1ST IFIP TC-5 WORKING CONFERENCE ON CAI
IFIP-TC5 ULM, GERMANY. NOVEMBER 14-15 2005

COMPUTER-AIDED INNOVATION WITH THE SEMANTIC WEB

Steve Beaty, Iliya Georgiev, Gearold Johnson

Abstract

Information technology innovation is a key challenge faced by many organizations. One way to innovate is to integrate different resources and data and to configure new functionality of the information system. A promise of applying Semantic Web technologies to enable data integration of disparate data sources based on meaning of data, not just the data themselves. We consider how Semantic Web technologies may support the design of innovative computing infrastructures and how to build ontologies to facilitate this innovation. This paper presents an approach for using ontological engineering in creation of innovative ideas. Both top-down and bottom-up approaches are considered in this creation. We motivate using the Semantic Web for computer-aided innovation using a simple problem space, but the ideas easily generalize to many problem domains.

Keywords: ontologies, innovation process, Semantic Web, bottom-up and top-down approach of ontology creation

1. Introduction

"The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." [1]. Another way of stating this is that the Web displays information to people, the Semantic Web presents data in a structured form to computers and allows reasoning on those data.

Designers of computer-aided systems are beginning to implement Semantic Web technologies in many different domains. A key challenge faced by this effort is the computer-aided data integration within and across design boundaries. Understanding the design models incorporated in heterogeneous systems is the opportunistic way to accelerate the innovation process.

The innovation process often integrates several apparently disparate concepts to create novel solutions to existing problems. One approach to computerizing the innovation process is creating ontological models of the problem domain and reasoning about them with highly structured preexisting knowledge of possible approaches to a solution. Ontologies can give uniform understanding and access to the pool of design, manufacturing, and other models.

For this paper, we propose to structure both the modeling of the problem and the preexisting knowledge using techniques in place for adding machine-understandable information to the web. Systems will need to communicate with a variety of systems owned by others in order to properly search the solution space, and those systems will be queried in unexpected ways. This leads to efficiency, security, and intellectual property issues that must be addressed.

The paper is structured as follows. Section 2 highlights the Semantic Web and ontology engineering. This includes background necessary to develop ontologies, standards that have

to be followed and different approaches to develop and map across ontologies. Section 3 explains our idea to incorporate ontologies in web-based systems accessing the models of previous designs. We also compare both approaches of creating ontologies (top-down and bottom-up) giving arguments that the top-down approach is preferable for our approach. Section 4 presents one problem domain (mesoscale weather modeling) and how this approach might be applied to it. Section 5 describes some difficulties envisioned with this approach, section 6 briefly compares it to other approaches, and section 7 concludes.

2. Foundations

2.1 The semantic web

The web provides access to an enormous pool of information, but it was designed to be only human-readable. The Semantic Web, whose layers are on top of the current web, allows definition of semantics of this data precisely enough to make the data computer interpretable. This facilitates the use of software agents to better mine the created product models and filter the information supporting other upper-level information processing. This processing is beginning to include advanced reasoning techniques for more automated derivation and construction of information.

The Semantic Web expresses all data and their relationships in several XML-based languages. The basis of this information is the Resource Description Framework (RDF) [2] that "is a language for representing information about resources in the World Wide Web" and "can also be used to represent information about things that can be *identified* on the Web, even when they cannot be directly *retrieved* on the Web". Originally intended to provide metadata about Web pages (author, copyright information etc.), it has been used to express a rich variety of information that can be processed directly by computers instead by humans. RDF expresses the properties that resources can have, and values for those properties. Essentially, RDF allows statements in the form subject, predicate, object. For example, in "Steve has a daughter named Tessa", "Steve" is the subject, "has a daughter named" is the predicate, and "Tessa" is the object. Instead of using common English for each part, RDF uses Uniform Resource Identifiers (URI's) for each part.

While RDF provides a way to describe resources, it does not provide a common language (often called a *vocabulary*) for different entities to share resources descriptions. For example, if one entity used "daughter" but another used "parent" to describe relationships, communication and reasoning between them would be difficult. The RDF schema (RDFS) language [3] provides the ability to describe the descriptions, i.e.: for multiple entities to agree on how they will create the RDF that describes their similar resources. RDFS allows the definition of classes and subclasses of resources. For example, one can create a class of "Animals", and subclasses of "Mammals", "Insects", etc. This is similar to a class hierarchy in an object-oriented (OO) programming language. It deviates from type systems in OO languages in that a resource can present in multiple classes, whereas in most OO languages resources (variables) reside in a single class. As with some OO languages, RDFS allows for multiple inheritance.

