J. Serv. Sci. & Management, **2008, 1: 199-205**
Published Online December 2008 in SciRes (www.SciRP.org/journal/jssm)

Improving Engineering Data Management with Semantic Web Techniques

Kai Wang

Mathematics Department, Guizhou University
Email: mathsfan@hotmail.com

Received August 27th, 2008; revised December 15th, 2008; accepted December 22nd, 2008.

ABSTRACT

Throughout the IT communities, it has been acknowledged that ontology plays a key role in representation and reuse of knowledge. This paper discusses some issues of ontology construction for engineering data management, such as knowledge discovery, knowledge representation and semantic services. All discussions are followed by a running example of engineering data management. Based on the user needs with five phases of engineering design, the issue of knowledge discovery will be presented. The built knowledge base contains basic knowledge models and vocabularies for knowledge integration. With the semantics of semistructured data, the hierarchical relations of concepts in engineering data have been extracted for reuse in future engineering design based on some clustering techniques of data mining. Semantic services for engineering design will be provided with the ontology-based schema.

Keywords: *engineering data management, semistructured data, ontology, semantic web*

1. Introduction

Nowadays, throughout the IT communities, it has been acknowledged that ontology plays a key role in representation and reuse of knowledge. However, there is no so far general methodology for a domain ontology construction [1]. For reuse and representation of knowledge in engineering design, one may consider to use ontology as an unified knowledge model for knowledge representation and vocabularies.

In this paper, a methodology of ontology-based schema for engineering data management (EDM) will be discussed. This paper presents an ontology-based methodology aiming at reuse and representation of knowledge in engineering design. The built knowledge base will consist of basic knowledge models and vocabularies. The ontology-based schema provides formally defined semantics for capturing and reusing of design knowledge. With the machine-interpretable, flexible data structures that can be modified at run time, ontologies will provide a general way to knowledge representation and reuse in the EDM systems. For instance, semantic searching services will be provided among the heterogeneous engineering databases.

The rest of the paper is organized as follows. In Section 2, we present a brief overview of the state of the art for engineering data management. In Section 3, the user needs of an engineering data management are presented. Section 4 discusses the knowledge representation in engineering data management. In Section 5, some facets of ontology-based services will be discussed. In Section 6, the conclusion and future works will be discussed.

2. Related Work

2.1. Background

For ontology construction of engineering design, we need at first to represent a domain ontology in engineering design. With the integrated knowledgebase, semantic services will be provided with the ability to detect and eliminate inconsistencies in heterogeneous sources [2]. In literature, there are many standards of a construction project, including the phases in construction [2]. In general, these phases can be divided into three parts: planning; programming; design. In this paper, based on a five phases of design, we present a domain ontology of engineering design.

EDM (Engineering Document Management) systems support many facets for maintaining engineering ranging data from schema design to documentation stage [3]. In industry practice, EDM systems are typically embedded in PDM (Product Data Management), PLM (Product Lifecycle Management), and CAE (Computer Aided Engineering). PDM systems provide handing detailed product information, ranging from design to production stage. PLM systems integrate information on CAM systems and CAD systems, and the information about ERP (Enterprise Resource Planning) processes. However, traditional PDM/PLM systems suffer from the inability of reusing of the creative design knowledge, particular in the conceptual design stage. CAE (Computer Aided Engineering) systems also suffer the same deficiencies as PDM/PLM

systems, lacking of the well-structured product representation, especially in the early design phases. Like PDM systems, CAE systems fail to support more complex development activities. Thus, reusing of product knowledge has been hindered by the lack of capabilities to knowledge representation. Furthermore, the capabilities for direct process support are also hindered.

In [4,5], the knowledge representation (KR) formalisms of Quillian's was proposed as the semantic memory models. As a knowledge representation method, ontology represents the relevant domain entities and their relationships by means of classes and relations. Ontologies provide a unified knowledge model and vocabulary for knowledge integration in many specific domains. In general, ontologies provide machine- interpretable, flexible data structures that can be changed and adapted at run time [3,6,7]. In other ways, as XML provides an unified data exchange schema, the Web ontology language OWL [8,9,10] is a language for expressing sophisticated class definitions and properties. In [8,9], Tim Berners-Lee proposed for the formalization of the resources of web (no only the web page, but also all the data and resources on the web). The architecture of semantic web consists of uniform resource identifier at the lowest layer, XML and XML schema at upper layer and XML name space, constructing the fundamental syntax of semantic web.

