999).

Most formalisms have certain advantages and certain disadvantages. For example default reasoning is a problem with logic-based languages while semantic-nets and frame-based representations provide a natural way to deal with this type of reasoning. On the other hand, semantic-nets and frames have problems defining new concepts and expressing arbitrary disjunctions. To overcome such problems, several hybrid representations were developed. A brief comparative assessment of these formalisms is provided in Table 1.

## Computer Languages for Ontology Representation

Several languages are based on the formalisms discussed in the previous section. We use the term language to refer to those formalisms that can be used directly to create computer programs. Table 2 provides a comparison of some of the popular languages based on the criteria often needed to represent constructs in ontology. Corcho and Gomez-Perez (2000) provide a similar kind of analysis and we have adapted some of that information in Table 2. Table 3 exemplifies a few common ontologies and the language in which they have been implemented. The symbol ? used in the tables implies that the attribute concerned is not determinable based on the published information.

## Ontological Engineering

This section focuses on a methodology for ontology construction. This method 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:

1. What is the purpose for which an ontology is needed?
2. What skills are needed in conceptualizing and building the ontology?
3. What constitutes the proposed ontology?
4. 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. Ontological engineering is quite similar, except that it represents a significant magnitude of expansion. We synthesize an approach to ontological engineering by drawing from a large body of literature on the various facets of ontology development.

Noy and McGuinness (2001) summarize the major purposes of ontologies. In particular, these broad purposes translate to following specific objectives:

- 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 et al. 2002)?
- Can we enable communication and interoperability among analysts when dealing with diverse system components (Uschold 1998)?
- Can we create design templates at various levels of detail granularity that would lead to rapid systems design and development (Zhang et al. 2003)?
- 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 and Uschold 1999)?
- Can we enable interoperability among heterogeneous systems through a shared understanding at a meta-level ontology (Baker et al. 1999)?

**Table 1. A Comparative Assessment of the Formalisms Discussed in This Section**

<b>Formalism</b>	<b>Advantages</b>	<b>Disadvantages</b>	<b>Implementation Examples</b>	<b>Ontology Based on Formalism</b>
Informal	Quick	No Structure Maintenance difficult Interpretation problems	List	None well known
Semi-Formal	Quick Better clarity than informal formalism Good for intermediate representations	No common formal semantics No model or proof theory possible	Lists, Labeled graphs, Markup languages	Chemicals (Loez et al. 1999)
Logic-based	High expressive power Allows for creation of arbitrary attributes and constraints. No overt ontological content	No naturalness with expert knowledge Semi-decidable	FOPC, First order predicate logic, Second order predicate logic, KIF, LOOM, OML, etc.	TOVE (Fox 1996)
Production Rules	Naturalness with expert knowledge Modularity Restricted syntax Problem-solving process	Limitations in expressive power. Difficult to express structure	OPS	—
Semantic Nets	Conceptually simple representation	No semantics to support interpretation No axioms to support reasoning	SNePS (Shapiro, 1979), Conceptual Graphs (Sowa 1984), KL-One (Brachman, 1979)	—
Frame-based	Naturalness with the way domain experts think, Hierarchical structure, Supports default reasoning	Absence of clear semantics (Implementations have provides some mechanisms to overcome this disadvantage)	KRL, CLIPS, XOL	EngMath (Gruber and Olsen 1984), EcoCyc (Karp 2000)
Description Logics	Well understood theoretical principles, Logic can be precisely expressed, Automatic derivation of classification taxonomies	One has to build sanctions or restrictions as needed. Formalism does not provide it.	GRAIL (Rector, Bachhoffer, Goble, Harrochs, Nowlan, and Soloman 1997), Classic (Borgida 1989)	GALEN (Rector 2002)
Mixed-Formalisms	Removes many problems of other formalisms	Depends on hybrid	F-Logic (Kifer, 1995), OIL	TAMBIS (uses: OIL, GRAIL) (Steven et al. 1998)

