NSF GRANT # 0523052
NSF PROGRAM NAME: DMI

A Method of Capturing Product Data Semantics By Building Engineering Ontology

Pei Zhan
Washington State University

Uma Jayaram
Washington State University

Sankar Jayaram
Washington State University

Abstract: The primary objective of this paper is to conceptualize and develop an approach to capture product data semantics for the integration of engineering applications in a design/analysis domain by building engineering ontologies and metadata. In this paper, we introduce our method of building ontologies, and using metadata to link product data to the concepts in ontologies. We present a method of capturing product data semantics to be utilized in a CAD/CAE integration framework. An example is also given to show how product data semantics is captured from applications in different domains. This is work in progress

1. Introduction

A design process is a collection of activities that usually focus on varied aspects of a product and is performed by people with different areas of expertise and using a variety of engineering application tools. Although each engineering software tool can solve the problem within its scope in an effective manner, when there is an inter-disciplinary task requiring collaboration between people with different areas of expertise and different tools, these tools do not talk to each other in a satisfactory manner. The data formats for these tools vary greatly and more important, engineers working in different domains tend to use a different point of view to interpret the product data. Hence the product data semantics generated in one application in a certain domain cannot be understood in applications in other domains.

We define product data semantics as "understanding about data of a product design from certain point of view". A product is considered from different viewpoints and perspectives, and different representations and terminologies may be used in these different viewpoints, hence there are different product data semantics. For example, consider an automobile interior design evaluation. One viewpoint may focus on ergonomic comfort and accommodation while another viewpoint may focus on assembly simulation for the dashboard. The representation of the product, the terminologies used, and the communication of the results of these evaluations vary greatly even though the underlying product is the same.

This paper will present a method of capturing product data semantics to be utilized in a CAD/CAE integration framework. The difference between the traditional CAD/CAE integration framework and our new CAD/CAE integration framework is that we concentrate on a semantic level integration, in contrast to traditional integration framework where information exchange is mostly limited to a pure data level and product data semantics generated during the design/analysis process are always discarded. In our approach, product data semantics are collected efficiently and translated from one point of view to another point of view in the new integration framework.

To implement the semantic level integration, product data semantics needs to be captured and exchanged. The first problem that comes up in semantic level integration is: how do we capture product data semantics? Our method is to build engineering ontologies for each specific domain and then create metadata about design data on top of it.

In this paper we introduce some of the preliminary work we have done to capture product data semantics for engineering design/analysis for integration of engineering applications at a more sophisticated semantic level to support these design activities. In addition to a general method of building ontologies, a layered structure of product data semantics is proposed to improve the flexibility and link the actual design database to the semantics. An application is developed to maximally automate the process.

In example scenario is also given at the end of the paper to illustrate the mechanism of capturing product data semantics.

2. Background

2.1 Capture Semantics Using Ontology

Product data semantics can be represented in some product information models, but it is still difficult to capture. The primary obstacle is the technical challenges of organizing and managing knowledge and the design challenges of a human-centered approach to build a useful and usable design semantics recording system[1].

As an important technology in AI to share and reuse knowledge, ontology has been proposed as a possible solution to represent semantics during the design process. Ontology can be regarded as "an explicit specification of conceptualization"[2], or it can be defined as "domain theories that specify a domain-specific vocabulary of entities, classes, properties, predicates, and functions, and a set of relationships that necessarily hold among those vocabulary items. Ontologies provide a vocabulary for representing knowledge about a domain and for describing specific situations in a domain." [3]

Ontologies are being used in several projects to represent conceptualizations of products and represent and capture the design semantics [4-8]. Also, in the integration of engineering applications, ontologies have been used in several projects to solve the interoperability problems including semantic clashes and multiple translators. In one instance, two approaches were developed: using a shared common ontology or using ontology as an interlingua[8]. In an integration framework that consists of heterogeneous applications which use different languages to represent data, the latter approach seems to be a better choice. As an example, PSL[9;10], an ontology language developed by NIST is used to represent ontologies in the manufacturing process which works as an interlingua to integrate a process modeling tool and a scheduling tool. Also a presentation of product data semantics PSRL [11] has been developed at the University of Michigan for the purpose of semantic interoperability. However, in most of the applications mentioned above, it is assumed that the semantics has already been built and the ontology merely acts as a translator.