2.2. Document Management System

Engineering Document Management systems(EDM) are important tools for maintaining engineering data [3]. EDM Systems such as Windream [11] or Documentum [12] are widely used in industrial practice for the storage, maintenance, and distribution of documents. A step further, Product Data Management (PDM) systems provide extended facilities for the handling of detailed product information, ranging from design to production stage. Product Lifecycle Management (PLM) systems is the successor of PDM. These systems are widely applied in the fields of industry manufacturing design. However, PLM/PDM systems lack essential capabilities to reflect the function, behavior, and structure for each design phase. Thus, the management and reuse of design knowledge are less suited, particular in the conceptual design stage.

In [13,14,15,16,17,18], the semistructured model was proposed for the data model and algebra of XML.

As our running example of document management system, the semistructured data model will be used for the query and navigating the engineering data, In practice, the running example offers typical services such as version management, change management, notification (if some changes have been committed), and simple navigation and retrieval functionality, shown as Figure 1. In our running example of EDM, the database is modeled in terms of labeled, directed graph, a semistructured data model. For improving efficiency, An indexing structure has also been provided, where the indexing data with its address are organized in double-level with hierarchy. For example, when a date data is indexed, the year and month data at the first level are stored, meanwhile, the date data is stored at the second level. At running time, the year and month data will be searched at first level. The second searching will be completed within the first matching results. The implementation aspects include some techniques in C++ programming, such as the multiple- user interface, P2P and components implementation in distributed environment. As a consequence, irregular engineering data can be accommodated in this model. In the sequel, the running example offers traditional services, such like query and navigating the data for five stages (introduced in the earlier section) in distributed multiple-user environment, meanwhile, a data object is a collection of attributes without blank-value.

3. User Needs

So far, functional modeling research within engineering design theory studies the flows-the materials, energies, and signals on which functions operate [19,20,21]. For functional modeling, domain knowledge conce- ptualization is critical to ontology construction.

A general knowledge systems methodology of ontology engineering for creating engineering ontology can be found in literature [22], including four typical steps as follows:

- Step 1–Determine the ontologies for engineering data
- Step 2–Enumerate important terms and rules in the ontology
- Step 3–Define the relationship within the ontology, including hierarchy and the properties of classes
- Step 4–Create instances for engineering data

Analysis of the user needs of engineering data management is critical to ontology construction for engineering data management. As an ontology-based methodology, we

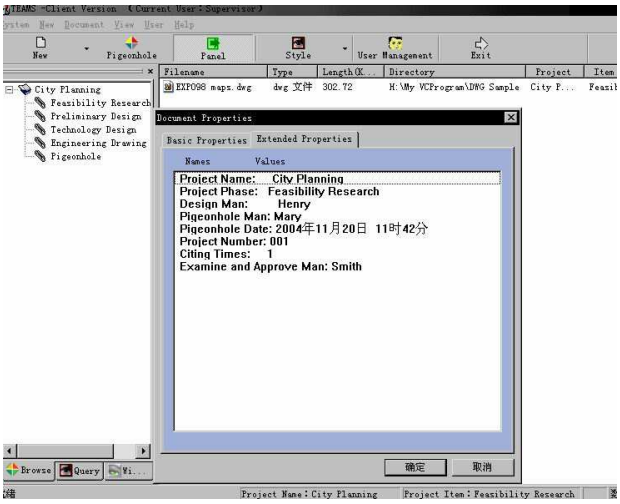

Figure 1. A running example of EDM

should establish our controlled vocabulary (terms) at first. In the running example of EDM system, there could be several kinds of ontologies for different purposes, e.g., design ontology, documentation (pigeonhole) ontology, citing ontology, etc. As machine-interpretable, flexible with data structures ontologies can be modified at run time.