Table 2. Comparison of Languages Based on Constructs Needed in an Ontology

CRITERIA	Ontolingua	GRAIL	XOL	SHOE	OML	RDF(S)	OIL	LOO M	OIL+ DAML
<b>CONCEPTS</b>									
Subclass of	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Not subclass of	Yes	Yes	No	No	No	No	Yes	Yes	Yes
Exhaustive decompositions	Yes	Yes	No	No	No	No	Yes	Yes	Yes
Disjoint decomposition	Yes	Yes	No	No	No	No	Yes	Yes	Yes
<b>Attributes</b>									
Instance Attributes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Class Attributes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes
Local Scope	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Global Scope	Yes		Yes	No	Yes	Yes	Yes	Yes	Yes
<b>Facets</b>									
Default Slot Value	Yes	Yes	Yes	No	No	No	No	No	No
Type constraints	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cardinality Constraints	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
<b>RELATIONS</b>									
Type Constraints	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integrity Constraints	No	Yes	No	No	Yes	No	No	No	No
<b>AXIOMS</b>									
First-order logic	Yes	Yes	No	☺	Yes	No	☺	☺	☺
Embedded logic	No	No	No	☺	No	No	No	No	No
<b>INSTANCES</b>									
Instance of Concepts	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Facts	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Claims	Yes	Yes	No	Yes	No	☺	☺	☺	☺
<b>FUNCTIONS</b>	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes

**Table 3. Languages Used to Implement Ontologies**

<b>Ontology</b>	<b>Purpose</b>	<b>Ontology Type</b>	<b>Reference</b>	<b>Language</b>
TAMBIS	Integration of heterogeneous bioinformatics sources.	Domain	(Baker 1999)	GRAIL and OIL
GALEN	Provide coherence in medical terminology, for applications such as medical record keeping, etc.	Domain	(Rector 1999)	GRAIL
Chemicals	To provide information about Chemicals [Elements from the periodic table].	Domain	(Lopez et al. 1999)	Semi-Formal. Implemented using Ontolingua
TOVE	Enterprise Modeling	Enterprise	(Fox 1996)	First-order predicate logic; Implemented using Quinus Prolog (axioms), and the rest in C++
ONIONS	Integration of terminological ontologies in medicine	Domain	(Gabgemi et al. 1996)	Ontolingua Formalism – Conceptual Graph
Gene Ontology [GO]	Provide structured vocabularies for the description of molecular function, biological processes and cellular component of gene products in any organism.	Domain	(Ashburner 2002)	☺ Copyrighted
CYC	The Cyc Knowledge Server is a very large, multi-contextual knowledge base and inference engine. Cyc is intended to provide a "deep" layer of understanding that can be used by other programs to make them more flexible.	Linguistics	(Lenat 1990)	Cycl (based on first-order predicate calculus (FOPC), with extensions to handle equality, default reasoning, skolemization, and some second-order features)
GUM (Generalized Upper Ontology)	Linguistic categories	Linguistics	(Bateman et al. 1995)	LOOM
SENSUS	Provides vocabulary to describe various senses of a word and the relationship between senses.	Linguistics	(Swartout 1997)	Ontolingua
EngMath	Mathematical modeling in Engineering	General	(Gruber 1994)	KIF
PhysSys	Modeling , Simulation and Designing Physical System	General	(Borst 1996)	☺
EcoCyc	Covers E. coli. genes, metabolism, regulation and signal transduction	Domain	(Karp 2000)	☺

The skills needed to build meaningful and tractable ontologies are conceptual modeling skills, domain-specific expertise, and systems engineering. Typically, an ontology consists of concepts, relationships and behaviors, together describing a specification of conceptualization (Gruber 1993). Using this view, we propose a practical methodology for ontology development.

## **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? At the outset, we set out with two caveats: no ontology is complete and 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 course-corrected as the ontology begins to take shape. In this process, the ontology builder may have to revisit some of the earlier developments with a view to refine and strengthen the ontology based on the persistent learning that occurs throughout the development process. This iterative process is not a pre-specified and fully structured ; clearly, such a prescription may not work with most ontology builders. Instead, we suggest a non-fully specified, semi-structured strategy of criss-crossing among the various developments that occur during the process. 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 given below, which developers can 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. A focus on the specific goals of the ontology is essential. While the criteria for evaluating an ontology suggested by Gruber (Gruber 1993) are more appropriate while actually developing the ontology, we suggest the notion of informal competency questions proposed by Gruninger and Fox (Gruninger and Fox 1995) at this stage. These questions can be used to guide the scope definition, such as what ontology is needed, what should be its level detail, will it serve our purpose, and similar enquiries.