Several projects have been developed to capture design intent/design semantics. MUG[12] (Multi-user Groups for Conceptual Understanding and Prototyping) developed at Drexel, presents an approach to collaborative authoring of design semantics within a multi-user 3D environment. The design semantics (ontologies) are described using DAML. Also, semantic web technology has been used in constructing design repositories with reasoning mechanisms[13;14] and product family modeling[15]. A conceptual understanding and prototyping (CUP) [16] is introduced to capture the design intent. It is used to record the design idea in a form of function-behavior-structure. These projects are targeted to capture general design semantics, not specific to any existing commercial tools in any particular domains, thus the problem of exchanging product data semantics is not emphasized.

Thus, the ability to capture design semantics is still limited: there isn't a consistent representation to describe design semantics through out the whole PLM process, for example, in the design analysis/evaluation domain. In addition, it is hard for product data semantics generated in different domains to be exchanged in a real-time manner. What we need is a systematical way to describe product data semantics from different perspectives related to a design, and also be able to exchange these semantics in an integration framework.

2.2 Methodology and Tools for Building Ontology

Ontology development is a sophisticated process which requires both the skills and knowledge in the given domain. Approaches for building ontology have been developed by several researchers. The process of building engineering ontologies for engineering design is summarized as [17]:

1) Identifying the taxonomies (referred to as the root concept of the taxonomy);

2) Identify existing taxonomies for each of the root concepts from the previous stages;

3) Creating taxonomies if no existing taxonomy was found;

4) Testing the taxonomies for the particular application

5) Building a thesaurus for the integrated taxonomy

6) Refinement of the integrated taxonomy

There have been several ontology representation languages. Recently, in the development of the semantic web[18;19], several languages were invented to represent information to be exchanged on the web such as RDF, RDFS[20-22] and OWL[23]. One of the advantages of RDF/RDFS/OWL is that they employ XML syntax so the data format is widely accepted for exchange. Together with the ontology languages, there are many other development tools supported globally by different research groups such as Jena[24] and Protégé[35].

Jena [24], an open source semantic web framework, includes API for developing ontology and a reference engine, provides a programmatic environment for ontology engineering using RDF/RDFS and OWL.

Protégé is "a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies." [25], Protégé also provides a set of API for building customized programs based on it.

3. Integrating Engineering Applications at a Semantic Level

In the product lifecycle management process, communication is one of the essential issues. During the integration of CAD/CAE, people with different expertise usually have different views on the same product, and hence their descriptions about the product also varies. Furthermore, there lacks a formal way to express these multiple views in order to exchange this semantic information with others. Normally during the communication, a combination of picture, video, animation together with text annotation is used to describe product information, which is sometimes ambiguous and may even lead to confusion.

In addition to using ontology to solve the problem of ambiguity and semantic clashes, we also use engineering ontologies as knowledge base to describe and translate product data semantics – for example, a) what are the functions and structure of this product, b) what is the result of the design analysis and c) how does relate to the model etc. Through proper integration of the software tools, the user of a commercial design analysis software should be able to exchange product data semantics with a commercial CAD software user and communicate his/her opinions about the design from an analyst's point of view. If there are some problems during the analysis, the analyst and designer should be able to talk to each other about the problems and what properties of the design need to be modified to solve them in their own "native form" without any misunderstanding. We refer to this level of integration as "semantic level integration".

To integrate at a semantic level, two problems need to be addressed:

1) How to capture product data semantics

2) How to exchange product data semantics

In this paper, we present our method to capture the product data semantics. We are still working on our solution to exchange product data semantics and that will not be presented in this paper.

4. Structure of Product Data Semantics

In our integration framework, product data semantics is represented in two parts: 1) knowledge base (ontologies) and 2) metadata. The knowledge base describes the basic knowledge in different domains using concepts, relations and axioms in engineering ontologies. Metadata uses the concepts in engineering ontologies and instantiating them by instances to link the knowledge to the actual product design data. Their relations can be represented in Ffigure 1.