As the proposed paradigm to knowledge integration for engineering design, we at first investigate the domain semantics for engineering design.

There are different phases in the domain of engineering data management [2]. In general, these phases are divided into three parts: planning; programming; design. In this paper, without loss of generality, we consider five phases in the engineering design, e.g., (1) Feasibility Analysis, (2) Preliminary Design, (3) Conceptual Design, (4) Schematic Design, and (5) Documentation, shown in Figure 3. In each phase, the vocabularies are defined. At each stage, one need approving services once specific works completed.

• Step 1−Feasibility Study Phase

In this phase, some methods and techniques are applied to examine the technical feasibility with the cost evaluation in a project, which proceeds to Conceptual Design phase once Feasibility Study has been approved.

• Step 2−Conceptual Design Phase

The goal of the conceptual design phase is to identify very general types of solution. Based on the market requirements and the state-of-the-art of the relevant technology, the degree of innovation, and the scope for innovation in a design project within a product are determined. The project proceeds to Schematic Design phase once Conceptual Design has been approved.

• Step 3−Schematic Design Phase

The goal of the conceptual design phase is to identify the schematic solution of the project. Schematic Design establishes a general scope, a conceptual design, scale and relationships among the components of the project. The primary objective is achieved with a clearly definition, feasible concept. A series of rough plans, known as schematics, should be prepared. In practice, models may be prepared to help visualizing the project. The project proceeds to Design Development phase once Schematic Design has been approved.

• Step 4−Design Development Phase

This phase is to develop more detailed design with the other aspects of the proposed design. The detailed design results will served as the achievements of engineering development. The project proceeds to Documentation Management phase when Schematic Design has been approved.

• Step 5−Documentation Phase

Once the Design Development phase had been approved, the detailed works, as well as the documents in previous stages should be documented into the library for future citation and/or reuse.

One could illustrate five phases in the engineering design as shown in Figure 2, meanwhile, the engineering document process can be defined as the procedure shown in Figure 3.

As mentioned in the literature, the user needs of a domain ontology are served as the skeleton of a domain ontology. Based on the user needs of a domain ontology, the syntactic elements of an ontology, such as properties, vocabularies and their relationships, are obtained. Abstract concepts can be formed with some clustering techniques, such as formal concept analysis (FCA). Rules with respect to properties and vocabularies can also be formed primitively. In the sequel, knowledge about refined concepts and their relationship is learned statically or dynamically with earlier mentioned models or an iterated process.

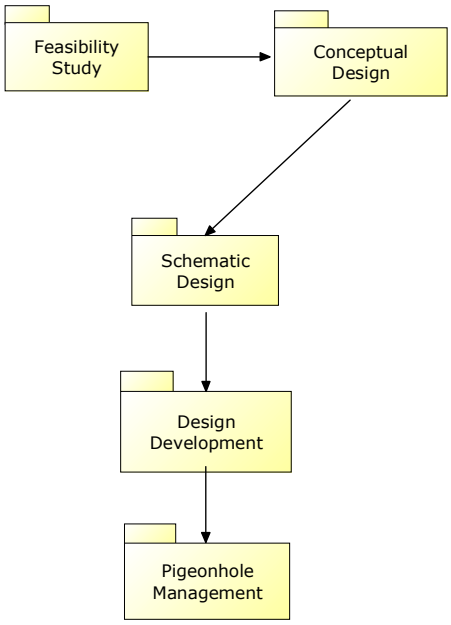

Figure 2. Overview of engineering design process

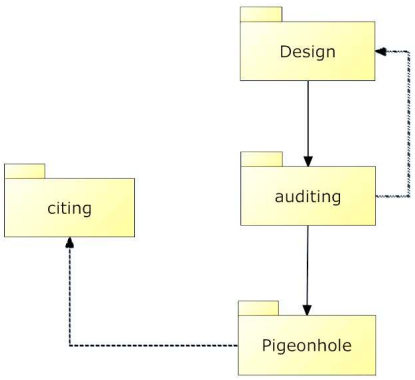

Figure 3. Overview of documentation process

4. Knowledge Discovery in EDM

In literature, knowledge discovery is the process of finding novel, interesting, and useful patterns in data. Domain conceptualization aims at identifying and defining concepts, specifying relationships among the concepts. Capturing domain concepts and their relationships is a bottleneck for ontology engineer [1]. One could capture concepts and their relations automatically by some clustering techniques, such as formal concept analysis (FCA). Then knowledge discovery and semantic services for engineering design will be provided.

