Semantic Web-based Product Lifecycle Management with Protected Interactive 3D Collaboration via Remote Rendering

Jonathan Asselin

A Thesis

In

The Department

Of

Mechanical and Industrial Engineering

Presented in Fulfillment of the Requirements

for the Degree of Master of Applied Science (Industrial Engineering) at

Concordia University

Montreal, Quebec, Canada

August 2009

Your file Votre référence
ISBN: 978-0-494-63047-1
Our file Notre référence
ISBN: 978-0-494-63047-1

Abstract

Semantic Web-based Product Lifecycle Management
with Protected Interactive 3D Collaboration via Remote Rendering

Jonathan Asselin

Product Lifecycle Management (PLM) is the activity of managing products over their lifecycle. The goal is to achieve a total integration of all components to better understand products and their environments and to establish competitive advantages. This thesis proposes the use of semantic web technologies to support PLM. An overview of the semantic web briefly describes the basic building blocks. A review of literature reveals many useful applications from automated question answering to information security. The framework for semantic web-based PLM collaboration is examined and an ontology development process is presented in the context of PLM. Software systems were developed to perform semantic web automation and interactive 3D visualization. Applications of semantic web technologies for information retrieval, step by step process support, and design and manufacturing support are demonstrated. A novel 3D collaboration approach that protects 3D virtual assets via remote rendering and the service of a trusted shared mask broker was developed. The novelty of the proposed approach is that companies can collaborate on shared digital mockups without having to share their 3D models. Instead, only the required views for specific tasks are automatically exchanged, thereby eliminating the need for 3D navigation.

Keywords: Semantic Web, Collaboration, PLM

Acknowledgements

Every new generation builds on the contributions of past generations. I am thankful for all the people who shared their knowledge and experiences making this work possible. Special thanks go to my supervisor, Dr. Ali Akgunduz, for his support and guidance during my graduate studies. I would like to thank the examination committee for their questions, comments, and suggestions, especially Dr. Ketra Schmitt who extensively reviewed my thesis and provided valuable insights.

I am grateful to my parents, Sylvie Lamothe and Charles Asselin, for their everlasting support and encouragement. Also, I would like to thank my girlfriend, Negar Tavakoli, for supporting me every step of the way.

Table of Contents

List of Figures viii

1 Introduction 1

1.1 Brief Description of PLM 1

1.2 The Need for PLM 2

1.3 Evolving Environment and Opportunities 2

1.4 Research Objectives and Overview 3

2 Overview of the Semantic Web 4

2.1 Vocabulary 5

2.1.1 Unicode 6

2.1.2 Uniform Resource Identifiers (URI) 6

2.1.3 Extensible Markup Language (XML) 8

2.1.4 Resource Description Framework (RDF) 8

2.1.5 Web Ontology Language (OWL) 11

2.1.6 SPARQL 12

2.1.7 Semantic Web Rule Language (SWRL) 13

2.2 Logic 14

3 Literature Review 15

3.1 Related Work in Ontology-based Systems 15

3.1.1 Collaboration 15

3.1.2 System Integration 17

3.1.3 Security ... 18

3.1.4 Management ... 19

3.2 Related Work in Image-based Collaboration ... 20

4 Semantic Web-based PLM Collaboration ... 22

4.1 Human Agent ... 23

4.2 Information ... 25

4.3 Machine Agent ... 26

5 Collaborative PLM Ontology Development ... 27

5.1 Identify PLM Requirements ... 28

5.2 Define PLM Concepts and Relationships ... 30

5.3 Gather Information ... 36

5.3.1 Existing Ontologies ... 37

5.3.2 Standards ... 37

6 Software Systems Integration and Implementation ... 39

6.1 Semantic Web Automation and Visualization System ... 39

6.1.1 Information Parsing and Encoding ... 41

6.1.2 Java2D Force-based Graph ... 42

6.2 Interactive 3D Collaboration System ... 43

6.2.1 OpenGL 3D Visualisation System ... 45

7 Applications of Semantic Web for PLM ... 47

7.1 Information Retrieval ... 47

7.1.1 Result Accuracy Compared to Popular Search Engines 52
7.2 Step by Step Process Support 57
7.3 Design and Manufacturing Support 59
8 Protected Remote Rendering Collaboration 61
8.1 Contribution of Literature Review 62
8.2 Remote Rendering Collaboration Technique 62
8.2.1 Step 1: Render Images 63
8.2.2 Step 2: Extract Depth Maps 64
8.2.3 Step 3: Compute Shared Masks 65
8.2.4 Step 4: Compute Shared Images 66
8.2.5 Step 5: Compute Complete Images 66
8.3 Semantic Description of Image based Collaboration 67
8.4 Automated Remote Rendering Collaboration 68
9 Conclusion 70
References 72
Appendix A C++ Code for Creating a Depth Map using OpenGL 77
Appendix B GLSL Fragment Program to Render the Shared Mask 78
Appendix C GLSL Fragment Program to Render the Shared Image 79
Appendix D GLSL Fragment Program for the Complete Image 80

List of Figures

Figure 1 Semantic Layer Cake (Bratt, 2007) 5

Figure 2 Example of Unicode Encoding in UTF-8 6

Figure 3 URI Example 7

Figure 4 RDF Basic Structure 8

Figure 5 RDF Example 10

Figure 6 SPARQL Query 12

Figure 7 Rule that cannot be expressed using DL 13

Figure 8 Semantic Web-based Collaboration System 23

Figure 9 Example of a Semantic Description for a Consultant 24

Figure 10 Semantic Description for a 3D Model 25

Figure 11 Semantic Description for a Semantic Web Reasoner 26

Figure 12 PLM Ontology Development Process 28

Figure 13 Relationship between Natural Language and Description Logic 30

Figure 14 Definition of Activity (Soanes & Stevenson, 2005) 31

Figure 15 DL axiom defining Activity 32

Figure 16 Product Ontology Example 33

Figure 17 People Ontology Example 34

Figure 18 Activity Ontology Example 35

Figure 19 Taxonomy of Product and Services based on the UNSPC 38

Figure 20 Semantic Web Automation and Visualization System Architecture 40

Figure 21 Force-based Graph Applet 43

Figure 22 Interactive 3D Collaboration System Architecture 44

Figure 23 User Interface for the Interactive 3D Collaboration System 46

Figure 24 Searching for PLM Benefits using a DL Query in Protégé 4.0 49

Figure 25 Semantic Description of a PLM Benefit Linked to Source 50

Figure 26 Searching for Companies that Sell Products for PLM with Protégé 4.0 51

Figure 27 Google Search using Keywords 53

Figure 28 Google Search with Linked Keywords 54

Figure 29 Searching Concordia's Google Powered Search 54

Figure 30 Searching with Bing 55

Figure 31 Retrieving the Information about Concordia's Vision 56

Figure 32 Product Planning Process 58

Figure 33 Design and Manufacturing Information for an Oak Table 60

Figure 34 Using a Trusted Broker to Generate Shared Masks 63

Figure 35 Orthogonal Projection Depth Map for of a Car Body 65

Figure 36 Complete Image 66

Figure 37 Illustration of Semantic Relationships for Image based Collaboration 67

Figure 38 Sequence Diagram: Successful Automated Remote Rendering Collaboration 69

1 Introduction

The 21^{st} century will be defined by the establishment of a global economy. Countries from all around the world are forging alliances and signing trading treaties in order to facilitate the flow of products and services. Because the global economy is in a transition phase, it is very important for companies to be flexible and adaptive to the market. Companies that understand their position in this global ecosystem of industries will strive and benefit from the global market. On the other hand, companies that are not careful will probably be overtaken by a competitor. Product Lifecycle Management is a multidimensional management activity that enables companies to take control of their products (Stark, 2004). By promoting an integrated management approach, companies will be able to benefit from the global market and the sea of information that follows. Semantic Web technologies are proposed to effectively support PLM activities.

1.1 Brief Description of PLM

PLM is the activity of managing products and services across their lifecycle from the very first ideas and requirements to the environmentally friendly disposal of the product. Companies that make use of PLM will be able to take control of what happens to their products and services. PLM is a multidimensional activity that aims toward the total integration of relevant components to create, sustain, and optimize world class products that are in harmony with the world and their users over its lifecycle.

1.2 The Need for PLM

In a global market, companies will have access to more resources and customers than ever before, but they will also face global competition. In the short term, this environment will push companies to produce faster, better, and cheaper products. There will be more and more products to support and improve the product development process. This creates an environment that is difficult to understand and control. Product Lifecycle Management (PLM) is a holistic management approach that allows companies to take control of their products and services in the global economy. Unfortunately, half of the PLM initiatives fail to succeed in practice (Stark, 2004). Many companies fail to correctly implement PLM because it is difficult to fully understand all of the implications.

1.3 Evolving Environment and Opportunities

Global resources such as people, platforms, techniques, and applications are growing at a very rapid pace. With every new generation of information technology products, new features are added and others are improved. Keeping track of everything that might affect a product over its lifecycle is a difficult task, if not impossible due to the large amount of information that is spread all over the world in different languages and in different encoding schemes. A study by the IDC research firm predicts that the world wide annual growth of data requirements between 2006 and 2011 is expected to be almost 60%. (IDC, March 2008). This represents a sea of data of about 281 billion GB by the year 2011. To unlock the value of data, users have to know its meaning (Stark, 2004). For companies to make the most of global markets and the sea of data that follows, they first need to understand their product's environment which is continuously evolving. In other words, they must be effective at product lifecycle management.