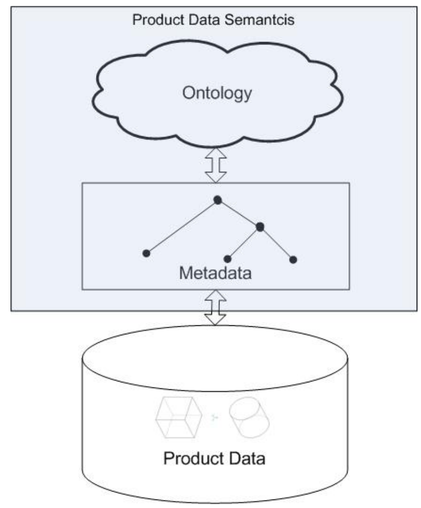

Figure 1

4.1 Structure of Engineering Ontologies

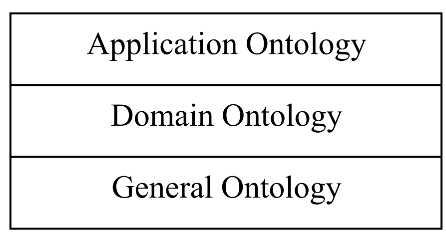

Figure 2

A layered structure [Figure 2] is used to build engineering ontologies in product design/analysis domain: the 3-tier ontology structure includes general product ontology, domain ontology and application ontology, each built upon the previous layer.

1) *General product ontology* describes basic elements that represent commonalities for the product, which can be applied to any domain in design//analysis. In this layer, the function-behavior-structure triple concept is used since this can be applied to any design/analysis

domains. The concepts in this layer are defined as below:
1) *ProductDesign* 2) *Function* 3) *Behavior* 4) *Structure.* Properties include *hasFunction*, *hasBehavior*, and *hasStructure* which describe the relations between *ProductDesign* and the other 3 concepts
[Figure 3] shows part of the RDF/OWL source code that defines the concepts and relationships.

```
<owl:Class rdf:ID="Function"/>
<owl:Class rdf:ID="ProductDesign"/>
<owl:Class rdf:ID="Behavior"/>
<owl:Class rdf:ID="Structure"/>
<owl:ObjectProperty rdf:ID="hasBehavior">
  <rdfs:domain rdf:resource="#ProductDesign"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFunction">
  <rdfs:domain rdf:resource="#ProductDesign"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStructure">
  <rdfs:domain rdf:resource="#ProductDesign"/>
</owl:ObjectProperty>
```

Figure 3

For the presentation and categorization of each of the concepts in the function-behavior-structure triple, many researchers have done important work [26-30] in addressing this. Readers are encouraged to read the referred papers if interested.
2) *Domain ontology* describes elements in a particular domain. Examples of domains are feature based modeling or product design ergonomics analysis. Each domain has concepts and relations described in this layer that are widely accepted by those working in this domain, but might be different from the concepts in other domain-specific ontologies. For example, in a feature based modeling domain the concepts of assembly, part, feature, parameter and constraint are commonly understood and used by the designer, regardless of what kind of feature-based CAD tools are used.
3) *Application ontology* describes elements in a specific application that work in a certain domain. Even though in the same domain, different applications use different terminologies and concepts, and their way of representing the structure of design are also different. For example, some feature concepts being used to describe structure of a product in CAD applications such as Pro/Engineer, are different from concepts and terminologies used in other CAD applications like Catia, Solidworks and Unigraphics, for example, concept BaseExtrude and BossExtrude in Solidworks are correspondent to the concept Extrusion in Unigraphics[11].

4.2 Structure of Product Metadata

Product metadata describes the structure of product data using predefined concepts and relations. Instances of concepts in the engineering ontologies are used as main components in product metadata to link product data to the concepts in the ontology layers. In Figure 4, a metadata is used to represent a simple part consists of two features made in Pro/Engineer [Figure 5]: a protrusion feature and a hole feature with their key dimensions.