For knowledge reuse and semantic services in engineering data management, knowledge discovery is crucial for a knowledge management system. However, in industrial practice, people are reluctant to share knowledge by making their knowledge explicit. One of the reasons for this is that it requires extra effort and they seldom hardly benefit from the explicit knowledge. Meta information about products is often chosen inconsistently and, similarly, work processes and decision making procedures are captured by different users in different ways. Consequently, knowledge acquisition is severely limited and could be found by some query. A solution to these problems is to automatic knowledge acquisition, called automatic knowledge discovery. The current ontology learning systems extracts relevant domain terms and relationships from a corpus of text. As an unsupervised learning method, formal concept analysis is a good tool to extract concepts and implied rules. With formal concept analysis (FCA), one can extract formal concepts and their relationships fro a boolean context. The basic relation of the concepts is the hierarchical relation, which is the core component of all ontologies.

Formal concept analysis (FCA) is a clustering technique which concerns about the hierarchical grouped structure of objects, which is a good start point for concept construction [23]. FCA reflects the philosophical understanding of a concept as a unit of thought consisting in two parts: the extent containing all the entities belonging to the concept and the intent which is the collection of all the attributes shared by the entities, based upon Galois connections [2,23,24]. A boolean context C is defined as a triple (G,M,I), where G is the set of objects or entities and M the set of properties or attributes. The formal concepts set F can be derived from this context as follows. For $A \subseteq G$ and $B \subseteq M$, let us define the Galois connections:

$$A' = \{m \in M \mid (\forall g \in A) gIm\},$$
$$B' = \{g \in G \mid (\forall m \in B) gIm\}.$$

Thus A' is the set of the attributes that are common to all the entities in A and B' is the set of the entities possessing their attributes in B. These Galois connections verify all the usual properties of the duality in particular the following relation $A \subseteq A''$ stands for any subset of the context (consequently $A' = A'''$). Concept is then a pair (A,B) such that $A \subseteq G, B \subseteq M$, $B = A'$, and $A = B'$. A is the extent of the concept and B the intent. For concepts (A_1,B_1) and (A_2,B_2), the hierarchical subconcept superconcept relation is formalized by $(A_1,B_1) \le (A_2,B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1)$. The set of all the concepts of the context (G,M,I) together with this order relation is a complete lattice which is called the concept lattice Therefore for every set of concepts there exists a unique largest subconcept, the infimum and a unique smallest superconcept, the supremum. Supremum and infimum are respectively given by $\vee_{j\in J}(A_j,B_j) = ((\cup_{j\in J} A_j)'', \cap_{j\in J} B_j)$ and $\wedge_{j\in J}(A_j,B_j) = (\cap_{j\in J} A_j, (\cup_{j\in J} B_j)'')$.

In semistructured data model, such as Lorel and XML, the data model is a multi-labeled graph, where the data objects are nodes and the attributes, labels [2]. In our running example, a data object is a collection of attributes without blank-value. Thus, it is very easy to identify whether a data object has an attribute. By conceptual scaling, an engineering document can be transformed into a boolean context.

With standard FCA algorithms, the hierarchical grouped structure of engineering data objects, called concept lattice, can be formed. Then the collection of implications (association rules) will be discovered. This will provide useful and applicable knowledge integration for reusing the engineering data.

5. Knowledge Representation

In philosophy, ontology is a theory about the nature of existence. A formal definition of ontology used in this paper bases on [25]: "An ontology is a tuple $\Omega = (C,R,\psi)$, whereas C is a set of concepts, R a set of relationship names, $R^H := "is\_a" \in R$ the inheritance relation on C, $R^A \in R$ the attribute relation on C and $\psi : R \to \wp(C \times C)$ a function". In short, an ontology is the conceptualization of concepts and their relations.

