Ontology as a mechanism for application integration and knowledge sharing in collaborative design: a review

Xie, H.; Shen, W.

Publisher’s version / Version de l'éditeur:

10th International Conference on Computer-Supported Cooperative Work in Design (CSCWD 2006) [Proceedings], pp. 843-849, 2006-05-03

Ontology as a mechanism for application integration and knowledge sharing in collaborative design: a review

IMTI-XP-443

Xie, H.; Shen, W.

A version of this document is published in / Une version de ce document se trouve dans:
10th International Conference on Computer-Supported Cooperative Work in Design (CSCWD 2006), Nanjing, China, May 3, 2006, pp. 843-849

Ontology as a Mechanism for Application Integration and Knowledge Sharing in Collaborative Design: A Review

Helen Xie and Weiming Shen
Integrated Manufacturing Technologies Institute
National Research Council of Canada
800 Collip Circle, London, ON, Canada N6G 4X8
helen.xie@nrc.gc.ca; weiming.shen@nrc.gc.ca

Abstract

In this paper, we draw attention to ontology as a mechanism for application integration and knowledge sharing in collaborative design. The paper provides a comprehensive literature review on this topic. It starts with its concepts, classification, and difference from other related technologies. Then the usage of ontology is discussed. Following the ontology life cycle, this paper compares several ontology authoring languages, ontology building methodologies and tools. It reviews some recent work on applying ontology to collaborative design. It concludes with a discussion on future R&D directions.

Keywords: Ontology, Collaborative Design, Application Integration, Knowledge Sharing.

1. Introduction

In collaborative design environment, often there are needs for several application systems to work together to solve a problem, i.e. CAD and analysis packages. These application systems need to have shared understanding on common concepts. Moreover, intelligent application systems need to build knowledge bases for automatic reasoning. When a new task is specified, the specialized knowledge should be created and interoperated with existing systems to perform its reasoning. Again, the intelligent application systems need to have shared understanding on declarative knowledge, problem-solving techniques, and reasoning services.

In this paper, we draw attention to ontology as a mechanism for application integration and knowledge sharing in collaborative design. We begin with ontology concept, classification, and difference from other related methodologies, i.e. conceptual modeling and knowledge base, in the following section. In section 3, we discuss the usage of ontologies, with emphasis on data interoperability and knowledge reuse. Following the ontology life cycle, this paper compares several ontology authoring languages, and ontology building methodologies and tools. It reviews some recent work on applying ontology to collaborative design. It concluded with a discussion on what can be done further to enhance the ability for the application integration and knowledge sharing.

2. Ontology Overview

The term ontology has been in use for many years. While it has been rather confined to the philosophical sphere in the past, ontology is now gaining interest and acceptance in computer science. Guarino [16] provides a collection of research fields in computer science that recognize its importance, including knowledge engineering, knowledge representation, qualitative modeling, language engineering, database design, information modeling, information integration, information retrieval and extraction, knowledge management and organization, agent-based systems design.

In computer science literature, there are many definitions of ontology. One widely cited definition is given by Gruber [21]: an ontology is an explicit specification of a conceptualization. Ontology may take a variety of forms, but necessarily it will include a vocabulary of terms and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms [45].

2.1 Classification of Ontologies

Ontology can be classified in terms of types and formality. Basically, there are four types of ontologies: domain ontologies, task ontologies, common sense ontologies, and meta-ontologies. Ontologies can also be highly informal, semi-informal, semi-formal, and rigorously formal.

2.1.1 Types of Ontologies

Domain ontologies provide a vocabulary for describing a given domain. They typically include terms related to: (1) objects in the domain and its components; (2) A set of verbs and paraphrases that name activities and processes that take place in the domain; (3) Primitive concepts appear in the theories, relations, and formulas that govern the domain. Typically, more specialized domain specific schema must be created to make the data useful for real world decisions [19].

Task ontologies describe the strategies taken by problem solvers (fuzzy logic, neural network, constraint solver, etc.) to attach domain problems [34] [41]. They provide a vocabulary for describing terms involved in problem-solving processes, which could be attached to similar tasks that may or may not be in the same domain. They include nouns, verbs, paraphrases, and adjectives related to the task [19].