In this metadata, concepts in Pro/Engineer application ontology are used:

ProeDesign:Hole,ProeDesign:Protrusion,ProeDesign:Hole_Depth,ProeDesign:Hole_Diameter,ProeDesign:Protrusion_Depth and their instances are created respectively: *Hole1, Block1, Hole_Depth1, Hole_diameter1, Block1_Depth*

In addition, the relations between the instances are also described in the metadata. such as:
has_feature to describe the ownership between *Part1* anf *Hole1*, *Block1;*
has_dimentions to describe the ownership between *Hole1* and *Hole1_diameter, Hole1_Depth*

```
  <ProeDesign:Hole rdf:ID="Hole1">
    <has_dimension>
      <ProeDesign:Hole_depth rdf:ID="Hole1_depth">
        <dimension_value
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
        >50.0</dimension_value>
      </Hole_depth>
    </has_dimension>
    <has_dimension>
      <ProeDesign:Hole_diameter rdf:ID="Hole1_diameter">
        <dimension_value
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
        >100.0</dimension_value>
      </Hole_diameter>
    </has_dimension>
  </Hole>
  <ProeDesign:Protrusion rdf:ID="Block1">
    <has_dimension>
      <ProeDesign:Protrusion_depth rdf:ID="Block1_Depth">
        <dimension_value
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
        >100.0</dimension_value>
      </Protrusion_depth>
    </has_dimension>
  </Protrusion>
  <ProeDesign:Part rdf:ID="Part1">
    <has_feature rdf:resource="#Hole1"/>
    <has_feature rdf:resource="#Block1"/>
  </Part>
</rdf:RDF>
```

Figure 4

There are many advantages of this metadata of product data including product data management, machine interoperability. Here in our CAD/CAE integration framework, one advantage of using metadata to describe product data is that it is very easy to present design alternatives. For example, an instance is created for the hole with 100 as diameter, while another instance of the same concept is created with diameter equal to 120. While all the rest of instances still can be reused, the new design can be differentiated by using another instance of concept *ProeDesign:Hole_diameter* with an alternative value. This will make the designer easily realize the semantics behind the two different designs.

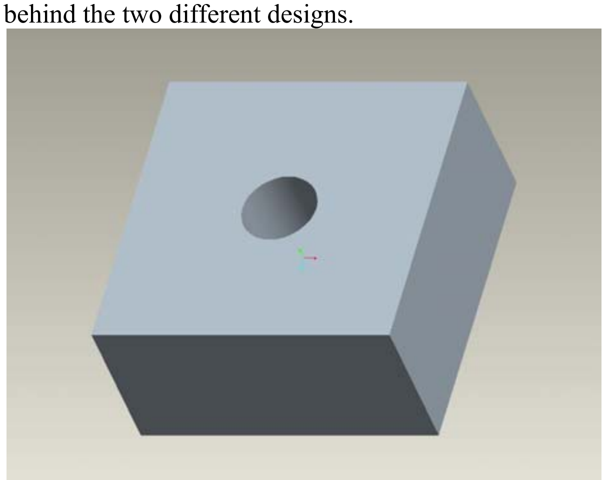

Figure 5

5. Capturing Product Data Semantics

5.1 Methodology of Capturing Product Data Semantics:

The process of capturing product data semantics can be regarded as a process of building product ontologies and instantiating concepts with instances. To capture product data semantics, and maximize the automation, following process methods are used to build the ontologies:

1) Building ontologies: Basic concepts about product data from different viewpoints and the relations between these concepts are built into different ontologies.

2) Generation of metadata of product data: According to the concepts from the viewpoint, and based on the pre-defined concepts and their relations in engineering ontology, a customized view of a product data is built into metadata.

Since most of the commercial CAD/CAE applications provide open API to access information from the application. Information about the product structure can be obtained through calling the API to build the metadata. After the pre-build ontology is given, based on the existing concepts, an application is built to automatically extract the product information from the CAD database for metadata. In this paper, an application is implemented based on Java, Jena(OWL API) and Jlink (Pro/Engineer API) as an example.

3) Editing and refinement of product data semantics obtained in previous processes. Since the functionalities of application API are limited, some of the information cannot be obtained by calling API function, and some information is not included in existing product data, it may sometimes be necessary to edit the metadata to refine the information.