There are many kinds of representation of ontology. For instance, a typical ontology may include the conceptual entity of the domain, called concept, attributes describing a concept, called property, relationship between concepts, called relation, relation via logic expression, called axiom. Ontology represents the interrelationships between entities or resources, while XML and RDF deal with metadata. An ontology is typically defined as a specification of conceptions, where there are definitions of entities and their relationship with each other, based on semantic nets used in artificial intelligence.

One can discover a domain knowledge through supervised or unsupervised methods. As mentioned earlier, FCA is an unsupervised method to forming abstract concepts and rules. In issues of ontology learning, one can construct an ontology for engineering design automatically. However, the resulted ontologies are often unsatisfiable without human intervention. In this paper, we present an example of knowledge representation for our running example of EDM, shown in Figure 4. In the engineering knowledge domain, there are at least two kinds of entities, persons and documents. The former includes the attributes as name, email and can be divided into deviser, document managers and auditors, etc; the latter includes various attributes of documents, such as visiting time, documentation time, file location, data size, data type, etc. The relationships include hierarchical attributes, etc.

Then we attempt to represent the knowledge in our running example of EDM as axioms of DLs. DLs are the historical descendants of attempts to formalize semantic networks, which can be addressed by utilizing a subset of first order logic [23]. A DL-based system can support modeling of rules for engineering data in a hierarchical fashion. A typical DLs knowledge bases consists of a TBox calT, an ABox $\sum$ (a set of assertions), and a set of rules Ř. The set of rules Ř is of the form $C \Rightarrow D$, where C, D are concepts. Given an individual a (a documented document or person), a new assertion $D(a)$ will be added into the ABox $\sum$ if $C(a)$ is already believed to hold. The lower expressivity of description logics, e.g., $AL\varepsilon N$, limits the ability to define detailed semantics for a domain. However, in return they make several positive tradeoffs, including desirable computability and tractability results [23]. Within the repository application, this loss of expressiveness is acceptable so long as enough design knowledge can be captured to enable necessary operations, such as matching devices against adequately sophisticated searches and comparing mechanisms.

We denote by the set of atomic concepts in $\sum$, $Symbols_T$ [23]. Now let us demonstrate some axiomatic representation with respect to the model shown in Figure 4.

In graph-based presentation of ontology, the atomic concept is often presented as a class. In the ontology of engineering design below, the ABox $\sum$ consists of the assertions with respect to the documented documents and persons, and

$$Symbols_T = \{Person, Document\_Manager, Designer, Auditor, Document\} \quad (1)$$

Besides the inheritance relations, the properties (slots) of concepts could also be described as roles. Thus, the set of atomic roles contains

$$\{hasName, hasEmail, desgin, audit, document, designBy, auditBy, \quad (2)$$

$$documentBy, cite, hasLocation, \quad (3)$$

$$hasType, hasSize, hasDocumentationtime\} \quad (4)$$

T consists of the following axioms

$$Person \sqsubseteq \exists hasName.\top \sqcap \exists hasEmail.\top \quad (5)$$

$$Designer \equiv Person \sqcap \exists desgin.Document \quad (6)$$

$$Auditor \equiv Person \sqcap \exists audit.Document \quad (7)$$

$$Document\_Manager \equiv Person \sqcap \exists document.Document \quad (8)$$

$$Document \sqsubseteq \exists isCited.\top \sqcap \exists hasLocation.\top \sqcap \exists hasType.\top \sqcap \exists hasSize.\top \quad (9)$$

$$\sqcap \exists hasDocumentationtime.\top \quad (10)$$

We provide some rules for the ontology above, where the rules set Ř contains

$$\exists desgin.Document \Rightarrow Designer$$

$$\exists audit.Document \Rightarrow Auditor \quad (11)$$

$$\exists document.Document \Rightarrow Document\_Manager \quad (12)$$

$$\exists cite.Document \Rightarrow \exists isCited.Document \quad (13)$$

6. Semantic Search Service

After the preliminary ontology construction for engineering data, further steps, e.g., integrating all concepts and relations in distributed environment and evaluating the on-