### ***Guideline 2: Perform a Baseline Analysis***

The baseline analysis consists two components: Brainstorming and Review of existing ontologies and relevant literature. Uschold and Gruninger (1996) suggest the use of sustained brainstorming sessions to produce all relevant concepts and relationships, eliminating redundancies and ambiguities, and building a tentative structure of the ontology. Concurrently or in some sequence, the review step may be carried out. The ontologies should be reviewed with the following questions in mind:

1. How have they been constructed?
2. How do they represent knowledge?
3. How are they used?
4. What construction approaches, representation structures and applications from the existing ontologies are relevant to our needs?
5. At what level can they be used for our needs?
  - (a) Knowledge capture process level
  - (b) Meta knowledge levels
  - (c) Specific knowledge levels
  - (d) Knowledge representation levels
  - (e) System design and development levels
  - (f) Application levels

### ***Guideline 3: Anchor Your Ontology Well and Use Cues to Guide Its Development Throughout***

It is essential that an ontology is well anchored. The anchor points could be domain-specific, context-specific, or even literature-specific. Clearly, the brainstorming and review components could cause a tremendous 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:

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.

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 crucial 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 evidences tend to drive the 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 evidences that are appropriate and necessary for the goals of the ontology; this strategy is an applied one and is more closely allied with an engineer's approach. The cue-n-anchor notion is central to the Pull approach and is vital 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 assume more definitive shapes at this stage, such as whether the ontology is complete enough to serve the ultimate needs and whether it is sound and free of any internal and external contradictions. We use the glossary and the refined competency questions to constitute the baseline ontology document.

#### ***Guideline 5: Structure the Baseline Glossary into a Specifiable Ontology Using a Criss-Cross Strategy***

An ontology specification could consist of (a) simple taxonomic structures, (b) specific data modeling structures such as object specifications, (c) behavior models. A taxonomic structure usually serves as the backbone of an ontology and usually encompass the basic and extended distinctions. The taxonomies should be sound; this means no internal contradictions are 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 taxonomies as indicated above. Incorporating class structures and their properties such as slots, facets etc., within the extended distinctions and linking the distinctions to different behavior models, a more formal specification of the ontology is obtained. Finally, applying logical and evidential reasoning, the axioms and constraints are derived. The emerging structure should then be tested for soundness using the constraints. This process constitutes the verification and validation steps in ontology construction.

#### ***Guideline 6: Using the Cue-N-Anchor Approach, Decide on Integrating Existing Ontologies with the One Being Built and Evaluate Formal Representation Mechanisms***

#### ***Guideline 7: Develop the Formal Representation of the Ontology***

The formal specification should include: (a) the foundational conceptual model of the universe being modeled at appropriate levels of granularity, and (b) full schema of the ontology describing the concept-relations 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 and also Gruber's criteria for ontologies.

### **Putting It All Together**

Using the above guidelines, we developed a new ontological engineering process model, termed the Helix-Spindle model (Kishore et al. Forthcoming). The Helix-Spindle process model (illustrated in Figure 1) consists of three major phases captured and represented using the imagery of a helix and a spindle. The three phases – a conception phase, an elaboration phase, and a definition phase – are represented as one full-loop each of the helix.