Ontology which is not tied to a particular problem domain but attempts to describe general entities, such as time, space, events, etc. is known as a common sense ontology or foundation ontology or upper ontology. Some of these ontologies include CYC [28], Dublin Core [48], SUMO, and WordNet.

Meta-ontologies provide the basic core of concepts used to codify either domain ontology, task ontology, or common-sense ontology in a formal knowledge representation language. Depending on a particular formal language, the meta-ontologies capture the representation primitives used in the knowledge representation language.

2.1.2 Formality Dimensions

The degree of formality by which a vocabulary is created and meaning is specified varies considerably. Four dividing points are [44] [33]:

- Highly informal: expressed loosely in natural language, i.e., catalogues, glossary, thesauri.
- Semi-informal: expressed in a restricted and structured form of natural language, greatly increasing clarity by reducing ambiguity, i.e. Yahoo's categories, providing a basic notion of generalization and specialization, but not a strict subclass or "isa" hierarchy.
- Semi-formal: expressed in an artificial formally defined language, i.e. strict subclass hierarchies, formal instance relationships.
- Rigorously formal: meticulously defined terms with formal semantics, theorems, and proofs of such properties as soundness and completeness, i.e. classes including property information, value restrictions, more expressively, arbitrary logical statements, first order logic constraints between terms and more detailed relationships such as disjoint classes, disjoint coverings, inverse relationships, part-whole relationships, etc.

Based on different usages of ontology, ontology can be defined in a variety of degrees of formality.

2.2 Scope of Ontology

2.2.1 Ontology vs. Object Models or Conceptual Analysis

What are the dividing line between ontology and a number of other approaches (e.g. object models, conceptual analysis) of representing concepts and conceptualization? Ontology presents physical existence, while object model presents application point of view. According to [16], ontologies present their own methodological and architectural peculiarities. On the methodological side, the main peculiarity is the adoption of a highly interdisciplinary approach, where philosophy and linguistics play a fundamental role in analyzing the structure of a given reality at a high level of generality and in formulating a clear and rigorous vocabulary. On the architectural side, the centrality of the role that ontology can play in an information system leads to the perspective of ontology-driven information systems.

2.2.2 Ontology vs. a Knowledge Base

Is there a difference between ontology and a knowledge base? What is the difference? The differences are now summarized as follows [19]:

Contents and scope: According to [36], ontology consists of classes, properties, and restrictions. Ontology together with a set of individual instances of classes constitutes a knowledge base. However, in reality, there is a fine line where the ontology ends and the knowledge base begins. Deciding whether a particular concept is a class or an individual instance depends on what the potential applications of the ontology are. The lowest level of granularity in the representation is considered as an individual instance.

Features of the language used to codify the knowledge: Ontologies should be written in an expressive, declarative, portable, domain-independent, and semantically well-defined, machine-readable language, which should be independent of any particular choice of target machine-readable language of the application, such as LOOM [30] [31], CycL [28], and Ontolinga [12][21].

Goal of the knowledge codification: Ontologies are designed for knowledge sharing and reuse purposes and knowledge bases are not. As a result, their definitions should be conceptualized with enough abstraction and generality. These features guarantee that ontology definitions are independent of their final uses.

3. Use of Ontology

At a high level, the use of ontology seems aiming at reuse. Some common usages of ontology found in literature have mainly served for three purposes: communication and shared understanding between people, interoperability among computer systems and agents, and knowledge sharing and reuse [44] [46]. These three usages are built in three levels: vocabulary, data, and knowledge.

3.1 Vocabulary for Communication and Shared Common Understanding between People

This area involves informal use of ontology. The intent is to build shared and controlled vocabulary – i.e., a finite list of terms, so that people use the same set of terms to communicate. Catalogs, glossary, and thesauri are all contributed to this intend. In addition, taxonomy may be used as ontology for Website organization, browsing and navigation support. It may also be used as "umbrella" structure from which to extend content.

3.2 Data for Interoperability

Building on the goal of reuse, the data for interoperability is the next level to the shared understanding among computer systems and agents. Let's consider two ways the interoperability can support the communication between systems or software agents.

First, it provides the support in different domains for multiple systems or software agents. Considering a scenario in product life cycle support, there are multiple domains for ordering, product design, production planning, manufacturing, assembly, inventory control, delivery, and maintenance. Ontologies provide standardizing terminology in the intersection of any two domains.