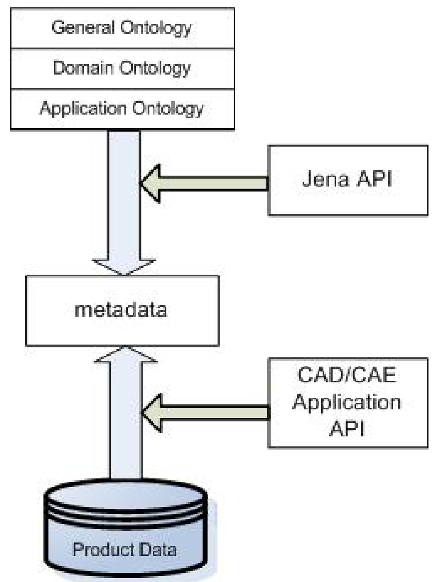

Figure 6

5.2 Process of automated generation of metadata:

By accessing product data and finding the corresponding concepts and relations in engineering ontologies, metadata is created by instantiating the concepts with instances (individuals). The following procedures are used for automated generation of metadata:

1) Matching concepts in Engineering Ontologies to Product Data

After specific API functions are identified for accessing the product database, data of product design including name (id) and value is obtained through application API and matched to the corresponding concept in ontologies. The application ontology layer in engineering ontologies makes sure all the data from product data belongs to at least one concept.

2) Instantiating concepts with product data

Product data can be considered as objects which instantiate design concepts with specific data. Since there are many concepts and relations defined in the engineering ontologies, for each concept and relation, there could be product data embraces them. By iterating the concepts in engineering ontologies and finding the matching product data, instances (or called individual) of concepts and their properties in ontologies are created in OWL with specific values from product data, which plays a role to link the actual product data to concepts. For each instance represented using RDF/OWL, there is an URI(Uniform Resource Identifier) to uniquely identify it from other instances. Figure 7 shows an instance Block1_Depth which instantiates the concept of *Protrusion_depth* in Pro/Engineer application ontology (marked by *ProeDesign* prefix) with a value of 100.0.

```
<ProeDesign:Protrusion_depth rdf:ID="Block1_Depth">
      <dimension_value
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >100.0</dimension_value>
    </Protrusion_depth>
```

Figure 7

3) Assigning relations to instances

There are huge data dependencies exist in product data, reflect in the engineering ontoligies, there are the relations between concepts. The data dependencies can be obtained through API and matched to the relations in product ontology and finally assigned to the corresponding instances. Overall there are two types of basic relations in product ontology: “has-a” and “is-a”, other more specific relations are directly or indirectly derived from them. In creating instances, “is-a” relation is assigned to indicate one instance belongs to a concept. for “has-a” relationship, it is used to describe composition relationships commonly exist in product design, for example, between Assembly/Part, Part/Feature, and Feature/Dimensions.

The relations between concepts are presented as properties in RDF/OWL. For example, figure 8 shows the “has_feature” relations, a sub-property of “has-a” property between Part1 and two features: Hole1 and Block1.

```
<ProeDesign:Part rdf:ID="Part1">
   <has_feature rdf:resource="#Hole1"/>
   <has_feature rdf:resource="#Block1"/>
 </Part>
```

Figure 8

4) Editing current ontologies

After extracting information from CAD database and building the corresponding instances and properties, since some concepts and relations might not be covered by API functions, it is necessary to edit the ontology, to add new concepts and relations to the existing ontologies, or add more annotation and change some of the code to be easier to understand by engineer. An ontology editor for design is being developed as a plug-in for protégé, in this plug-in, user can add and customize concepts and relations in design domain.

Figure 9

6. Example Scenario

Here we use an example of how the product data semantics is captured in two different domains: an ergonomics system to evaluate visibility and a CAD system (Pro/Engineer). During the integration, the CAD tool (Pro/Engineer) is used to design an automobile cab dashboard assembly which consists of several parts including gauges and panels. A tracked person in an immersive virtual environment (VADE[31]) is used to evaluate the visibility of the gauges on the dashboard while performing certain tasks [Figure 9].