Now that we can consistently describe resources, we would like to reason about them. The Web Ontology Language (OWL) [4] provides that capability. OWL adds constraints to resources, allowing automated reasoning. There are three varieties of OWL: Lite, DL, and Full. OWL Lite adds only simple (*cardinality*) constraints and only two values (0 or 1, true or false) for those constraints. OWL DL assures that the system of rules created are

computationally complete and decidable (i.e.: all derivations are computable and finish in a finite amount of time). DL stands for Description Logics, a well-understood area of automated reasoning. OWL Full allows for the richest set of constraint descriptions, but is not guaranteed to be computable in finite time. OWL is derived form another effort to add ontological information: the DARPA Agent Markup Language (DAML) [5] and its extension DAML+OIL [6].

One proposal for reasoning about OWL information is the Semantic Web Rule Language (SWRL) [7] "Horn-like" rules (clauses) are added to OWL's axioms to allow reasoning. Horn clauses are well known in the artificial intelligence community and consist of a set of conjunctive propositions (axioms AND'ed together) and a consequent. The programming language Prolog is based on Horn clauses. Tools exist to reason using SWRL

All of this then allows us to express that resources (of any different type) exist in a consistent way and reason about those resources. With this, we propose the Semantic Web can be used as an excellent adjunct to aid in innovative solutions to many domains.

Let us look briefly at some examples. First, here is part of an RDFS description of wines:

```
<rdf:RDF
xmlns="http://ontolingua.stanford.edu/doc/chimaera/ontologies/
wines.daml#"  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#">
   <rdf:Property rdf:ID="COURSE">
      <rdfs:domain rdf:resource="#MEAL"/>
      <rdfs:range rdf:resource="#MEAL-COURSE"/>
   </rdf:Property>
   <rdf:Property rdf:ID="GRAPE-SLOT">
      <rdfs:range rdf:resource="#WINE-GRAPE"/>
      <rdfs:domain rdf:resource="#WINE"/>
   </rdf:Property>
   <rdf:Property rdf:ID="FOOD">
      <rdf:type
rdf:resource="http://www.daml.org/2001/03/daml+oil#UniquePrope
rty"/>
      <rdfs:range rdf:resource="#EDIBLE-THING"/>
      <rdfs:domain rdf:resource="#MEAL-COURSE"/>
   </rdf:Property>
   <rdf:Property rdf:ID="SUGAR">
      <rdf:type
rdf:resource="http://www.daml.org/2001/03/daml+oil#UniquePrope
rty"/>
      <rdfs:range rdf:resource="#WINE-SUGAR"/>
      <rdfs:domain rdf:resource="#WINE"/>
   </rdf:Property>
[…]
<rdf:Description rdf:ID="CHATEAU-D-YCHEM-SAUTERNE">
      <rdf:type rdf:resource="#SAUTERNE"/>
      <GRAPE-SLOT rdf:resource="#SAUVIGNON-BLANC-INDIVIDUAL"/>
      <GRAPE-SLOT rdf:resource="#SEMILLON-INDIVIDUAL"/>
      <FLAVOR rdf:resource="#STRONG"/>
      <MAKER rdf:resource="#CHATEAU-D-YCHEM"/>
```

```
</rdf:Description>
[…]
```