Second, it provides the support for multiple systems in the same domain, either by standardizing terminology among different systems, or by providing the semantic foundations for translators among different systems.

3.3 Knowledge Reuse

Knowledge reuse was one of the driving forces behind recent surge in ontology research. It concerns the details of knowledge in a knowledge base and requires a deeper level of understanding and commitment. An ontology for knowledge reuse most likely contains constraints and restrictions in addition to concepts and properties. Task ontologies and description logic based ontologies are mostly discussed in the literature for the knowledge reuse purpose.

Problem solving methods (PSMs) are designed to support knowledge sharing and reuse. While ontologies capture a shared terminology, problem solving methods define generic algorithms, which can be applied to different tasks and domains. Problem solving methods and ontologies provide the two essential technologies which enable the development of knowledge-based applications by reuse.

The task ontology depends upon the problem solving process. It is a system of vocabulary for describing the information structure of the problem solving process domain independently. The ontology is explicit of problem solving context. It contributes to extracting necessary domain knowledge for performing the problem solving. Constraint satisfactory is an example of a problem solving process. An ontology for constraint satisfactory would contain vocabulary of information structure and computational mechanisms. The knowledge for problem solving forms a knowledge base.

Problem solving independent of a problem domain through knowledge reuse requires reasoning capability. Description logic provides knowledge representation formalisms for reasoning. Description logic ontologies differ in their approach to construction. Rather than manually create a hierarchy and then assign properties to concepts, the process is turned on its head. Each concept is assigned a logic definition which is then used to derive a classification. There is more than one way to classify a set of concepts. This approach allows different classifications to be produced for different purposes based on the same underlying terminological knowledge [32].

4. Authoring Languages for Ontology

4.1 Ontology Languages

Authoring languages are essential in ontology development. Significant efforts have been contributed to the development of various ontology languages. Before looking at different languages, it is necessary to understand the requirements for the languages.

A list of requirements has been created by the Web-Ontology (WebOnt) Working Group of W3C when developing the latest Web ontology language, OWL. These requirements give a good overview of what kind of functionalities are possible and expected to be useful in semantic web applications. The requirements address issues such as ontology sharing, ontology interoperability, ontology evolution, balance of expressivity and scalability, inconsistency detection, compatibility with other standards, ease of use and internationalization [23]. Bechhofer [3] also presented a collection of ontology language requirements.

Well known ontology languages include:

- XML, XML Schema [55]: XML provides a set of rules for creating vocabularies that can bring structure to both documents and data on the Web.

XML Schema is a metadata modeling language for defining and sharing XML documents.

- SHOE [29]: SHOE is an HTML-based knowledge representation language. SHOE is a superset of HTML which adds the tags necessary to embed arbitrary semantic data into web pages.
- RDF [51]: The Resource Description Framework (RDF) is a metadata framework that provides a degree of semantic interoperability among applications that exchange machine-understandable metadata on the Web. The goal of RDF is to define a mechanism for describing resources that makes neither assumptions about a particular application domain nor the structure of a document. The data model of RDF consists of three types: resource (subjects), entities that can be referred to by an address at the WWW; properties (predicates), which define specific aspects, characteristics, attributes or relations used to describe a resource; and statements (objects), which assign a value for a property in a specific resource [8].
- RDF Schema (RDFS) [54]: RDF Schema is recognized as an ontology/knowledge representation language. It provides a means to define vocabulary, structure, constraints by using classes and properties (binary relations), range and domain constraints on properties, and subclass and sub-property relations. However, formal semantics for the primitives defined in RDFS are not provided, and the expressive power of these primitives is limited to ontology modeling and reasoning.
- DAML and DAML-ONT [24]: The DARPA Agent Markup Language (DAML) program was initiated with the aim of providing the foundations of a next generation "semantic" Web [Hendler 2000]. RDFS was seen as a good starting point, and was already a proposed W3C standard, but it was not expressive enough to meet DAML's requirements. A new language called DAML-ONT was therefore developed that extended RDF with language constructors from object-oriented and frame-based knowledge representation language. Like RDFS, DAML-ONT suffered from a rather weak semantic specification, and it was soon realized that this could lead to disagreements, both amongst humans and machines, as to the precise meaning of terms in DAML-ONT ontology.
- OIL [49]: Like DAML-ONT, OIL has RDFS based syntax (as well as an alternative XML syntax) and a set of language constructors based on frame-based languages [13] 14]. The developers of OIL placed a stronger emphasis on formal rigor, and the language was explicitly designed so that the semantics could be specified via a mapping to very expressive description logic [25]. OIL provides most of the modeling primitives commonly used in frame-based and description logic oriented ontologies. It features simple, clean and well-defined first-order semantics. It provides automated reasoning support, e.g., class consistency and subsumption checking [5].
- DAML+OIL [25] [53]: DAML+OIL is designed to describe the structure of a domain. DAML+OIL takes an object oriented approach, with the structure of the domain being described in terms of classes and properties. An ontology consists of a set of axioms that assert, e.g., subsumption relationships between classes or properties. Asserting that resources are instances of DAML+OIL classes is left to RDF. From a formal point of view, DAML+OIL can be seen to be equivalent to a very expressive description logic (DL), with a DAML+OIL ontology corresponding to a DL terminology.
- OWL [52]: OWL (Web Ontology Language) is a semantic markup language for publishing and sharing ontologies on the Web. OWL is developed as a vocabulary extension of RDF and is derived from the DAML+OIL Web Ontology Language. OWL builds on RDF and RDF Schema and adds more vocabulary for describing properties and classes: among others, relations between classes (e.g., disjointness), cardinality (e.g., exactly one), equality, richer typing of proprieties, characteristics of properties (e.g., symmetry), and enumerated classes.