1.4 Research Objectives and Overview

The objectives of this thesis are to study semantic web technologies and demonstrate their possible applications for PLM. An overview of the semantic web is presented with brief descriptions of the basic building blocks. A review of literature reveals that there are many useful applications from automated question answering to information security.

A framework for semantic web-based collaboration is presented in the context of global enterprises. In the absence of a standard PLM ontology, a systematic process for PLM ontology development is shown. Software systems were developed to perform semantic web automation and interactive 3D visualization. A novel 3D collaboration approach that protects 3D virtual assets via remote rendering and the service of a trusted shared mask broker was developed.

2 Overview of the Semantic Web

The Semantic Web is an extension to the World Wide Web where information is given well-defined meaning, enabling computers and people to work in cooperation (Berners-Lee, Hendler, & Lassila, 2001). The primary objective of the semantic web is to facilitate the use of data. After years of research, the semantic web has a solid foundation based on a common framework which supports a wide range of applications. It provides common data formats and languages that enable open and efficient collaboration. The semantic web layer cake diagram, first defined by Sir Tim Bernard Lee, is a common illustration of the main components of the Semantic Web. The most recent illustration is shown in figure 1 (right) along with a pyramid that represents the ascension toward higher expressivity and logical understanding (left). This section presents the semantics and syntaxes of the World Wide Web Consortium (W3C) languages for the semantic web.

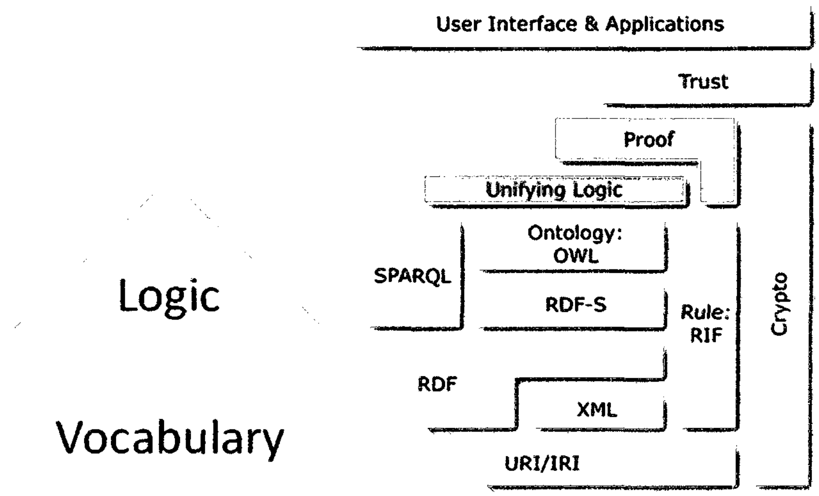

Figure 1 Semantic Layer Cake (Bratt, 2007)

2.1 Vocabulary

Within a global collaborative environment where people are geographically separated and speak different languages, misunderstandings can easily occur. The large amount of data increases the complexities of understanding and managing information simply due to the volume. The semantic web can be used to assign meaning to data. For this meaning to be shared and understood by people or machines, a series of languages with very specific syntax and semantics were developed. Those languages enable a consistent representation of different types of knowledge. The W3C defined a series of languages with open specifications based on the web and knowledge representation science. Every specification is meant to solve a specific knowledge representation issue such as how to identify different resources or how to define abstract concepts. Some examples are URI, XML, RDF, RDFS, OWL, SPARQL, and SWRL.

2.1.1 Unicode

Throughout history man has used different kinds of media such as rock, tissue, and paper as a means to record knowledge. With the arrival of computer systems a new and more powerful medium began to emerge; zeros and ones can be used to represent anything that can be imagined. Nowadays, a wide range of information can be created, stored, secured, shared, and managed by millions of people from around the world. To support any written language, the W3C uses Unicode encoding as a base component for its family of specifications. The universal character set (Unicode/ISO 10646) provides a standard encoding system where every character from any language, including historical languages, can be represented by a unique number. Using Unicode, writing from any language can be encoded as hex numbers and stored inside some computer system. Figure 2 shows an example where the word knowledge is represented in UTF-8 Unicode.

BASIC LATIN		UTF-8 UNICODE
Knowledge	→	4B 6E 6F 77 6C 65 64 67 65

Figure 2 Example of Unicode Encoding in UTF-8

While Unicode text can be read by every modern computer system, the information can be effectively secured using cryptography.

2.1.2 Uniform Resource Identifiers (URI)

A uniform resource identifier (URI) consists of a sequence of characters that is used to identify resources. URIs can provide the location of a resource (URL), the name of a resource (URN), or both. URIs offer the possibility of defining different types of addresses for different types of systems. This is accomplished through the definition of a URI scheme used to validate system specific syntax. The URI scheme identifies the type of system that contains the resource. A commonly used URI scheme is "http" and it identifies a web resource. Figure 3 shows an example of a URI for the resource "Knowledge" that is located in the file "plm.owl" on the "concordia.ca" http web server.

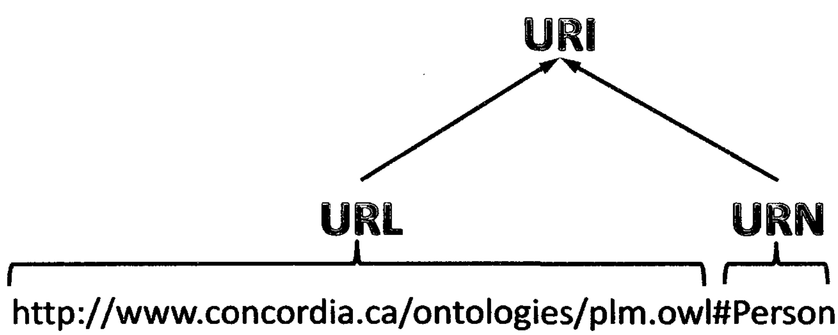

Figure 3 URI Example

While the URI syntax defines a grammar that is a superset of all valid URI, the URI scheme may further restrict the syntax in order to ensure the validity of the system specific address. The complete specification can be found on the W3C web site. (Berners-Lee, Fielding, & Masinter, 2005)

2.1.3 Extensible Markup Language (XML)

Extensible Markup Language (XML) is a plain text markup language much like HTML. It is designed to carry data in an open and human readable format. Unlike HTML, it does not rely on a set of predefined tags. The user defines the tags that would be relevant to define and structure the information. While XML is essentially plain text, the simplicity and flexibility of the language made it a widely used standard for sharing information. XML is now used as a building block for everything from web pages to professional applications. It is recommended by the World Wide Web Consortium (W3C).

2.1.4 Resource Description Framework (RDF)

The Resource Description Framework (RDF) is a general-purpose language for representing information. It expresses relationships using triples that consist of a subject, a predicate and an object each having their own URIs. The basic structure of RDF triples is shown in figure 4.

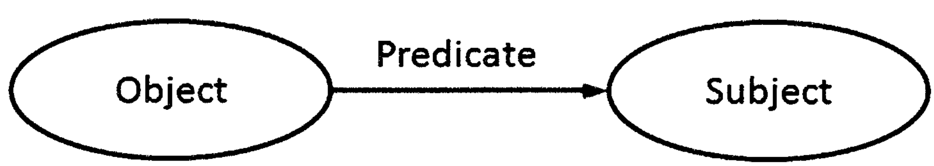

Figure 4 RDF Basic Structure

A set of triples is called an RDF graph and it can be used to represent the relationships that define the different resources. An RDF graph that represents some information about this thesis is presented in figure 5. The thesis is identified with a resource named Paper_ID1, it has a *title* "Semantic Web-based Product Lifecycle Management with Protected Interactive 3D Collaboration via Remote Rendering" and an *author* Student_ID1. Student_ID1 has a *fullname* "Jonathan Asselin", has a *degree* Bachelor of Industrial Engineering (BIE), and has a *supervisor* Professor_ID1. Professor_ID1 has a *fullname* "Ali Akgunduz", and has *degrees* Bachelor of Science (BS), Master of Business Administration (MBA), and Doctor of Philosophy (PhD).