This RDF/RDFS description includes many aspects of wine and their relationship to each other. It is part of a tool that can help one choose what wine to go with what food etc. Here is a fraction of a SWRL file from [7]:

```
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#example1"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom  swrlx:property="hasParent">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom
swrlx:property="hasBrother">
      <ruleml:var>x2</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom  swrlx:property="hasUncle">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

It shows how a simple relationship property can be modeled in SWRL.

2.3 Ontology engineering

Ontology engineering deals with the presentation, development, mapping, maintenance and retirement ontology lifecycle.

Ontology development depends on many factors: the semantics of the domain, data presentation in databases or repositories, specific policies of data exposure, etc. Here again two approaches are dominating: top-down and bottom-up approach. In top-down approach ontology is created to describe the main characteristics of the domain or community of interest, based on the high level concepts within the domain. The bottom-up approach uses the existing data sources, develops ontologies specific to individual databases, and then continues to extend these ontologies toward a greater semantic level.

3. Web-based computer-aided innovation – a vision

3.1 The idea

Figure 1 is an illustration of our idea and understanding of web-based computer-aided innovation process.

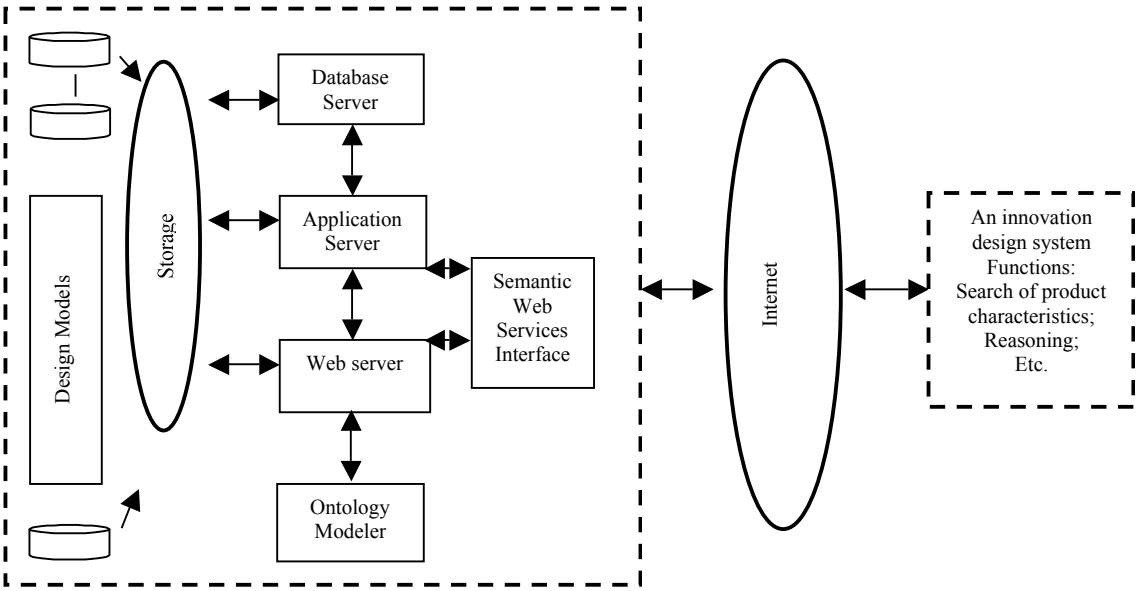

Figure 1

Servers that contain design models, product characteristics, etc., are connected to the Web and expose product data by Semantic Web services. Acting as a semantic translator, ontologies insulate each system from changes occur on the other side of the interface. A new component in such systems is the ontology modeling software, used at all stages of ontology development, implementation, maintenance and replacement.

On the other side of the figure is where the design innovation occurs. This might be an upgrade of an existing CAE system or brand-new application for this specific purpose. A current example of such a tool is an ontology browser/prover such as SWOOP [8]. In the future, such a tool would connect to the Web and search for a wide variety of product models that might assist in the problem under study. It models knowledge so that it can be easily interchanged and reasoned about by innovation applications, for a well-defined region of interests (domain) explained as concepts, relationships and rules about their properties.

3.2 Ontology modeling approaches

Ontological modeling is a way of managing knowledge and semantic integration. It is especially suited for difficult functions such as comparing a model within an ocean of others. Ontology is ultimately concerned with the consistency of communication between systems (or people). In ontological modeling, ontologies are used to reconcile metadata standards, XML dialects, and databases access mechanism.

3.2.1 Top-down ontology development

Top-down approach forces designers, domain experts, and resource producers to think through the characterization of their particular domain and identify key properties and relationships of the products models. Reaching agreement across the domain is a challenge, but in the globalization era such semantically-rich domain ontologies provide a clear characterization of the domain, which can be subsequently used to map concepts across domains.

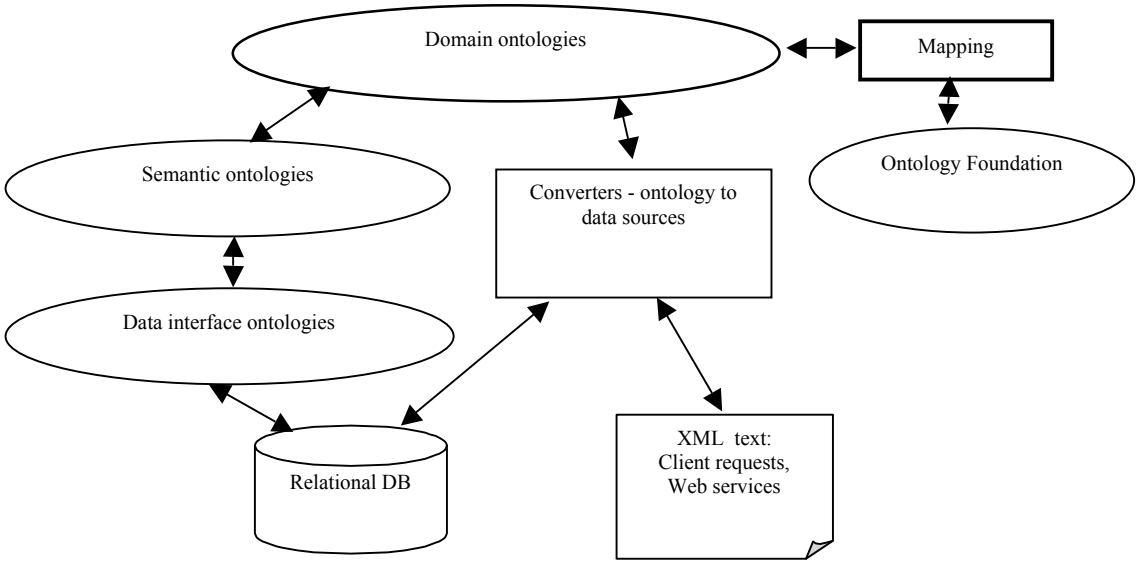

Figure 2