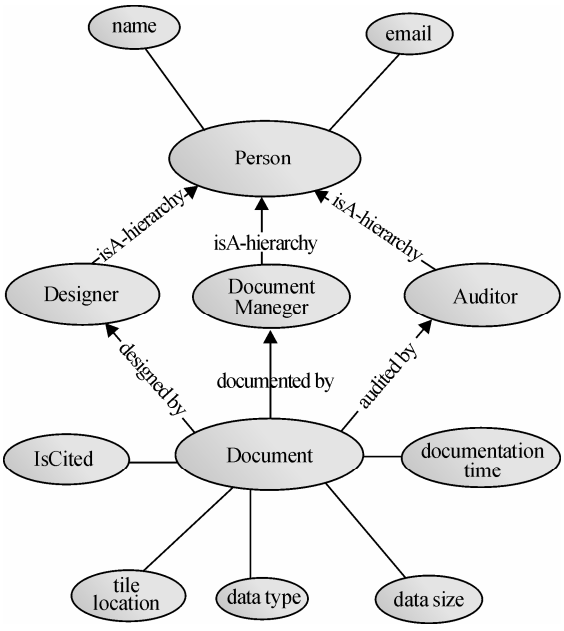

Figure 4. An ontology example of engineering design

tology, should be investigated. As mentioned earlier, the ontology-based semantic services should include constructing domain concepts and their relationships. Furthermore, a sophisticated search and comparing mechanism, which are advanced semantics services, should also be provided. One can demonstrate some possible DL-based semantics services as follows:

1) Asserting new information about an existing term (individual)
2) Recognizing that the updated term is an instance of a class (concept name)
3) Firing a rule on the term that is associated with the class
4) Propagating information from the updated term to related terms

One can offer inference service to reason with DLs-based knowledge. For example, given an individual a (a documented document or a person), one can find the most specific concepts for a e.g., the most proper description of a within the domain. One can also provide other kinds of intelligent services, such knowledge discovery, semantics searching, etc [19].

In DLs, given an ontology $\sum$, a query ϕ could be written in the form of $C(a)?$, where a is an individual and C, a concept description. It could also be expressed as an evaluation of the consequence relation $\sum \models \phi$, with the only two answers to that query: either 'yes', or 'no'. For example, in the ontology above, the problem whether the *Designer* are *Auditor* are same for the document *Doc* can be expressed as the entailment problem:

$$\sum \models \exists designBy.(Designer \sqcap Auditor)(Doc) \qquad (14)$$

7. Conclusions and Future Works

As mentioned above, we discuss some issues of ontology-based schema for engineering knowledge systems. Based on the user needs of engineering data management systems, we discuss the issues of knowledge discovery method with formal concept analysis. The knowledge representation of engineering data management systems has also been discussed. Furthermore, we have investigated the domain semantics. The proposed works could be served as major steps for constructing an engineering data knowledge management system.

However, many problems are still open. For example, it is still challenging to meet the needs of providing far more semantics services for creative design processes in engineering data management. In all, creative engineering data management is attractive but challenging topic. As knowledge discovery is currently the major bottleneck of knowledge-based system, one needs more effective and efficient methods to extract relevant domain terms automatically. Appropriate concepts can meet the requirements of engineering data knowledge management system and describe the domain knowledge properly.

REFERENCES

[1] L. Ling, Y. Hu, X. Wang, and C. Li, "An ontology-based method for knowledge integration in a collaborative design environment," The International Journal of Advanced Manufacturing Technology, Volume 34: pp. 843–856, 2007.

[2] S. D. Miller, "A control-theoretic aid to managing the construction phase in incremental software development (extended abstract)," In Proceedings of the 30th Annual international Computer Software and Applications Conference (Compsac'06)–Volume 02.