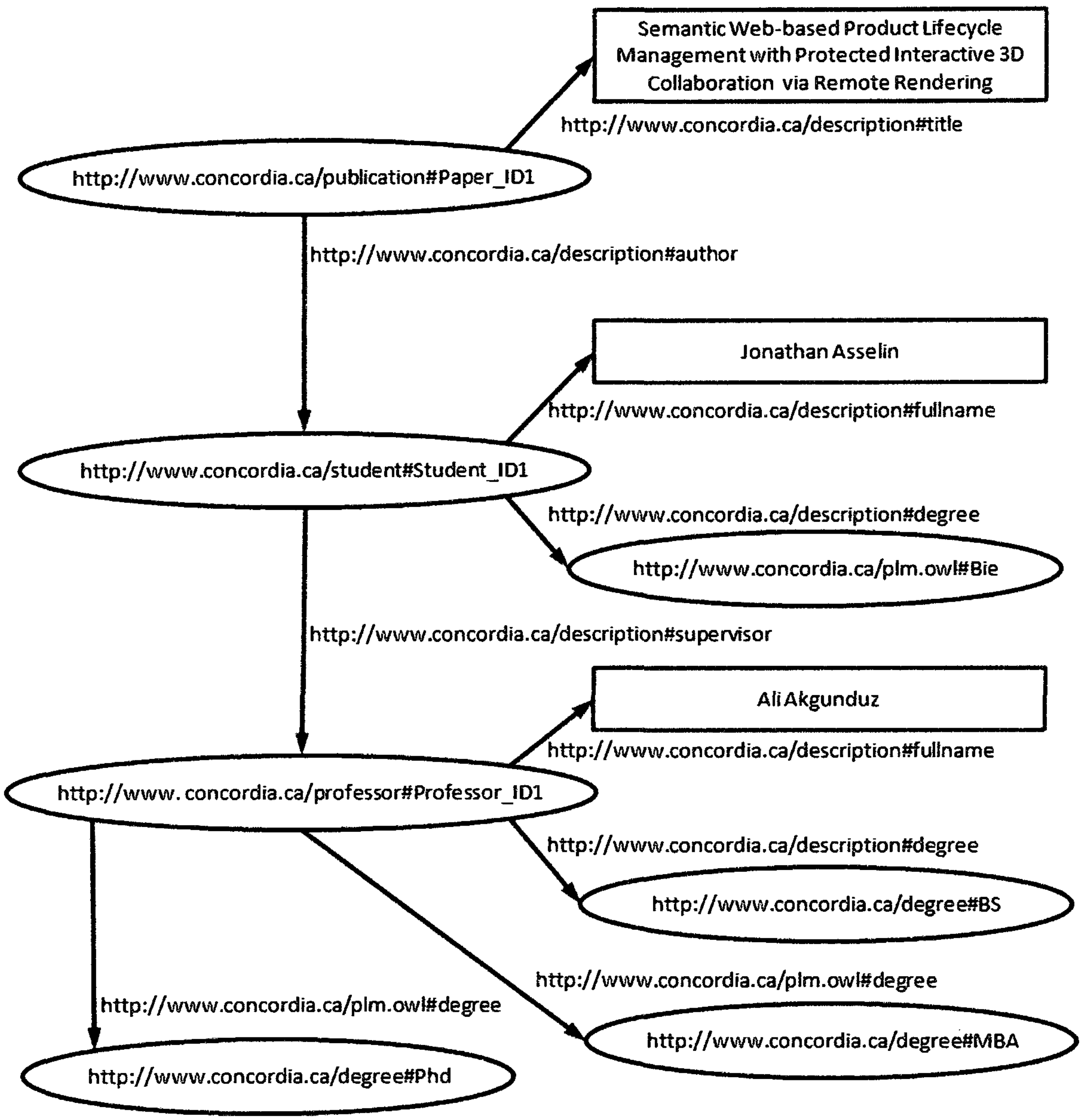

Figure 5 RDF Example

In order to further describe resources and their implications, the RDF's vocabulary description language, RDF Schema (RDFS), can be used. RDF Schema extends RDF to define classes, properties and other resources. RDF Schema is implemented on top of RDF; therefore every valid RDFS document is a valid RDF document.

2.1.5 Web Ontology Language (OWL)

OWL is a web ontology language that can be used to define a more formal description known as ontology. It provides additional vocabulary along with formal semantics based on knowledge representation (KR). The use of formal semantics facilitates the processing of information by machines. OWL is designed to integrate with applications that process information such as reasoners or other types of computer agents. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL for Description Logic, and OWL Full. Each of these sublanguages is an extension of its simpler predecessor. While the added expressivity can allow for a more complete representation, it comes at the cost of computation, decidability, and predictability. The choice of expressivity depends on the requirement for the ontology. Ontology designed using lower level sublanguages can be later extended using higher level sublanguages. Every legal OWL Lite ontology is a legal OWL DL ontology which is a legal OWL Full ontology. On the other hand, higher level languages are not legal representation with respect to lower level languages.

The basic elements of OWL ontology are classes, properties, and instances of classes also known as individuals. Knowledge can be encoded using logical axioms that describe basic elements and their relationships. Taxonomies are used to define the classification of classes and properties into hierarchical structure. OWL is based on the open world assumption which implies that logical conclusion cannot be drawn from the lack of knowledge.

2.1.6 SPARQL

SPARQL is a query language and protocol for RDF. The protocol defines how queries and results should be transferred between clients and processors. This enables a consistent deployment of RDF query web services. The SPARQL query can be used to search a wide range of data sources. The queried data can be natively stored as RDF or it can be extracted as RDF from other systems such as SQL databases. SPARQL allows for required or optional graph patterns along with conjunctions, and disjunctions to be queried. An example of an SPARQL query is presented in figure 6. The query shown in figure 6 will return a list of titles for which the full name of the author was defined. Also, if the degree of the author is known, it will be provided.

```
PREFIX plm: <http://www.semavision.com/plm.owl#>
SELECT ?title ?author ?degree
WHERE {
        ?x plm:title ?title ;
           plm:author ?y .
        ?y plm:fullname ?author .

OPTIONAL {
        ?y  plm :degree ?degree }
```

Figure 6 SPARQL Query

2.1.7 Semantic Web Rule Language (SWRL)

The Semantic Web Rule Language extends OWL DL and OWL Lite sublanguages with the ability to include rule axioms. Rules generally convey implications such as when an antecedent (body) holds then the consequence (head) must also hold. If the antecedent is empty the consequence is considered an unconditional fact. Using SWRL different types of knowledge can be captured using rules and integrated to an OWL knowledge base. Since SWRL allows the use of variables, it can express some knowledge that description logic cannot. Figure 7 shows an example of a rule that cannot be expressed using DL. If someone has a parent who has a brother, then the brother of the parent is his uncle.

```
hasParent(?child, ?parent) ^ hasBrother(?parent; ?uncle) ⇒ hasUncle(?child; ?uncle)
```

Figure 7 Rule that cannot be expressed using DL

While SWRL offers added expressivity and capabilities, it comes at the price of potentially losing decidability. Whenever possible, knowledge should be recorded using description logic to maintain decidability. When rules are used special consideration must be taken to maintain decidability.

2.2 Logic

Semantically linked data enables computational analysis to draw logical conclusions which can be used to further construct the web. Historically logical inferences about complex problems were only possible by humans, with the advancement of computer science and description logic, it is now possible to automate this process. This means that humans and machines are now able to work together using logic as a means of "understanding" each other. That is humans can understand and machines can compute the information.

There are many possible approaches and techniques to reasoning. In order to demonstrate the ideas in this thesis, Description Logic reasoning implemented through software solutions, mainly Racer (Haarslev & Moller, 2003) and Pellet (Sirin, et al, 2007) were used.

Details about Description Logic and other reasoning approaches can be found in *The Description Logic Handbook* (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003), and in *Knowledge Representation and Reasoning* (Brachman & Levesque, 2004).

3 Literature Review

There is a large body of research dating as far back as 1950 about artificial intelligence and advanced computer systems capable of providing meaningful information. The semantic web can be used to integrate different software technologies to create a more complete system infrastructure for PLM. The literature review shows some examples of what can be done with ontology-based systems in different areas related to PLM. Also, the literature that inspired the work for the image based collaboration technique is presented.

3.1 Related Work in Ontology-based Systems

It would be possible to integrate all of the ontology-based systems discussed in the literature review through the use of the semantic web. Ontology-based systems that could improve collaboration, system integration, security, and management have been studied.

3.1.1 Collaboration

An ontology-based collaborative assembly design platform is presented by Kim et al. (2006). They created ontology with a standard vocabulary which describes mechanical assemblies with positioning and constraint properties. Assembly constraints are extracted from CAD software, such as CATIA or NX, and explicitly defined in the ontology. Because the information is represented in a computer interpretable manner;

the constraints can be queried and reasoned by computer tools. Furthermore, the standard vocabulary can be used to facilitate collaboration among heterogeneous stakeholders and systems.

A manufacturing system engineering (MSE) ontology model for inter-enterprise collaboration is presented by Lin & Harding (2007). The MSE ontology has been designed to model the foundation of all manufacturing business applications. The applications have been captured in seven base classes: Extended_Enterprise, Project, Flow, Enterprise, Process, Resource, and Strategy. These classes have further defined subclasses. The authors propose a collaboration environment where users are not forced to change their vocabulary. This is accomplished by manually mapping the languages of individual collaborators together. While the manual method can be slow, future advances in automated ontology integration will facilitate the task of linking vocabularies. Common vocabularies for different industries and business functions could be developed through global collaboration. This type of approach could be very useful in international and/or multidisciplinary environments where different people want to use their own languages or terms while being able to communicate meaningfully.

An ontology based-group memory system for representing, recording, retrieving, and managing knowledge was developed by Vasconcelos, Kimble, and Gouveia (2000). This type of system can be used to preserve organizational memory and prevent enterprises from loosing part of their intellectual property. An ontology for multi-agent corporate memory systems is presented by Fabien (2001). The research suggests that ontologies will become a central component of multi-agent information systems (MAIS).

3.1.2 System Integration

Automated integration of PLM objects using ontology is presented by Kwak and Yong, (2008). The system uses OWL and a C# ontology model of class hierarchical structure and bill of material (BOM) information to automate the mapping of PLM objects from a product data management (PDM) system. The author demonstrated the automated integration of two PLM objects from MEMPHIS, a Middleware for Exchanging Machinery and Product Data in Highly Immersive Systems. They demonstrated the integration of automobile parts using synonym, class hierarchical structure, and Restriction. Since MEMPHIS is based on C# language and does not have semantic export, the integration system was implemented in a C# environment with a subset of the OWL vocabulary. A PDM system with OWL export could allow an organization to greatly facilitate and standardize their data integration tasks.