4.2 Comparison of Ontology Languages

In comparison of these ontology languages, Gil and Ratnakar [17] proposed dimensions for the comparison, including context, subclasses and properties, primitive data types, instances, property constraints, property values, negation, conjunction and disjunction, inheritance, definitions, and expressiveness. Based on these dimensions, XML, RDF, DAML+OIL along with their associated schema were reviewed.

Corcho and Gomez-Perez [8] established a common framework to compare the expressiveness and reasoning capabilities of "traditional" ontology languages (ontolingua, OKBC [7], OCML [35], FLogic [26], LOOM) and "web-based" ontology languages (SHOE, XOL [27], RDF, OIL, etc.). The framework distinguishes between knowledge representation and inference mechanism. Domain knowledge describes the static information and knowledge objects in application domain. According to Gruber [21], domain knowledge in ontologies can be specified using five kinds of components: concepts, relations, functions, axioms and instances. Concepts in the ontology are usually organized in taxonomies. The inference mechanism describes how the static structures represented in the domain knowledge can be used to carry out a reasoning process [2]. There is a strong relationship between both dimensions, as the structures used for representing knowledge are the basis for the reasoning process. The criteria consist of inference engine, automatic classification, exception, inheritance in monotonic, non-monotonic, simple, or multiple, executable procedure, constraint checking, and forward and backward chaining.

When developing domain ontologies for an application, it is not only necessary to study the knowledge representation and reasoning needs for the application, but also the knowledge representation and reasoning capabilities provided by the language. The above framework will help developers make wise decisions on the selection of the ontology language to use.

5. Ontology Building Methodology and Tools

5.1 Ontology Building Methodology

Gomez-Perez [19] proposed an ontology building methodology which involves design criteria, development and integral activities.

Design Criteria

Based on the work of Gruber [22] and Borgo [4], Gomez-Perez [19] summarized the design criteria and principles as follows:

- Clarity and objectivity: The ontology should provide the meaning of defined terms by providing objective definitions and also natural language documentation of all terms.
- Completeness: A definition expressed by a necessary and sufficient condition is preferred over a partial definition.
- Coherence: It should permit inferences that are consistent with the definitions.
- Maximize monotonic extendibility: New general or specialized terms should be included in the ontology in such a way as does not require the revision of existing definitions.
- Minimal ontological commitments: Making as few claims as possible about the world being modeled. In other words, the ontology should specify as little as possible about the meaning of its terms, giving the parties committed to the ontology freedom to specialize and instantiate the ontology as required.
- Ontological distinction principle: Classes in an ontology should be disjoint. This criterion used to isolate the core of the properties considered to be invariant for an instance of a class is called the Identity Criterion.