Top-down development (Figure 2) could leverage and reuse existing ontologies that have been developed by others. Such enterprise or open-source ontologies allow the domain and innovation designers to not only reuse existing ontologies, but also encourage more creative thinking through the representation of product domain, and will result in a more complete and concise ontological model.

The key difficulty with this approach is extending the product ontology such that lower concepts share a strong correspondence with data sources of the domain, while still maintaining model integrity.

3.2.2 Bottom-up ontology development

This approach (Figure 3) ensures that data sources are accurately represented. Created ontologies are the backbone for the domain ontology. Since the lower level ontologies are uniform interfaces to the well proved data presentation, the bottom-up development would provide an accurate model of the domain. On the other side, inconsistency and database specifics are made visible in the higher level domain ontology. Finally, the resulted domain ontology may not provide a complete presentation of the domain and relationships within it, because the ontology is focused on the existing data specifications. Distributing the ontology development on separate ontologies, the ontology structure mapped close to the data and interfaces representation. This mapping is pretty natural and useful, as the ontology provided a cleaner interface of the data source. At that point, the semantic of the data were not yet exposed. The ontologies close to the information sources are the first layer. They serve as a common interface to extract instance data from the respective data sources. Creating such ontologies is extreme intellectual activity, but such approach has two strategic benefits. First, it eliminates the heterogeneity of the data access mechanism to the underlying data sources. Second, such ontologies are the platform to create lightweight semantically related web services processes. The second layer of the ontology hierarchy abstracts the information into more wide-ranging concepts. Lower level concepts can be associated into these higher level classes. Further, the third level of the ontology hierarchy provides high level of semantics and

allows defining the main characteristics of the domain. This layer could represent some meaning elements of the domain.

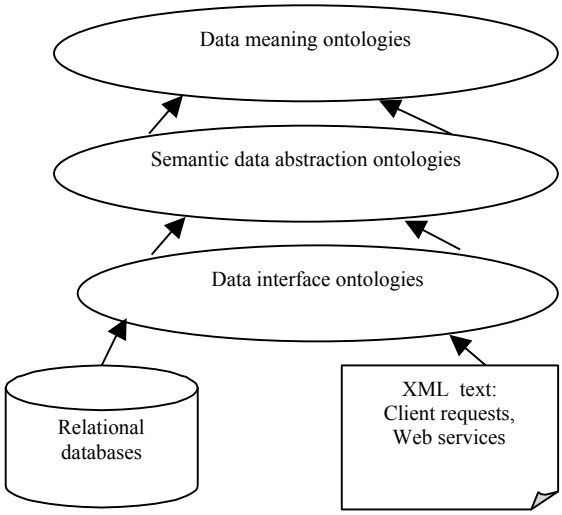

Figure 3

Quite a bit of work has been done in both top-down (creating ontologies form scratch to represent a particular domain) and bottom-up (starting with an existing data set and modeling it).

4. A motivating example

In order to motivate using the Semantic Web and ontologies for computer-aided innovation, let us turn to a specific problem domain that does not have a single, widely accepted solution: mesoscale (cloud scale) weather prediction. Business or government decisions related to weather are usually short-term (3 to 48 hours) but over a reasonably large area such as a coastal region, major city, etc. Such weather predictions might be required for applications such as weather related disaster warnings such as for hurricanes or tornados, emergency planning, agriculture, electrical or other energy allocation schemes, airport operations or other transportation activities, insurance, construction, communications networks, and many others. Weather predictions on this temporal and spatial scale require detailed calculations with a large number of grid points. For example, an area 300 km by 300 km using one km grid spacing and 40 vertical slices through the atmosphere requires 3,600,000 grid points. Some mesoscale weather prediction schemes use nested grids of decreasing spatial and temporal scales.