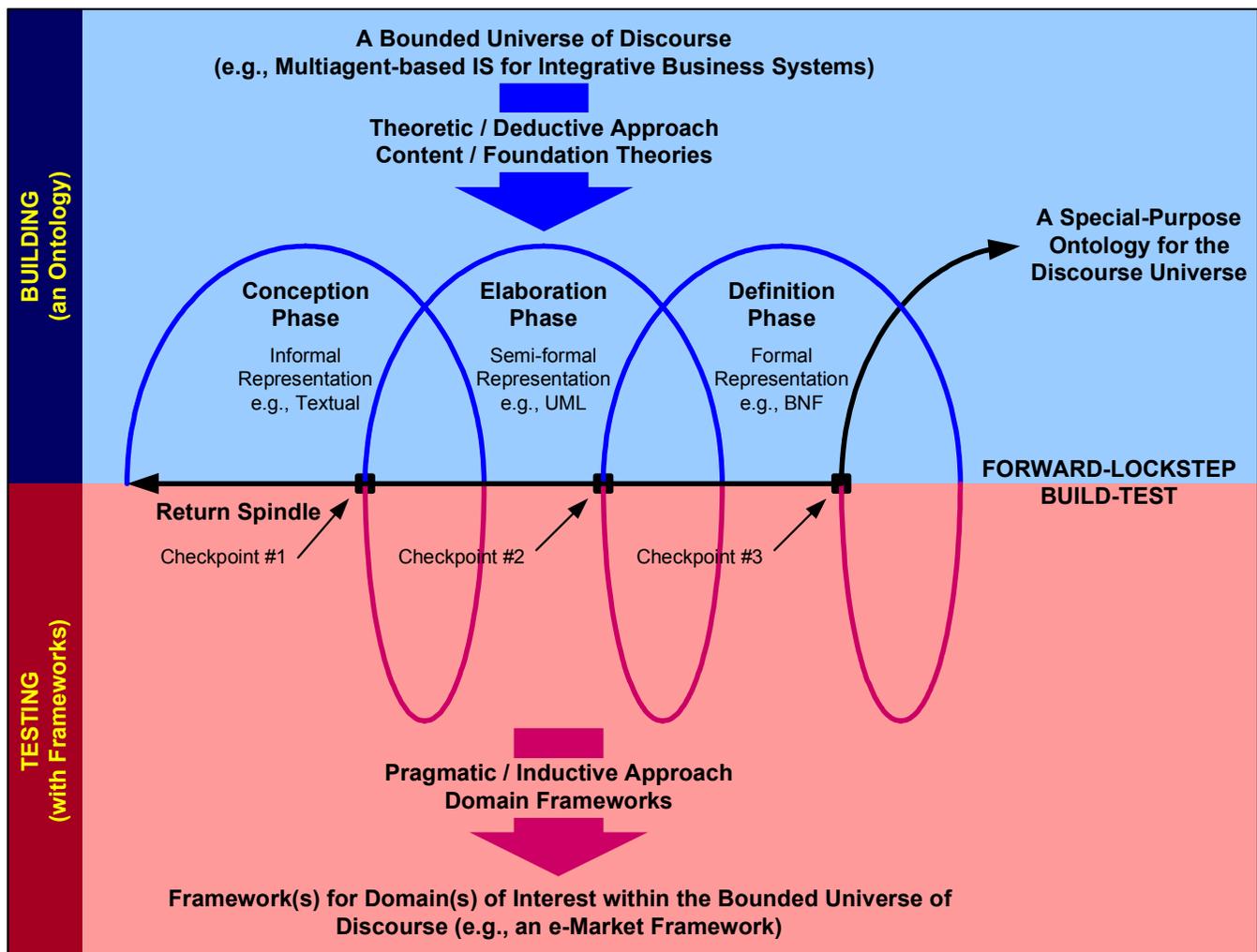


Figure 1. The “Helix Spindle” Process Model for Ontological Engineering

## Unfinished Business in Ontology

For an ontology to be really useful over the long haul the issue of mapping an ontology to other parts of the system such as databases, user-interfaces, and organizational processes needs to be addressed better (Ding and Foo 2002). There is a need to establish theoretical and empirical foundations to mapping. We also need better theoretical, empirical, and best practice papers that provide sound 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 the area of semantic nets, where we are seeing significant development.

Another open research issue is metrics. How do we evaluate ontologies? Not much research activity is ongoing in this area currently. We as a community need to develop more tools, and better languages that could help in the creation and integration of ontologies with information systems.