Development Activities

The following tasks describe the practical skills, techniques and methods used to develop an ontology [19]:

- Specify: An ontology should not be developed without knowing why this ontology is being built and what are its intended uses and end-users.
- Conceptualize: The goal is to build a conceptual model that describes the problem and its solution.
- Formalize: This activity transforms the conceptual model into a formal model that is semi-computable. Frame-oriented or description logic representation systems could be used to formalize the ontology.
- Implement: To make the ontology computable, ontology needs to be codified in a formal language.
- Maintain: Someone may ask for definitions to be included or modified in the ontology at anytime and anywhere. Guidelines for maintaining ontologies are also needed.

Integral Activities

The following activities interact with the above development activities are required for the successful development of ontologies [19]:

- Acquire knowledge: Knowledge acquisition is the first step for knowledge sharing. An extensive work on capturing knowledge was reported by Uschold [44].
- Integrate: Ontologies are built to be reused. Therefore, existing ontologies should be reused as much as possible.
- Evaluate: Before making an ontology available to others, make a technical judgment with respect to a framework of reference. A framework for evaluating ontologies is available in [18].
- Document: The absence of a sound documentation is also an important obstacle when reusing/sharing ontologies. So if an ontology is to be reused/shared by others, try to document it as detailed as possible.

5.2 Ontology Building Tools and Environments

Numerous commercial and open source software tools are available for building and deploying ontologies. They can be used for building a new ontology from scratch or reusing existing ontologies. Apart from the common editing and browsing functionality, these tools usually include ontology documentation, ontology exportation and importation from different formats, graphical views of the ontologies built, ontology libraries, attached inference engines [20]. Increasingly, these tools support the emerging standard ontology languages. Many more are offering platforms to interchange information among mutually heterogeneous resources including legacy databases, semi-structured repositories, industry-standard directories and vocabularies, and streams of unstructured contents as text and media [9]. Denny's survey covered 94 tools with ontology editing capabilities that can be used to build ontology schemas (terminologies) and/or instance data. These editors may be available as standalone, plugin or online software and not necessary in production level.

Well known ontology building tools include: Apollo [47], LinkFactory [6], OilEd [2], OntoEdit [42], Ontolingua Server [12], OntoSaurus [43], Protégé [37], WebODE [1], and WebOnto [10]. A comparison study of ontology building tools can be found in [11]. An evaluation framework and evaluation results of various ontology tools can be found in [20].

6. Ontology in Collaborative Design

In the area of collaborative design, ontologies are usually: (1) to improve communication among humans; (2) to improve data exchange among programs; and (3) to facilitate knowledge management, particularly knowledge sharing.

Improving communications among humans involves standardizing the vocabulary and integrating new concepts. The goal is to increase mutual understanding among people from different departments, e.g., between design department and production department. A typical example was reported by Genc et al. [15], who provided a hierarchical classification scheme in the domain of snap-fit assemblies.

Improving electronic data exchange requires compatible representation models. The ontology can be used to specify the concepts and vocabulary needed for developing exchange software (using frameworks like STEP/EXPRESS), or in integrating legacy systems when implementing concurrent engineering. When software agents are used, ontology is critical in sharing knowledge among the agents [38].

Ontologies can play an important role in facilitating knowledge management/sharing, automated collaborative design environments. Ontologies can improve a design process by building knowledge base for reuse or guiding the design process. Ontologies have been a very active research in the area of collaborative design. There are at least 6 papers on this topic presented at CSCWD 2005 [39] and at least 3 papers at CSCWD 2004 [40].

7. Conclusions

From the literature review, we can conclude that both domain ontologies and task ontologies are required for supporting knowledge reuse. However, most researchers treat these two types of ontologies separately, resulting integration issues later on. We propose a hierarchical framework in which task ontologies provide a foundation for building domain ontologies. Domain ontologies can be tailored to different problem solving methods with adapters. The combination of task ontologies and domain ontologies can be readily served for building knowledge bases. An ontology building tool based on this idea can be used by domain experts or engineers for knowledge acquisition.