Obviously, the weather predictions must be produced in a timely fashion. Short-term predictions of less than three hours would require computational times in the minutes to be useful. A 24 to 36 hour prediction should be accomplished in less than an hour to be truly useful. Depending on the model grid size, temporal resolution, boundary and initial conditions, computational requirements can vary from 10 GFLOPs to 200 GFLOPs, up to 1000 Gbytes of memory, I/O bandwidth of 50 Mbit/sec, and local disk storage of one or more terabytes. Computer configurations that are being used for these types of computations are

large-scale vector supercomputers, large arrays of parallel processors, and distributed systems of supercomputers.

However, it could be that a large cluster could solve the problem in the necessary time with much less expense. Or a grid of non-dedicated computers. Or a cluster of high-performance but low cost game consoles based on an advanced processor [9]. Or a rack of graphics cards [10]. Or some heretofore unimagined combination of these, or other processing elements. While one cannot know a priori all the possibilities, if

1. organizations published their computing resource capabilities (including metrics such as FLOPS, MIPS, Mbit/S, etc.) in agreed-upon ontologies,
2. useful tools such as browsers and provers were widely used, and
3. users were educated in their use,

then we could truly see a revolution in computer-aided innovation in mesoscale weather prediction. SWRL can express many of the constraints and relationships necessary to examine and reason about possible solutions. For example, there is a simple constraint "`swrlb:greaterThan`" that can be used for FLOPS, MIPS, etc., and SWRL includes many mathematical and logical functions for combining constraints. Indeed, this type of innovation could extend to all computationally complex problems, and if the three conditions existed for other problem domains (electrical, mechanical, etc.), then all areas would benefit from this technology. Given the fact that all the supporting technologies exist and are in common use, little seems to stand in the way of just such a revolution occurring.

5. Related work

There have been done much research work in Semantic Web by ontologies of distributed computing. Since our idea and proposal is oriented to the specifics of the virtual innovation process, we cannot provide a comparison to these methods. For the same reason, it is also difficult to compare it to them. Nevertheless, several publications discussing ontology modeling have influenced our work.

Experience in prototyping of Semantic Web technology and explanation of challenges and corresponding research is given in [11]. Survey of the most relevant ontological modeling approaches is made in [12]. Several related work consider the top-down and bottom-up methods, some emphases that the bottom-up is practically useful for most application: [13] shares the argumentation, [14] explicitly declares that targeting completeness for the domain model appears to be practically unmanageable. Formal ontology framework is proposed in [15]. Additionally, our research is partly based on our previous work [16], [17].

6. Conclusion

Leveraging web semantic model, we propose a framework, which can help designers and programmers alike move more quickly from innovative idea to implementation. Even at such an early stage we can address serious difficulties and challenges.

- Creating ontologies is intellectual product of human beings and is a very difficult task. Ontologies that represent even just a concept become complex quickly. Ontologies evolve by default more slowly than human ideas, so there needs to be a systematic way to manage change.

- Ontological engineering requires declarative thinking and understanding of formal reasoning, and users will need to be educated in these.
- Ontological engineering is expensive. The creation of markup from unstructured text description of an innovation model is tedious and time-consuming. The good news is that there are several useful tools, which automatically generate markup.