An ontology-based approach for explicitly specifying, capturing, interpreting, and reusing the product semantics to facilitate heterogeneous information sharing across CAD systems was proposed by Abdul-Ghafour, et al. (2007). The authors propose the use of Common Design Features Ontology (CDFO) to define the semantic and mapping of different CAD systems. Common modeling features of Catia V5, Pro/engineer, and

SolidWorks where encoded in OWL DL. A set of mapping rules are under development as part of their approach. The authors pointed out that not all concept in CAD systems have equivalents which results in loss of information. Ideally, CDFO descriptions would be provided by CAD vendors in order to facilitate the integration with other products. While software vendors are resistant to this idea, it will become an increasingly important feature for companies who want to be effective at PLM.

An analysis of ontology-based automation systems for integrating information of manufacturing equipment over its life-cycle is presented by Gössling and Wollschlaeger (2008). Different factory automation approaches are analyzed for automated integration of field device information over distinct lifecycle phases: design, production, planning, purchasing, application engineering, integration or deployment, operation, maintenance. To allow the seamless integration of information between devices over their lifecycle, the concept of foldable integration ontology is introduced. The foldable ontology has connections between different models that are described using a semantic property named CONSISTS-OF. This approach could enable the flexible integration of ontologies.

3.1.3 Security

Semantic Web technologies were used to develop a context-aware environment that can be used to empower users with contextual and conditional access control shown by Gandon and Sadeh (2004).

Ontology-based knowledge representation and an access control model for collaborative knowledge sharing throughout the product lifecycle is presented by Chen (2008). Chen developed an access control model and a knowledge access control policy (KACP) language used to describe the knowledge access control and sharing rules for virtual enterprises (VE) and their members. This approach offered more flexibility for defining authorization access. Basic access privileges are analyzed along four dimensions; who, what, where, and when. Extended access privileges are determined by considering the semantic relationships between three domain ontologies; product, organization and activity.

3.1.4 Management

The TOVE (Toronto Virtual Enterprise) ontology project presented by Fox (1992) aims to create a shared representation of the enterprise that each agent can understand, with precise and unambiguous term definitions. The semantic is implemented in a set of axioms that will allow TOVE to automatically deduce the answer to "common sense" questions. To model Enterprises, a set of ontologies have been developed: Activity, Resource, Organization, Product and Requirements, ISO9000 Quality, Activity-Based Costing. A cost ontology that maps to Activity-Based Costing (ABC) concepts has been described by Tham and Fox (1994). Activity based cost for enterprise operation can be reasoned, deduced, and computed using their micro-theory of costing.

A Feature ontology was used to support construction cost estimation by Staub, et al. (2003). The feature ontology describes some design conditions that affect construction costs. This knowledge can be reused to create feature-based product models that can provide costing information.

3.2 Related Work in Image-based Collaboration

A semantic description of 3D scenes to create an interactive 2.5D mobile environment has been developed by Mikovec, et al. (2006). The system was demonstrated by providing real time support on a mobile device for an inspector who was task to revise the facility of an office. Instead of providing the inspector with the complete 3D scene, a set of 2.5D images for different views is exchanged. Using image is particularly useful in a mobile collaborative environment where the bandwidth might be limited or costly. Also, it provided the added benefit that users do not have to navigate the 3D environment. Experimentation with the two approaches showed that users were generally more comfortable with the images.

The interaction between the user and the system relies on semantic descriptions encoded in OWL to represent a shared model of the 3D environment. Using the semantic description model, users can perform text based or vocal queries such as:

- What is contained in a specific room?
- What is the area of the floor?

- Show me a 2D rendering of the floor?

A more secure interactive 3D visualization via remote rendering has been developed by Koller, et al. (2004). Instead of providing the user with the full model, they provide a simplified version that can be moved and manipulated to get the desired view. Every time a user changes the view, a request is sent to a rendering server which provides a high quality rendering that has been secured against shape reconstruction attacks. The solution was presented in the context of sharing high quality rendering of historical artifacts in a way that will prevent unauthorized reproduction. Due to security concerns of curators, high quality rendering could not be provided in real time to the general public. With this solution, people from around the world can enjoy the intricacy of those beautiful pieces of art.

4 Semantic Web-based PLM Collaboration

Collaboration processes and methodologies continuously evolve and change over time. In order to remain competitive, companies need to adapt to a changing environment with more people, tools and techniques. Organizations need a collaboration system that is flexible and capable of easily integrating with different people and systems. Currently, integrating software solutions can be quite complex and costly, if not impossible. This is mainly due to the fact that many software vendors use proprietary vocabularies and formats. Semantic web-based collaboration can overcome these challenges. Regardless of the format, information can be linked together to be usable and "understandable". Another benefit is that it allows the deployment of a web of knowledge that will grow similarly to the World Wide Web.

Shared semantic web vocabularies can be used to break down collaboration barriers between people, machines, and information. Agents and information can be described using semantic descriptions that are written with a shared semantic vocabulary and encoded in an open format. The W3C proposed the use of Composite Capabilities/Preferences Profile (CC/PP) as a specification for defining agent capabilities and preferences. For business applications, clear security and data management policies could be semantically defined and enforced by human and machine agents.

This section describes the main components of a semantic web-based PLM collaboration system: people, information, and machines. Figure 8 illustrates the main components and their communication channels represented by bi-directional arrows.

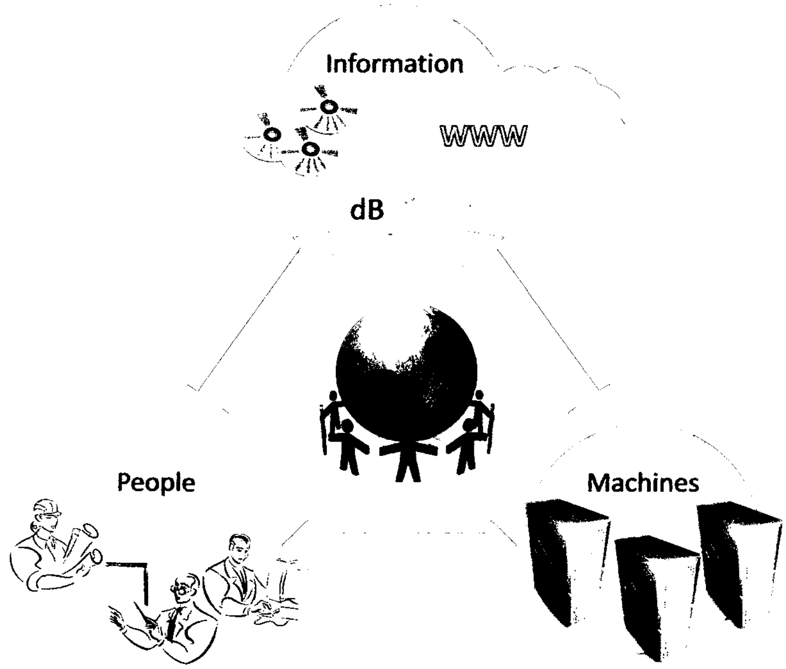

Figure 8 Semantic Web-based Collaboration System

4.1 Human Agent

The semantic environment will change the way people collaborate. It is important to understand how people will fit in to this new collaborative system. Everyone is different in their own way. People have different physiques, interests, qualifications, qualities, problems, and so on. The semantic web offers a way to encode all of this information and more with standardized syntax and semantics. Using Unicode encoding to capture human languages, people from different parts of the world will be able to communicate and exchange meaningful information. Semantic descriptions are used to record information about people. This can be used to better understand people and facilitate their ability to collaborate together. Information about customers can be used to better understand their needs and identify market segments. Also, management can use this information to hire the right people and form the best teams. A semantic description for a consultant specialized in semantic web technologies and visualization is presented in figure 9. Within the context of this example, a manager that is looking for a semantic web consultant would be able to find this person from anywhere around the world as soon as the description is posted on the web. Very expressive selection criteria could be used to narrow the group of potential collaborators to the ones that would best fit the needs of the organization.

```
<owl:Thing rdf:about="#Consultant_01">
   <name rdf:datatype="&xsd;string">Jonathan Asselin</version>
   <height rdf:datatype="&xsd;double">5.9</version>
   <rdf:type rdf:resource="#Consultant"/>
   <citizenOf rdf:resource="#Canada"/>
   <hasDegree rdf:resource="#Bachelor_of_Industrial_Engineering"/>
   <pursueDegree rdf:resource="#Master_of_Science"/>
   <workAt rdf:resource="#Concordia_University"/>
   <specializedIn rdf:resource="#Semantic_Web_Technologies"/>
   <specializedIn rdf:resource="#Visualization"/>
   <fluentIn rdf:resource="#English"/>
   <fluentIn rdf:resource="#French"/>
</owl:Thing>
```