[3] M. Uschold, M. King, S. Moralee, *et al.*, "The enterprise ontology," Knowl Eng Rev 13(1): pp. 31–89, 1998.

[4] A. M. Collins and M. R. Quillian, "Retrieval time from semantic memory," Journal of verbal learning and verbal behavior 8 (2): pp. 240–248, 1969.

[5] A. M. Collins and M. R. Quillian, "Does category size affect categorization time?" Journal of verbal learning and verbal behavior 9 (4): pp. 432–438, 1970.

[6] Description Logics. Baader, Franz, Horrocks, Ian, Sattler, and Ulrike, Volume, Handbook on Ontologies in Information Systems of International Handbooks on Information Systems, chapter I: Ontology Representation and Reasoning, pp. 3–31. Steffen Staab and Rudi Studer, Eds., Springer, 2003.

[7] Description Logics. Baader, Franz, Horrocks, Ian, Sattler, and Ulrike, Volume, Handbook on Ontologies in Information Systems of International Handbooks on Information Systems, chapter I: Ontology Representation and Reasoning,. Steffen Staab and Rudi Studer, Eds., Springer, pp. 3–31, 2003.

[8] T. Berners–Lee, "Semantic Web-XML2000,".

[9] T. Berners–Lee, J. Handler, and O. Lassila, "The semantic web," Scientific American, Vol. 184, pp. 34–43, 2001.

[10] P. F. Patel-Schneider, P. Hayes, and I. Horrocks, "OWL web ontology language semantics and abstract syntax, W3C Recommendation 10 February 2004", http://www.w3.org/TR/owl-semantics/, 2004.

[11] H. G. Windream (2006), Managing documents by windream," Available at http://www.windream.com/. Accessed October 2006.

[12] EMC2, (2006). Documentum. Available at http://software.emc.com/products/product family/documentum family.htm. Accessed September 2006

[13] S. Abiteboul and D. Quass, "The lorel query language for semistructured data," International Journal on Digital Libraries, pp. 68–88, 1997.

[14] D. Beech, A. Malhotra, and M. Rys, "A formal data model and algebra for XML, W3C. XML Query working group note, September 1999.

[15] M. Fernandez, J. Simeon, D. Suciu, and P. Wadler, "A data model and algebra for XML query," 1999.

http://www.cs.belllabs.com/wadler/topics/xml.html#algebra.

[16] B. F. Cooper, N.Sample, M. J. Franklin, G. R. Hjaltason, and M. Shadmon, "Fast index for semistructured data," Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.

[17] L. Cardelli, "Describing semistructured data," SIGMOD Record, Vol. 30, No. 4, December 2001.

[18] Y. Papakonstantinou, "Enhancing semistructured data dediators with document type definitions," ACM SIGMOD ICDE, pp. 136−145, 1999.

[19] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and Peter Patel−Schneider, editors. The description logic handbook: Theory, implementation, and applications," Cambridge University Press, 2003.

[20] J. Kopena and W. C. Regli, "Functional modeling of engineering designs for the semantic web," IEEE Data Engineering Bulletin, 26(4), pp. 55−61, 2003.

[21] Y. Kitamura and R. Mizoguchi, "Ontology-based description of functional design knowledge and its use in a functional way server," Expert Systems with Applications, 24(2), pp. 153−166, 2003.

[22] Angele, Staab, R. Studer, and D. Wenke, "OntoEdit: Collaborative ontology engineering for the semantic web," International Semantic Web Conference 2002 (ISWC 2002), Sardinia, Italia, 2003.

[23] B. Ganter and R. Wille, "Formal concept analysis: mathematical foundations," Springer, Heidelberg, 1999.

[24] A. Hotho and G. Stumme, "Conceptual clustering of text clusters," In Proceedings of FGML Workshop, pp. 37−45, Special Interest Group of German Informatics Society (FGML-Fachgruppe Maschinelles Lernen der GI e.V.), 2002.

[25] Faatz, Andreas, Steinmetz, and Ralf, "Ontology enrichment evaluation," Lecture Notes in Computer Science. Springer−Verlag. Vol. 3257. pp. 497−498, 2004.