- Ontology mapping and translation. Simply having a multitude of ontologies does not imply that intelligent and innovative knowledge search will occur. One ontology has to be translated or mapped to others even if the domain is only slightly different.

Ontological modeling and engineering should not be considered a panacea in intellectual areas such as innovation, but the field of ontology-related work is rapidly expanding due to its capability of producing context-sensitive knowledge discovery and delivery. In our paper we try to show that ontologies define how meaning can be embedded in models, and how these models can be used to create innovative solutions to problems.

While our example is from the computer systems design space, the techniques are general. Indeed, part of the power of this approach is that a lot of groundwork has been laid by the work on adding semantic information to the web. Leveraging this work in machine understanding for a different use might be considered the innovation of our work.

References

[1] Tim Berners-Lee, James Hendler, Ora Lassila. "The Semantic Web", *Scientific American*. May 2001.

[2] Frank Manola, Eric Miller, Editors, "RDF Primer", February 2004 http://www.w3.org/TR/2004/REC-rdf-primer-20040210

[3] Dan Brickley, R.V. Guha, Editors, "RDF Vocabulary Description Language 1.0: RDF Schema", February 2004, http://www.w3.org/TR/rdf-schema/

[4] Michael K. Smith, Chris Welty, Deborah L. McGuinness, Ediros, "OWL Web Ontology Language Guide", February 2004, http://www.w3.org/TR/owl-guide/

[5] http://www.daml.org/

[6] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein, "DAML+OIL (March 2001) Reference Description" http://www.w3.org/TR/daml+oil-reference

[7] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", May 2004, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

[8] Aditya Kalyanpur, Bijan Parsia, James Hendler "A Tool for Working with Web Ontologies", *International Journal on Semantic Web and Information Systems*, Vol.1, No.1, Jan-Mar 2005

[9] Dr. H. Peter Hofstee, "Cell Broadband Engine Architecture from 20,000 feet", http://www-128.ibm.com/developerworks/power/library/pa-cbea.html

[10] David Geer, "Taking the Graphics Processor beyond Graphics", *IEEE Computer* , September 2005.

[11] P. Kogut and J. Heflin. “Semantic Web Technologies for Aerospace”, *Proceedings of IEEE **A**erospace Conference*, 2003.

[12] L. Kalinichenko, M. Missikoff, F. Schiapelli, and N.Skvorzov, N. “Ontological Modeling”, *Proceeding of the 5th Russian Conference on Digital Libraries*, St.Petersburg, Russia, 2003.

[13] S. Salim, K. Hetherington-Young, and S. Frey, “Ontology Engineering: An Application Perspective”, *http://www.mitre.org/work/tech_papers/tech_papers_04/04_0847/*, The MITRE Corporation Publishing , 2004

[14] A. Maedche and S.Staab. “Ontology Learning for the Semantic Web”, *J. IEEE Intelligent Systems*, No.1, 2001.

[15] C. Pahl, and M. Casey. “Ontology Support for Web Service Processes”, *ACM SIGSOFT Software Engineering Notes***,** vol. 28 , issue 5, 2003, pp.208-216.

[16] I.Georgiev . “Ontology Modeling for Semantic Web-driven Application”, *Procedings of the International Conference Computer Systems and Technologies,* Varna, 2005, pp.192-197.

[17] I.Georgiev and J.Ovtcharova. “Modeling Web Services in a PLM N-tier Architecture”, *Proceeding of the International Conference on Product Lifecycle Management,* Lyon, France, 2005, pp.199-209.

Corresponding author: Steve Beaty, The Metropolitan State Collge of Denver, Department of Mathematical and Computer Science, Campus Box 38, P.O. Box 173362, Denver, Colorado, 80217-3362, USA, Phone: (303) 556-5321, FAX: (303) 556-5381, beatys@mscd.edu