In the CAD domain, the dashboard is described using feature and parameter concepts, similar to the example in section 4. For visibility analysis, the dashboard visibility Analysis ontology defines the product design from another point view:

The objective of the visibility analysis is to evaluate the visibility of dashboard, especially the visibility of gauges on the dashboard, from this point of view, the product data semantics of the dashboard is described using following concepts:

1) The structure of the dashboard is described using gauge and panel; also some key factors in visibility analysis are used in this case, for example, the factor of size of text.

Characteristics of a dashboard needed for visibility evaluation include: Opacity/transparency, color, alphanumeric character dimensions, shapes and positions of components on the dashboard and other concepts related to visual evaluation including human anthropometric measurement, vision rating etc. Geometry: gauge, contrast ratio, text size (height, width, stroke, spacing), placement of gauges

2) Functionalities of dashboard

A gauge within a vehicle must provide information to the operator. Gauge text size, width, stroke and spacing must be adequate to provide the information. Critical information, for example, such as vehicle speed or engine revolution speed (rpm) should be able to be gathered by the driver in a single glance. In an immersive environment, visibility studies can be performed on a range of human anthropometric sizes by scaling the human model within the virtual environment.

Currently we are working on the capturing of product data semantics on the dashboard in these two domains.

6. Future Work

Having implemented the method to capture product data semantics, we are now working on method of translating product data semantics from different points of view. Our method of translating product data semantics is to calculate similarity between concepts in ontologies for ontology alignment, and thus to map instances in product data semantics from one viewpoint to another. By capturing and translating product data semantics, a semantic-level CAD/CAE integration framework is being developed to be adaptive to different viewpoints.

7. Summary

In this paper a method of capturing product data semantic is proposed to improve the interoperability of CAD/CAE integration. In this approach, pre-defined and dynamically generated product ontologies are used in the integration framework to capture and represent product data semantics. To integrate with the traditional engineering applications, the data model is divided into concepts and metadata, the ontology and real data are linked through metadata file. During the data modeling process, an ontology builder is used for users to build in the product data semantics, an application is developed based on application API to extract the actual design data to generate metadata files of the concepts.

8. Acknowledgements.

This work was supported by National Science Foundation (NSF) Grant No.0523052

References:

[1] W. Regli, X. Hu, M. Atwood, and W. Sun, "A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval", Engineering With Computers, vol. 16, pp. 209-235, 2000.

[2] T. R. Gruber, "Toward Principles of the Design of Ontologies Used for Knowledge Sharing", International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation, 1993.

[3] R. Fikes and A. Farquhar, "Distributed Repositories of Highly Expressive Reusable Ontologies", IEEE Intelligent System, vol. 14, no. 2, pp. 73-79, 1999.

[4] M. Ciocoiu and D. S. Nau, "Ontology Based Semantics", Seventh International Conference on Principles of Knowledge Representation and Reasoning, 2000.

[5] Y. Nomaguchi, A. Ohnuma, and K. Fujita, "Design Rationale Acquisition in Conceptual Design by Hierarchical Integration of Action, Model and Argumentation", Proceeding of the 2004 ASME Design Engineering Technical Conference, 2004.

[6] G. Mocko, R. Malak, C. Paredis, and R. Peak, "A knowledge repository for behavioral models in engineering design", Proceeding of the 2004 ASME Design Engineering Technical Conferences, 2004.

[7] C. Kerr, R. Roy, and P. J.Sackett, "A product ontology for automotive seat specification", Proceeding of the 2004 ASME Design Engineering Technical Conferences, 2004.

[8] M. Ciocoiu, D. S. Nau, and M. Gruninger, "Ontologies for Integrating Engineering applications", Journal of Computing and Information Science in Engineering, vol. 1, pp. 12-22, Mar.2001.

[9] M. Gruninger, C.Schlenoff, and A. Knutilla, "Using Process Requirement as the Basis for the Creation and Evaluation of Process Ontologies for Enterprise Modeling", ACM SIGGROUP Bulletin Special Issue on Enterprise Modelling, vol. 18, no. 3 1997.

[10] C. Schlenoff, A. Knutilla, and S. Ray, "Unified Process Specification Language: Requirements for Modeling Process", Technical Report NISTIR 5910, Gaithersburg, MD,1996.