Figure 9 Example of a Semantic Description for a Consultant

4.2 Information

It is estimated that the amount of data worldwide will reach 281 billion GB by the year 2011 (IDC, March 2008). To unlock the value of data, users have to know its meaning (Stark, 2004). The information can be encoded using open or proprietary file formats. It can be encrypted or secured in a database system. The idea is not to read the information, but simply to define what it is and how it interacts with its environment. The semantic web offers the possibility to add meaningful annotations that can be understood by humans and computed by machines. Semantic descriptions can greatly facilitate the use of data especially when there are many pieces of information encoded in many different formats. For example, figure 10 show the semantic description for a 3D model of the New_Car_01 which was created using Catia_V5_R17.

```
<owl:Thing rdf:about="#Model_01">
    <rdf:type rdf:resource="#CatiaV5_Model"/>
    <modelOf rdf:resource="#New_Car_01"/>
    <hasOwner rdf:resource="#Semavision"/>
    <hasResponsible rdf:resource="#Jonathan_Asselin"/>
    <hasDesigner rdf:resource="#John_Woo"/>
    <createdUsing rdf:resource="#Catia_V5_R17"/>
    <version rdf:datatype="&xsd;string">01</version>
</owl:Thing>
```

Figure 10 Semantic Description for a 3D Model

4.3 Machine Agent

The advantage of using computer agents is that they can be instantiated in many different systems at practically no operational cost. Deciding when it is best to model an agent can be done with a simple tradeoff study of development time versus the amount of time to perform the process manually multiplied by the frequency of the task (Gruber, Vemuri, & Rice, 1997, p. 15). Another advantage of computer agents is that they can significantly reduce the lead time of most processes. For some high technology products such as computers or airplanes, the time to market is critical. Just like every other component of the collaboration system, a machine agent uses semantic descriptions as identity cards. Figure 11 shows the semantic description for a reasoner agent. As described, the machine agent is a Windows XP server named “Racer Server 1” that runs RacerPro 2.0 and it can be accessed at http://192.168.1.1:8080/.

```
<owl:Thing rdf:about="#Server_01">
   <name rdf:datatype="&xsd;string">Racer Server 1</version>
   <rdf:type rdf:resource="#Server"/>
   <rdf:type rdf:resource="#Semantic_Web_Reasoner"/>
   <url rdf:datatype="&xsd;string">http://192.168.1.1:8080/</version>
   <hasOwner rdf:resource="#Semavision"/>
   <hasAdministrator rdf:resource="#Jonathan_Asselin"/>
   <hasLocation rdf:resource="#Racer_Server_01_Location"/>
   <hasOS rdf:resource="#Windows_XP_SP3"/>
   <runSoftware rdf:resource="#RacerPro_2.0"/>
   <processorType rdf:datatype="&xsd;string">Intel Xeon E3110</version>
   <processorSpeed rdf:datatype="&xsd;int">3000</version>
</owl:Thing>
```

Figure 11 Semantic Description for a Semantic Web Reasoner

5 Collaborative PLM Ontology Development

To obtain a semantic web-based PLM collaboration system that will allow the main components, people, information, and machines, to collaborate, the PLM ontology must be developed. The PLM ontology is a semantic model for the industrial domain. The idea is to create a logical representation of product lifecycle management concepts and their relationships. This section provides a general framework for developing the foundation for the PLM ontology in three phases; identify PLM requirements, define PLM concepts and relationships, and gather information. The enterprise ontology development process of Gruninger & Fox (1995) was used in the context of semantic web-based PLM.

The proposed PLM ontology development process is illustrated in Figure 12. The curve on top of the process represents the involvement level of collaborators during the development process. In order to develop a useful ontology, important concepts and relationships are identified using a set of competency questions (Gruninger & Fox, 1995). Important components are then defined in order to understand their relevance within the context of the organization. If a standard PLM ontology exists, companies should try to use it as a foundation to facilitate collaboration. Once the PLM ontology foundation is complete, different people and groups will be able to make comments, change requests, and add relevant information. They will be able to build their own domain specific ontology on top of the PLM ontology which will allow them to collaborate with other people and departments.

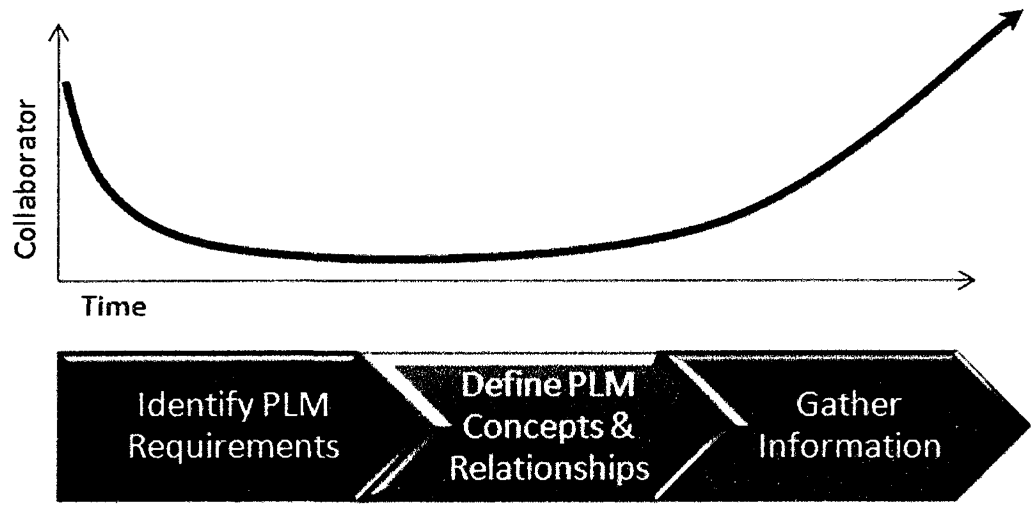

Figure 12 PLM Ontology Development Process

5.1 Identify PLM Requirements

Gathering questions from potential users of the PLM ontology should be the first step of the development process. It is relatively inexpensive and it provides a solid foundation for further development. In an industry setting, a group of employees could be asked to submit a list of questions that arise in their daily business. Some general questions relating to PLM are:

- What are the different features of product X?
- Who is responsible for module X?

- What is the cost breakdown for product X?
- Who understands system X in department Y?
- What modules for product X come from location Y?
- What is the length, width, height, or weight of module X?
- Which products offers feature X?
- What is the standard process to develop a new part?
- What is the collaboration process between X and Y?
- Can I share information X during presentation Y?
- What can be used to perform activity X?

Analyzing the competency questions provides a good understanding of the vocabulary, requirements and scope of the ontology. The frequency of terms is a good indicator of the relative importance for different people and groups. The other advantage of competency questions is that ontology designers will have a good idea of what is expected from the system (Gruninger & Fox, 1995). Figure 13 shows an example of how a question can be decomposed into fundamental concepts. While it would be possible to automate this process using natural language processing, this task was done manually for the scope of this project. The next step will be to define the concepts and begin capturing their semantic relationships as shown in the description logic part of figure 13.

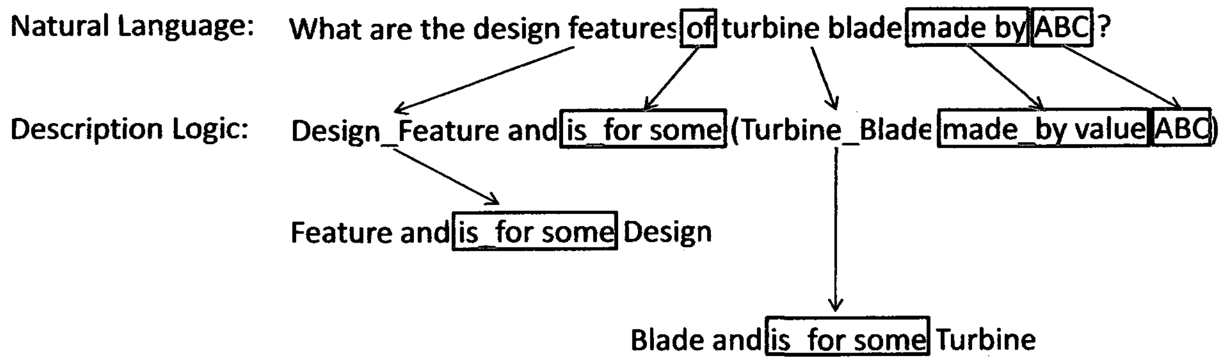

Figure 13 Relationship between Natural Language and Description Logic

5.2 Define PLM Concepts and Relationships

In this phase, PLM components are modeled as concepts and defined in the context of a particular business. Furthermore, some common relationships are defined as semantic properties linking concepts together. Some important concepts of PLM are people, product, activity, management, process, system, information, location, time, cost, quality, and lifecycle.

The first step is to examine the important concepts and to define company specific interpretations. This simple exercise will ensure that there is a common understanding of specific terms. For example a company using the activity concept might have a chemistry department that defines activity as a component in a solution rather than a thing that a person or group does or has done (Soanes & Stevenson, 2005). Figure 14 shows the different definitions of the word activity from the Oxford Dictionary of English.

activity

→ *noun* (pl. **activities**)
1. [mass noun] the condition in which things are happening or being done: *there has been a sustained level of activity in the economy.*

▪ busy or vigorous action or movement: *the room was a* ***hive of activity.***
2. (usu. ***activities***) a thing that a person or group does or has done: *the firm's marketing activities.*

▪ a recreational pursuit or pastime: *a range of sporting activities.*
3. *(Chemistry)* a thermodynamic quantity representing the effective concentration of a particular component in a solution or other system, equal to its concentration multiplied by an ***activity coefficient.***
- ORIGIN late Middle English: from French *activité* or late Latin *activitas*, from Latin *act-* 'done', from the verb *agere.*

Figure 14 Definition of Activity (Soanes & Stevenson, 2005)

When there are many possible definitions for one concept, the definition accepted by the vast majority of people should be included in the base PLM ontology. Other definitions should be included in separate domain specific ontologies that can be connected to the base ontology through semantic links.