References

[1] J.C. Arpırez, O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez. "WebODE: a scalable workbench for ontological engineering", *In Proceedings of Interactive Tools for Knowledge Capture, KCAP-2001*, Victoria, Canada, October 20, 2001

[2] S. Bechhofer, I. Horrocks, C. Goble, R. Stevens. "OILEd: a Reason-able Ontology Editor for the Semantic Web", *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence*, September 19-21, 2001, Vienna. Springer-Verlag LNAI Vol. 2174, pp 396--408.

[3] S. Bechhofer, C. Globe, I. Horrocks, "Requirements of Ontology Languges", IST Project. IST-2000-29243 Ontoweb. Deliverable 4.1. 22.3.2002. 19p.

[4] S. Borgo, N. Guarino, C. Masolo, "Stratified Ontologies: The case of physical objects", *Workshop on ontological engineering, EVAI'96.* Budapest, Hungary, pp 5-16

[5] J. Broekstra, M. Klein, S. Decker, D. Fensel, F. van Harmelen, I. Horrocks, "Enabling knowledge representation on the Web by extending RDF Schema", *Computer Networks*, 2002, 39, pp. 609-634

[6] W. Ceusters, P. Martens, C. Dhaen, and B. Terzic, "LinkFactory: an Advanced Formal OntologyManagement System", *In Proceedings of Interactive Tools for Knowledge Capture, KCAP-2001*, Victoria, Canada, October 20, 2001

[7] V. Chaudhri, A. Farquhar, R. Fikes, P. Karp, J. Rice, "The generic frame protocol 2.2", July, 1997

[8] O. Corcho, A. Gomez-Perez, "A roadmap to ontology specification languages", *Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management, EKAW00.* 2000

[9] M. Denny, "Ontology tools survey, revisited", http://www.xml.com/pub/a/2004/07/14/onto.html

[10] J. Domingu, "Tadzebao and Webonto: discussing, browsing and editing Ontologies on the Web", *KAW98*, Banff, Canada, 1998

[11] A. Duineveld, R. Studer, M. Weiden, B. Kenepa, R. Benjamis, "WonderTools? A comparative study of ontological engineering tools", *KAW99*, Banff, 1999

[12] A. Farquhar, R. Fikes, J. Rice, "The ontolingua server: a tool for collaborative ontology construction", *Proceedings of 10th Knowledge Acquisition for Knowledge-based Systems workshop, KAW96,* Banff, Canada, 1996.

[13] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein, OIL in an nutshell. *Proc. of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW'00),* 2000, Lecture Notes in Artificial

Intelligence 1937, 1-16

[14] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneiger, "OIL: An ontology infrastructure for the semantic web", *IEEE Intelligent Systems*, 2001, 16(2): 38-45

[15] S. Genc, R.W.Jr. Messler, C.A. Gabriele, A Hierarchical Classification Scheme to Define and Order the Design Space for Integral Snap-Fit Assembly, *Research in Engineering Design*, 1998, 10(2), 94-106.

[16] N. Guarino, Formal Ontology and Information Systems, *Proceedings of FOIS'98*, Trento, Italy, June 1998, pp. 3-15

[17] Y. Gil, V. Ratnakar, "A comparison of (Semantic) Markup Languages", *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference*, 2002, pp. 413-418

[18] A. Gomez-Perez, N. Juristo, J. Pazos, "Evaluation and assessment of knowledge sharing technology", *In Towards very large knowledge bases*, Ed. by N. Mars, IOS Press, Amsterdam, 1995, pp. 289-296

[19] A. Gomez-Perez, "Knowledge sharing and reuse", *The handbook of applied expert systems*, edited by Liebowitz, J. CRC Press LLC, 1997, pp. 10-1—10-36

[20] A. Gomez-Perez, J. Angele, M. Fernandez-Lopez, V. Christophides, A. Stutt, Y. Sure, "A survey on ontology tools", OntoWeb deliverable 1.3, Universidad Politecnia de Madrid.2002.