[11] L. Patil, D. Dutta, R. Sriram, "Ontology-Based Exchange of Product Data Semantics", IEEE Transaction On Automation Science And Engineering, Vol. 2, No. 3, July 2005

[12] C. Cera, W. C. Regli, I. Braude, Y. Shapirstein, and C. V. Foster, "A Collaborative 3D Environment for Authoring Design Semantics", IEEE Computer

Graphics and Applications, vol.1, no.1, pp.3-11, 2002.

[13] J. Kopena and W. C. Regli, "Functional Modeling of Engineering Designs for the Semantic Web", IEEE Data Engineering Bulletin, vol. 26, no. 4, pp. 55-62, 2003.

[14] J. Kopena and W. C. Regli, "Design Repositories on the Semantic Web with Description-Logic Enabled Service", First International Semantic Web and Databases Workshop, 2003.

[15] J. Nanda, H. J. Thevenot, T. W. Simpson, and S. R. T. Kumara, "Exploring Semantic Web Technologies for Product Family Modeling", Proceeding of the 2004 ASME Design Engineering Technical Conference, 2004.

[16] L. Anthony, W. C. Regli, J. E. John, S. V. Lombeya, "An Approach to Capturing Structure, Behavior, and Function of Artifacts in Computer-Aided Design", Journal of Computing and Information Science in Engineering, June 2001, Volume 1, Issue 2, pp. 186-192

[17] S. Ahmed, S. Kim, K. M. Wallace, "A Methodology For Creating Ontologies For Engineering Design", ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 24-28, 2005

[18] W3C, "W3C Semantic Web", Available online via http://www.w3.org/2001/sw/> [accessed June 2004], 2001.

[19] G. Antoniou and F.V. Harmelen, "A Semantic Web Primer", MIT Press, 2004.

[20] W3C, "Resource Description Framework (RDF) ", Available online via <http://www. w3.org/RDF/> [accessed June 2004], 2004.

[21] S. Powers, "Practical RDF", O'Reilly, 2003.

[22] S. Ray, "Interoperability Standards in the Semantic Web", Journal of Computing and Information Science in Engineering, vol. 2, pp. 65-69, Mar.2002.

[23] W3C, " OWL Web Ontology Language Overview", Available online via <http://www.w3.org/TR/owl-features/> [accessed June 2004], 2004.

[24] "Jena - A Semantic Web Framework for Java", Available online via <http://jena.sourceforge.net/> [accessed June 2004], 2004.

[25] "The Protege Ontology Editor and Knowledge Acquisition System", Available online via <http://protege.stanford.edu/> [accessed June 2004], 2005.

[26] J. Vanwelkenhuysen, R. Mizoguchi, "Ontologies and guidelines for Modeling Digital Systems", Proceeding of 9th International Knowledge Acquisition for Knowledge-Based Systems Workshop, Bannf, Canada, 1995.

[27] Y. Kitamura, T. Sano, R. Mizoguchi, "Functional Understanding based on an Ontology of Functional Concepts", Proceeding of 6th Pacific Rim International Conference on Artificial Intelligence, pp. 723-733, Melbourne, Australia, August, 2000.

[28] Y. Kitamura, R. Mizoguchi, "Meta-Functions of Artifacts", Proceeding of 13th International Workshop on Qualitative Reasoning, pp. 136-145, Loch Awe, Scotland, 1999.

[29] S. Szykman, J.W. Racz, and R. D. Sriram. "The representation of function in computer-based design", ASME Design Engineering Technical Conferences, 11th International Conference on Design Theory and Methodology, USA, September 12-16, Las Vegas, NV 1999.

[30] S. Szykman, J. Senfaute, and R. D. Sriram. "Using xml to describe function and taxonomies in computer-based design", ASME Design Engineering Technical Conferences, 19th Computers and Information in Engineering Conference, September 12-16, Las Vegas, NV 1999.

[31] S. Jayaram, U. Jayaram, Y. Wang, K. Lyons, and P. Hart, "VADE: A Virtual Assembly Design Environment," IEEE Computer Graphics and Applications, vol. Nov/Dec 1999.