## References

- Ashburner M., and Lewis S. "On Ontologies for Biologists: The Gene Ontology—Uncoupling the Web," in *Silico Biology. Novartis Found Symp* 247: 66-80; discussion 80-3, 84-90, 244-52.
- Baker, P. Gloge, C., Bechhofer, S., Paton, N., Stevens, R., and Brass, G. "An Ontology for Bioinformatics Applications," *Bioinformatics*, 15 (6), 1999, pp. 510-520.
- Bateman, J. A., Magnini, B., and Rinaldi, F. "The Generalized Italian, German, English Upper Model," *Proceedings of the European Conference on Artificial Intelligence: Workshop: Comparison of Implemented Ontologies*, Amsterdam, 1994.
- Bateman, J., Magnini, B., and Fabris, G. "The Generalized Upper Model Knowledge Base—Organization and Use," in N. Mars (ed.), *Towards Very Large Knowledge Bases*, Amsterdam: IOS Press, 1995.
- Borgida, A. "Description Logics in Data Management," *Knowledge and Data Engineering*, 7(5), 1995, pp. 671-682.
- Borst, P., Benjamins, J., Wielinga, B., and Akkermans, H. "An Application of Ontology Construction," *Workshop on Ontological Engineering, ECAI '96*, 1996, pp. 5-16.
- Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. "What are ontologies and why do we need them?" *IEEE Intelligent Systems*, 14(1) Jan/Feb 1999, pp. 20-26.
- Corcho, O., and Gomez-Perez, A. "Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages," *Proceedings of the 14<sup>th</sup> European Conference on Artificial Intelligence, 2000, Workshop on Application of Ontologies and Problem Solving Methods*, Berlin, 2000.
- Ding, Y., Fensel, D., Klein, M., and Omelayenko, B. "The Semantic Web: Yet Another Hype?," *Data and Knowledge Engineering*, 41(3), 2002, pp. 205-227.
- Fox, M., Barbuceanu, M., and Gruninger, M. "An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structures and Behaviour," *Computers in Industry*, 29, 1996, pp. 123-134.
- Gabgemi, A., Steve, G., Giacomelli, F. "ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration", in *Proceedings of ECAI'96 Workshop: Ontological Engineering*, 1996.
- Gruber, T. R. "A Translation Approach to Portable Ontology Specification," *Knowledge Acquisition* 5(2), 1993, pp. 199-220.
- Gruber, T. R., and Olsen, G. "An Ontology for Engineering Mathematics," in J. Doyle, P. Torasso, and E. Sandewall, (eds.), *Fourth International Conference on Principles of Knowledge Representation and Reasoning, Gustav Stresemann Institut, Bonn, Germany*, San Mateo, CA: Morgan Kaufmann, 1994.
- Grüninger, M., and Fox, M. "Methodology for the Design and Evaluation of Ontologies (1995)," *Proceedings of the International Joint Conference on Artificial Intelligence, Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Quebec, Canada, April 13, 1995.
- Guarino, N. "Formal Ontology and Information Systems," *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems*, Trento, Italy, 1998, pp. 3-15.
- Karp, P., Riley, M., Saier, M., Paulsen, I., Paley, S., and Pellegrini-Toole, A. "The EcoCyc and MetaCyc Databases," *Nucleic Acids Research*, 28, 2000, pp. 56-59.
- Kifer, M., Lawsen, G., and Wu, J. "Logical Foundation of Object-Oriented and Frame-Based Systems," *Journal of the ACM*, 42(2), July 1995, pp. 741-843.
- Kishore, R., Zhang, H., and Ramesh, R. "Ontologies, Frameworks, and Systems: A Helix-Spindle Model for Ontological Engineering," *Communications of the ACM*, forthcoming.
- Lenat, D., and Guha, R. *Building Large Knowledge-Based System: Representation and Inference in the CYC Project*, Reading, MA: Addison-Wesley, 1990.
- López, F., Gomez-Perz, A., Sierra, J., and Sierra, S. "Building a Chemical Ontology Using Methontology and the Ontology Design Environment," *IEEE Intelligent Systems*, 14(1), Special Issue on Ontologies, Jan./Feb. 1999, pp. 37-46.
- MacGregor, R. "The Evolving Technology of Classification-Based Knowledge Representation Systems," in J. Sowa (ed.), *Principles of Semantic Networks; Explorations in the Representation of Knowledge*, San Mateo, CA: Morgan Kaufman Press, 1991.
- Miller, G., A. "WordNet: A Lexical Database for English," *Communications of the ACM*, 38(11), November 1995, pp. 39-41.
- Minsky, M. "A Framework for Representing Knowledge," in P. Winston (ed.), *The Psychology of Computer Vision*, New York: McGraw-Hill, 1975.
- Noy, N. F., and McGuinness, D. L. "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Medical Informatics Report SMI Technical SMI-2001-0880, 2001 (available online at <http://smi-web.stanford.edu/pubs/>).
- Rector, A., and Rogers, J. "Ontological Issues in Using a Description Logic to Represent Medical Concepts: Experience from GALEN," *International Medical Informatics Association (IMIA), Working Group 6: Workshop: Terminology and Natural Language in Medicine*, Phoenix, Arizona, 1999.
- Reichgelt, H. *Knowledge Representation: An AI Perspective*, Norwood, NJ: Ablex Publishing Corporation, 1991.