It is important to understand that the PLM ontology is a continuously evolving model. Much like Wikipedia, the amount and the quality of information will continuously increase over time. The idea is not to model the whole system but to focus on understanding what is important for the organization. The PLM ontology should remain general, and specific domain knowledge from different collaborators should be stored in different ontologies that could be built on top of each other. Specific domain knowledge of specialist will not impact the understanding of many during collaboration activities. However, if there are two equally accepted definitions both should be included in the base PLM ontology. Once the definitions are established, semantic descriptions can be developed allowing the computation of relationships between different concepts. For example, the activity concept can be defined using the description logic axiom shown in figure 15. If this definition is enforced, a machine will not be considered has capable of performing an activity. Maybe a better term for this definition would be a "Human Activity".

$$\begin{aligned}\text{Activity} \equiv \text{Thing} &\cap (\exists \text{is-done-by}(\text{Person} \cup \text{Group}) \cup \exists\ \text{was-done-by}(\text{Person} \cup \text{Group})) \\ &\cap (\forall \text{is-done-by}(\text{Person} \cup \text{Group}) \cup \forall\ \text{was-done-by}(\text{Person} \cup \text{Group}))\end{aligned}$$

Figure 15 DL axiom defining Activity

Figure 16 is an example of a product ontology using a visual interface. It illustrates the taxonomy of some products and the semantic properties that define them. For example a "Powered fixed wing aircraft" "has-part" some "Aircraft wing" and some "Engine". During the definition phase, the products that are relevant to the organization should be identified and defined.

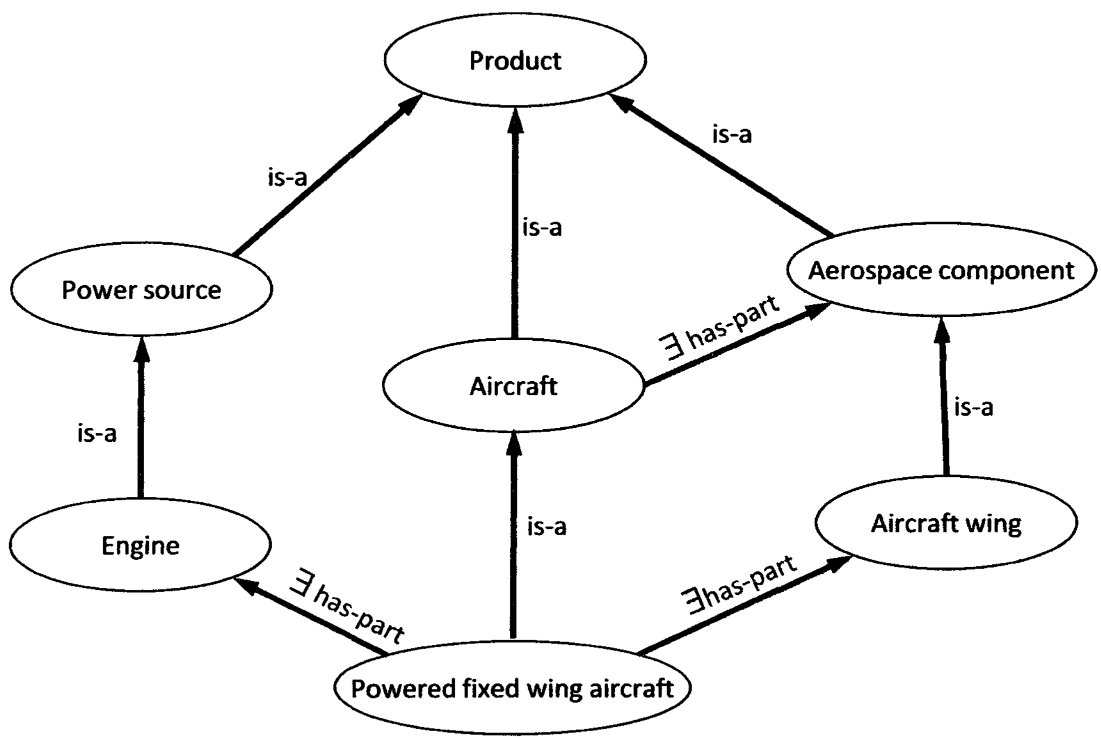

Figure 16 Product Ontology Example

Figure 17 illustrates a visual semantic description of people with relationships to the company and product concepts.

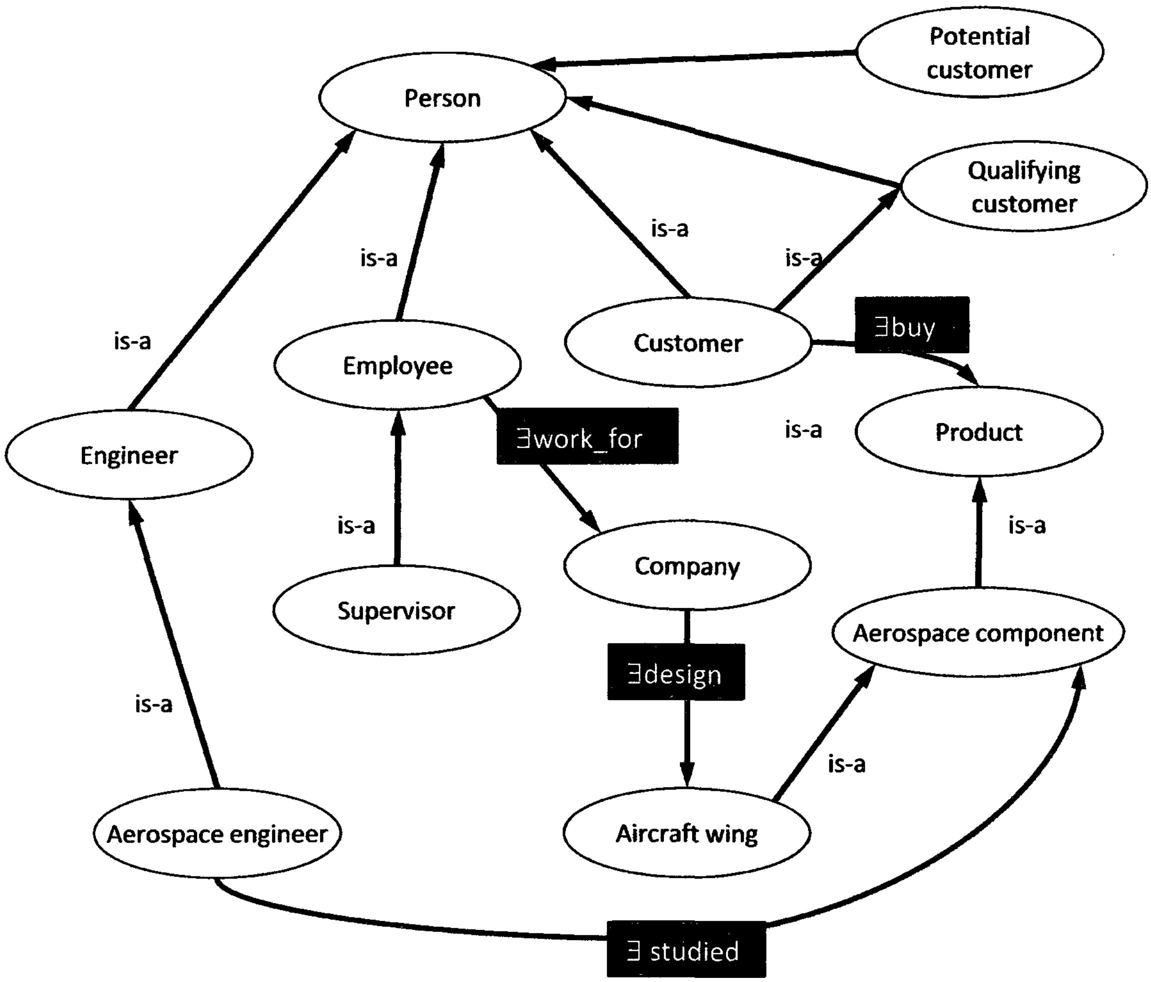

Figure 17 People Ontology Example

Figure 18 illustrates a visual semantic description of activity and some relationships to the people concept.