[21] T. R. Gruber, "A translation approach to portable ontologies", *Knowledge Acquisition*, 1993, 5(2):199-220

[22] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing", *Int. Journal of Human-Computer Studies*, 1995, 43, 907-928

[23] J. Heflin, R. Volz, J. Dale, "Requirements for a Web Ontology Language", *W3C Working Draft*, 08 July 2002. http://www.w3.org/TR/webont-req/

[24] J. Hendler, D. L. McGuinness, "The DARPA agnet markup language", *IEEE Intelligent Systems*, 20000, 15(6):67-73

[25] I. Horrocks, "DAML+OIL: A Reason-able Web Ontology Language", *EDBT 2002*, 2002, LNCS 2287, pp. 2-13

[26] M. Kifer, G. Lausen, J. Wu, "Logical foundations of Object-oriented and frame-based languages", *Journal of the ACM*, 1995

[27] R. Karp, V. Chaudhri, J. Thomere, "XOL: An XML-based ontology exchange language", July, 1999, http://www.ai.sri.com/pkarp/xol/xol.html

[28] D. B. Lenat, R. V. Guha, " Building large knowledge-based systems: Representation and inference in the Cyc project", Addison-Wesley, CA. 1990

[29] S. Luke, J. Heflin, "SHOE 1.01 proposed specification, SHOE project", February, 2000.

http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm

[30] R. MacGregor, "The evolving technology of classification-based knowledge representation systems", In J. Sowa, Ed. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Morgan Kaufmann, San Mateo, CA. 1991

[31] MacGregor, R. "Inside the LOOM classifier", *SIGART bulletin*, June 1991, 2(3):70-76

[32] D. L. McGuinness, "Description Logics Emerge from Ivory Towers", *the Proceedings of the International Workshop on Description Logics*. Stanford, CA, August 2001.

[33] D. L. McGuinness, "Ontologies come of age", in *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.

[34] R. Mizoguchi, M. Ikeda, K. Seta, and J. Vanwelkenhuysen, "Ontology for Modeling the World from Problem Solving Perspectives", *Proc. of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995, pp. 1-12

[35] E. Motta, "Reusable components for knowledge modeling", IOS Press. Amsterdam, 1999

[36] N. F. Noy, D. L. McGuinness, "Ontology Development 101: A guide to creating your first Ontology", SMI technical report SMI-2001-0880 (2001), Stanford University

[37] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, & M. A. Musen. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems,* 2001, 16(2):60-71

[38] W. Shen, D.H., Norrie, J.P. Barthes, *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor and Francis, London, UK, 2001

[39] W. Shen, A. James, K-M. Chao, M. Younas, Z. Lin, J-P. Barthes, (Eds.) *Proceedings of the Ninth International Conference on CSCW in Design,* Vol. 1&2, Coventry, UK, 2005

[40] W. Shen, T. Li, Z. Lin, J-P. Barthes, (Eds.) *Proceedings of the Eighth International Conference on CSCW in Design*, Vol. 2, Xiamen, China, 2004

[41] R. Stevens, C. A. Goble, S. Bechhofer, "Ontology-based knowledge representation for bioinformatics", *Briefings in Bioinformatics*, 2000, 1(4):398-416

[42] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke. OntoEdit: Collaborative Ontology Engineering for the Semantic Web, *Proceedings of the International Semantic Web Conference 2002 (ISWC 2002),* June 9-12 2002, Sardinia, Italia., Italy

[43] B. Swartout, P. Ramesh, K. Knight, T. Russ, "Toward Distributed Use of Large-Scale Ontologies", *Symposium on Ontological Engineering of AAAI*. Stanford, California, March, 1997

[44] M. Uschold, M. Gruninger, "Ontologies: principle, methods, and applications", *Knowledge Engineering Review*, 1996, 11(2), 93-136

[45] M. Uschold, "Knowledge level modelling: concepts and terminology", *Knowledge Engineering Review*, 1998, 13(1)

[46] M. Uschold, R. Jasper, "A framework for understanding and classifying ontology applications", *Proceedings of the IJVAI-99 workshop on ontologies and problem solving methods (KRR5)*, Stockholm, Sweden, August, 1999, pp 11-1—11-12

[47] http://apollo.open.ac.uk/

[48] http://dublincore.org/

[49] http://img.cs.man.ac.uk/oil/

[50] http://www.ontoprise.de/co_produ_tools.htm

[51] http://www.w3c.org/RDF/

[52] http://www.w3.org/TR/2002/WD-owl-ref-20020729/.

[53] http://www.w3.org/TR/daml+oil-reference

[54] http://www.w3.org/TR/rdf-schema/
[55] http://www.w3.org/XML/Schema/