- Robin, R. *The Peirce Papers: A Supplementary Catalogue*, Cambridge, MA: Harvard University Library Microreproduction Service, 1970.
- Smith, B. "Ontology: Philosophical and Computational," Working Paper, The State University of New York, 2003.
- Sowa, J. F. *Conceptual Structures: Information Processing in Mind and Machine*, Reading, MA: Addison-Wesley 1984.
- Sowa, J. F. "Logical Structures in the Lexicon," *Proceedings of the Special Interest Group on the Lexicon Workshop*, Berkeley, CA, 1991, pp. 39-60.
- Sowa, J. F. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, New York: Brooks/Cole, 2000a.
- Sowa, J. F. "Ontology, Metadata, and Semiotics," *Proceedings of the International Conference on Conceptual Structures*, Darmstadt, Germany, 2000b, pp. 55-81.
- Sowa, J. F. "Relating Diagrams to Logic," *International Conference of Conceptual Structures*, Quebec City, Canada, 1993, pp. 1-35.
- Steve, G., and Gangemi, A. "ONIONS Methodology and the Ontological Commitment of Medical Ontology ON8.5," in B. Gaines and M. Musen (eds), *Proceedings of Tenth Knowledge Acquisition Workshop*, Banf, Alberta, Canada, 1996.
- Stevens, R., Baker, G. P., Bechhofer, S., Ng, G., Jacoby, A., and Paton, N. "TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources," *Sixth International Conference on Intelligent Systems for Molecular Biology*, Montreal, Quebec, Canada, 1998.
- Storey, V. C., Goldstein, R. C., and Ullrich, H. "Naive Semantics to Support Automated Database Design," *IEEE Transactions on Knowledge and Data Engineering*, 14(1), 2002, pp. 1-12.
- Swartout, B., Patil, R., Knight, K. and Russ, T. "Toward Distributed Use of Large-Scale Ontologies," *American Association of Artificial Intelligence Spring Symposium Series, Workshop on Ontological Engineering*, Providence, Rhode Island, 1997.
- Uschold, M. "Where Are the Killer Apps?" *Thirteenth European Conference on Artificial Intelligence: Workshop on Applications of Ontologies and Problem Solving Methods*, Brighton, United Kingdom, 1998.
- Uschold, M., and Gruninger, M. "Ontologies: Principles, Methods and Applications," *Knowledge Engineering Review*, 11(12), December 1996.
- Uschold, M., and Jasper, R. "A Framework for Understanding and Classifying Ontology Applications," *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods*, Stockholm, Sweden, August 1999.
- Waterman, D. A. "Exemplary Programming in RITA," in D. A. Waterman and F. Hayes-Roth (eds.), *Pattern Directed Inference Systems*, New York: Academic Press, 1979.
- WordNet, Princeton University, 1997 (available online at <http://www.semanticweb.org/library/>).
- Zhang, H., Kishore, R., and Ramesh, R. "The GRITIKA Ontology for Modeling Multiagent-Based Integrative Business Information Systems," Working Paper, School of Management, SUNY at Buffalo, 2003.