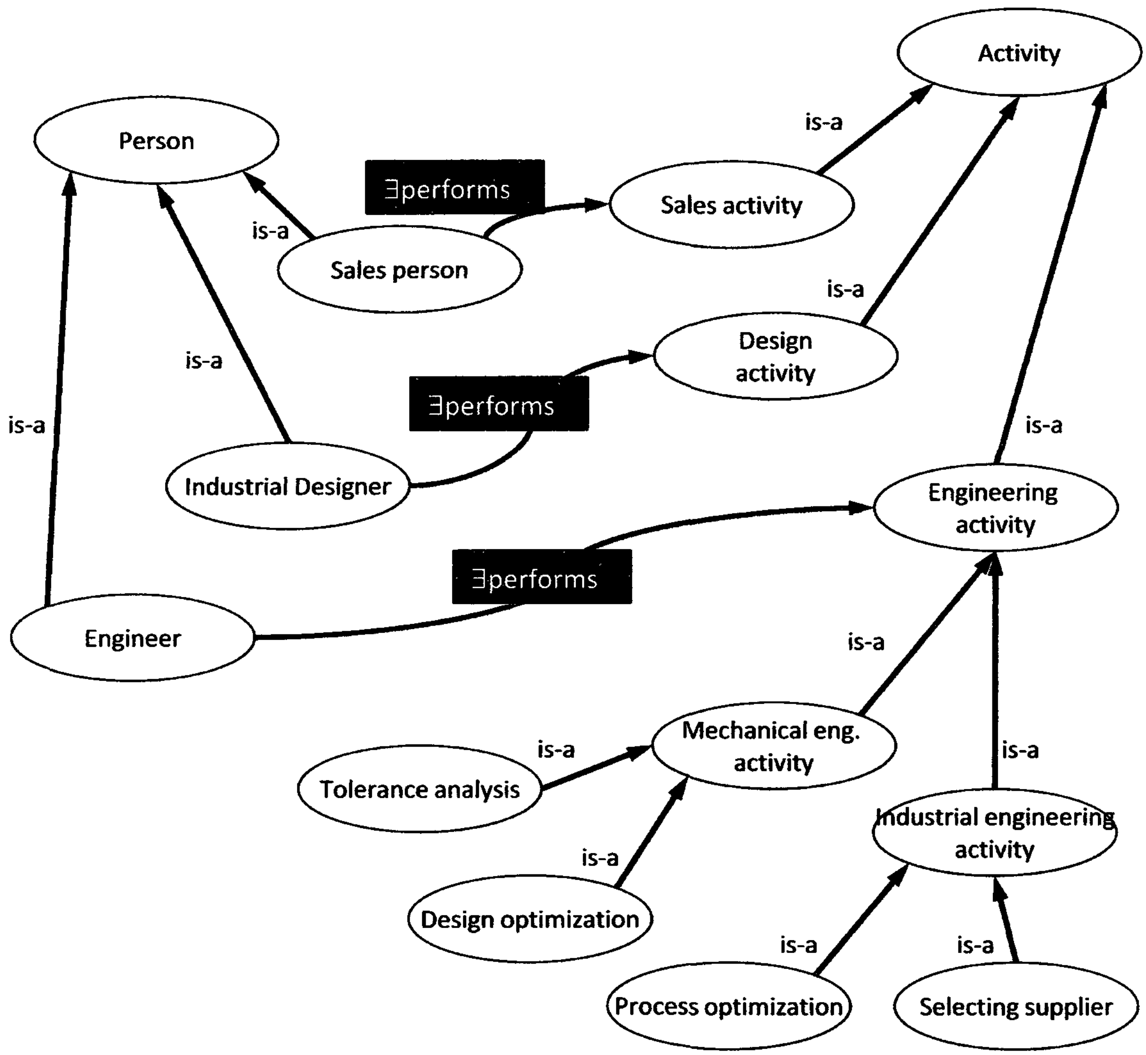

Figure 18 Activity Ontology Example

5.3 Gather Information

This phase is used to extend the PLM ontology in a collaborative environment. This section demonstrates possible integration with different knowledge sources. The additional information is encoded in a separate ontology that can be connected to the general PLM ontology with semantic relationships.

There is a vast amount of information that could prove to be useful for an organization's PLM initiatives. In fact, one of the main purposes of a PLM group is to find what could help the organization make better products. Making better products is a general concept that can include many components directly and indirectly related to the product. To get the full benefit of PLM ontology, it should be easy to add ideas, opinions, or information even if it appears to be unpopular or of little value to the business unit. In a global enterprise, what appears to be of little value for one business unit could be very important for another.

The PLM ontology is made to share information and its contents should not be over-controlled. It is possible to develop PLM ontologies with different levels of content control from information in the public domain to confidential information. An ontology provides a means of representing and connecting multidimensional sets of data from many types of sources. The PLM ontology should be continuously fed with the information from people, systems, standard, books, research papers, and other existing ontologies.

5.3.1 Existing Ontologies

Existing ontologies could be used if desired, however integrating the knowledge could be costly if it was not modeled to fit the specific business needs. Here are some examples. The Suggested Upper Merged Ontology (SUMO) with its domain ontologies forms the largest formal public ontology in existence today. (OntologyPortal) It has been proposed as a starter document for the Standard Upper Ontology Working Group which is an IEEE-sanctioned working group of collaborators from the fields of engineering, philosophy, and information science. (Niles & Pease, 2001) The Standard Upper Ontology (SUO) will act as a foundation for more specific domain ontologies.

5.3.2 Standards

In order to better understand product and services, ontology based on the United Nations Standard Products and Services Code (UNSPSC) was developed. The UNSPSC offers a global classification system for goods and services with five levels: segment, family, class, commodity, and business function. The UNSPSC information can be used for company-wide visibility of spend analysis, cost-effective procurement optimization, and a full exploitation of electronic commerce capabilities. Figure 19 shows the taxonomy of products classified based on UNSPC. There are over 40 000 concepts in this ontology. Reasoning on such a large scale is more time consuming and impractical if the knowledge is not required. This is a good example of a domain ontology for spend analysis, procurement optimization, and electronic commerce.

- Thing
  - Product_and_Service
    - Apparel_and_Luggage_and_Personal_Care_Products
    - Live_Plant_and_Animal_Material_and_Accessories_and_Supplies
    - Building_and_Construction_Machinery_and_Accessories
    - Building_and_Construction_and_Maintenance_Services
    - Chemicals_including_Bio_Chemicals_and_Gas_Materials
    - Cleaning_Equipment_and_Supplies
    - Commercial_and_Military_and_Private_Vehicles_and_their_Accessories_and_Components
      - Aerospace_systems_and_components_and_equipment
      - Aircraft
        - Civilian_and_commercial_rotary_wing_aircraft
        - Military_fixed_wing_aircraft
        - Military_rotary_wing_aircraft
        - Powered_fixed_wing_aircraft
          - Cargo_jet_aircraft
          - Cargo_propeller_aircraft
          - Commercial_passenger_jet_aircraft
          - Commercial_passenger_propeller_aircraft
          - Fixed_wing_agricultural_aircraft
          - Private_or_business_jet_aircraft
          - Private_or_business_propeller_aircraft
          - Seaplanes
        - Recreational_aircraft
        - Specialty_aircraft
      - Marine_transport
      - Motor_vehicles
      - Non_motorized_cycles
      - Railway_and_tramway_machinery_and_equipment
      - Spacecraft
      - Transportation_components_and_systems
      - Transportation_services_equipment
      - Vehicle_bodies_and_trailers
    - Defense_and_Law_Enforcement_and_Security_and_Safety_Equipment_and_Supplies
    - Distribution_and_Conditioning_Systems_and_Equipment_and_Components
    - Domestic_Appliances_and_Supplies_and_Consumer_Electronic_Products
    - Drugs_and_Pharmaceutical_Products
    - Editorial_and_Design_and_Graphic_and_Fine_Art_Services
    - Education_and_Training_Services
    - Electrical_Systems_and_Lighting_and_Components_and_Accessories_and_Supplies
    - Electronic_Components_and_Supplies
    - Engineering_and_Research_and_Technology_Based_Services
    - Environmental_Services
    - Farming_and_Fishing_and_Forestry_and_Wildlife_Contracting_Services
    - Farming_and_Fishing_and_Forestry_and_Wildlife_Machinery_and_Accessories
    - Financial_and_Insurance_Services
    - Food_Beverage_and_Tobacco_Products
    - Fuels_and_Fuel_Additives_and_Lubricants_and_Anti_corrosive_Materials
    - Furniture_and_Furnishings
    - Healthcare_Services
    - Industrial_Cleaning_Services
    - Industrial_Manufacturing_and_Processing_Machinery_and_Accessories

Figure 19 Taxonomy of Product and Services based on the UNSPC

6 Software Systems Integration and Implementation

This section describes the software systems that were used to experiment with semantic web and advanced computer graphic technologies for PLM applications. Different computer systems were integrated using application programming interfaces (API). Others were developed in Java and C++ forming an experimental system interconnected through the semantic web.

Protégé 4.0 is an open source development environment for semantic web applications (Knublauch, Fergerson, Noy, & Musen, 2004) that was used as a platform for ontology development. To derive inferences, Pellet, a complete and capable open source OWL-DL reasoner (Sirin, et al., 2007), and Racer, a core inference engine for the semantic web (Haarslev & Moller, 2003), were used.

Experimental software programs for semantic web automation and ontology visualization were developed using Eclipse Java development platform. The interactive 3D collaboration system was developed using Microsoft Visual Studio C++ and the Open Graphics Library (OpenGL).

6.1 Semantic Web Automation and Visualization System

A semantic web automation and visualization system was developed in Java to automate some information parsing operations. Advanced 2D graphics were implemented using Java2D. This allowed the creation of different graphical representations for the information. The system architecture is presented in figure 20. The OWL API is used as an interface to the Web Ontology Language (Horridge, Bechhofer, & Noppens, 2007). It provides parsing, reading, and in-memory reference implementation of OWL. Semantic web software such as Protégé, Pellet and Racer can work together with OWL ontology or the OWL API. JRacer is a java based API that was used to interface directly with a Racer server. Text file can be read, parse, or written using standard Java file input/output (I/O) functions.

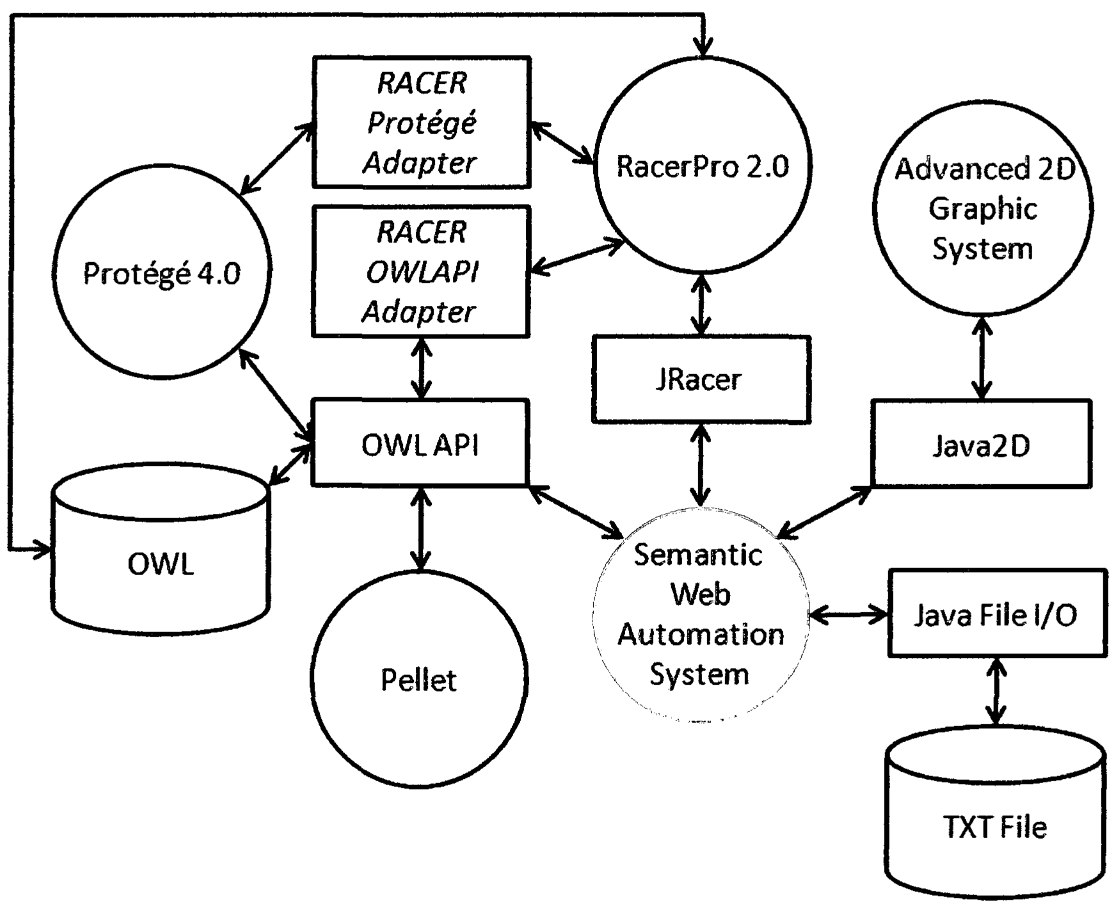

Figure 20 Semantic Web Automation and Visualization System Architecture

6.1.1 Information Parsing and Encoding

To facilitate the development of ontologies for PLM, a software system that can parse text documents and encode into OWL was developed. This system was used to create a taxonomy of over 40 000 products and services based on the United Nations Standard Products and Services Code (UNSPSC). The UNSPSC is a hierarchy of commodities that can be used by any organization to classify products and services. The free UNSPSC codeset, which is only avalaible in PDF format, was converted to a text document. A Java program was written to read the text document and to create a product concept for each UNSPSC.

The system was also used to parse information from *Product Lifecycle Management* by John Stark (2004). Information presented in point form about PLM was extracted from the book by scanning the pages. Optical character recognition (OCR) software was used to convert the scanned images into a separate text document for each topic. The text document required manual editing to ensure a consistent formatting. The topic concept such as "PLM Benefits" or "PLM Vendors" was written on the first line of the document. The following lines contained point form information that was encoded as individuals belonging to the topic concept.

6.1.2 Java2D Force-based Graph

A program to visualize ontologies was developed in Java using the Java 2D API. RacerPro 2.0 is the engine used to read, write, and reason OWL ontologies. The purpose was to develop a platform that could be used to generate custom graphs to facilitate the understanding of ontologies. Force based algorithms developed by Fruchterman and Reingold (1991) were implemented to graph the concepts of OWL ontologies. Elastic forces pull concepts that are related together and magnetic forces push concepts apart so that they do not overlap. Those forces move concepts toward their parent(s) and away from neighboring concepts. This type of graph can helps to understand the relationships between different concepts. Figure 21 shows the force based graph applet. The automatically generated graph represents the PLM knowledge extracted from *Product Lifecycle Management* (Stark, 2004). Additional control mechanisms are shown in the left hand side. This platform can be used to experiment with custom ontology visualization.

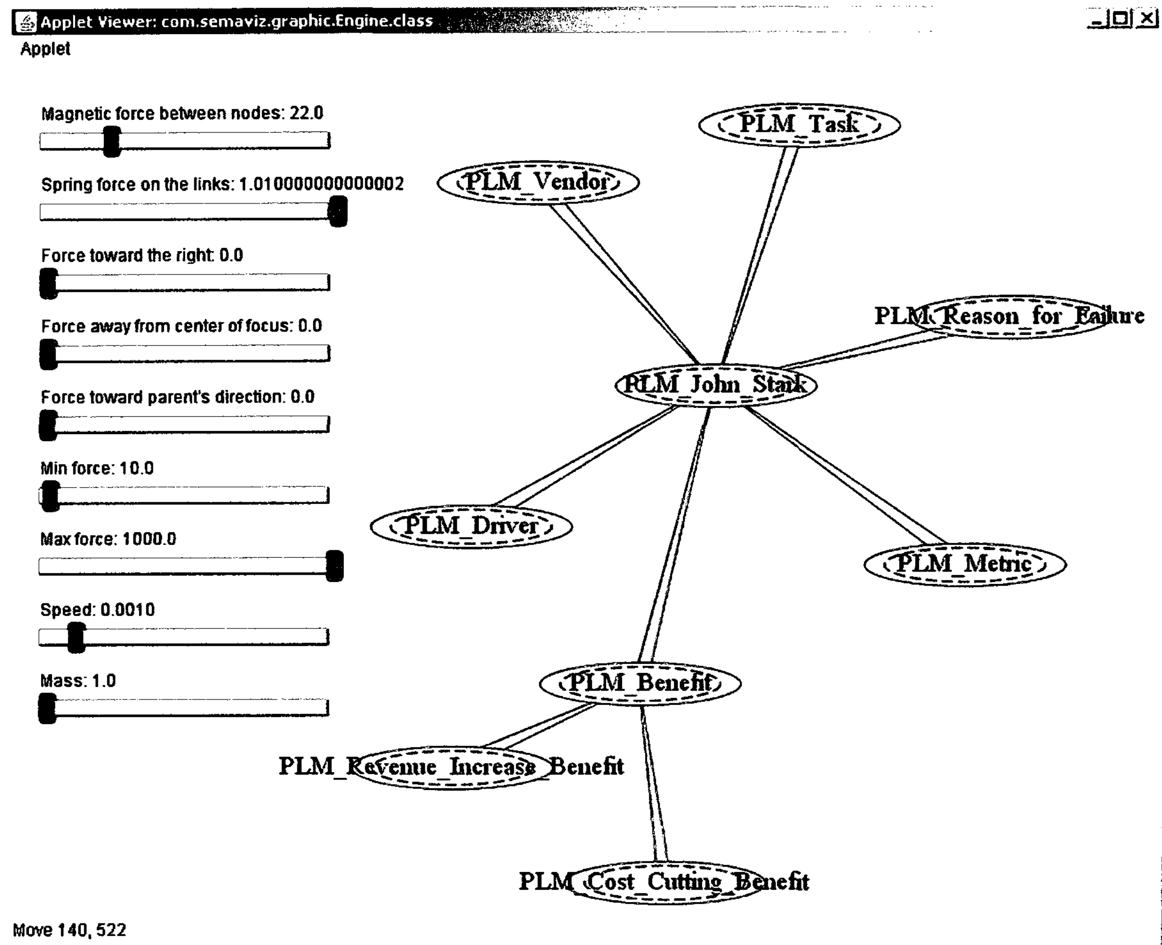

Figure 21 Force-based Graph Applet

6.2 Interactive 3D Collaboration System

The interactive 3D collaboration system was developed in C++ using OpenGL to provide accelerated 3D graphics for interactive virtual reality collaboration. The OpenGL Shading Language (GLSL) was used to program experimental shaders for protected remote rendering collaboration. This platform could be further extended to a VR immersive environment that uses stereoscopic rendering, motion tracking, and bio feedback. The basic system architecture is presented in figure 22. GLUT, a window system independent toolkit for writing OpenGL programs was used to ensure cross platform compatibility. FreeType fonts are rendered using the FTGL library. 3D models encoded in .3ds format can be read and written using the lib3ds library. SOIL is a simple image library used for reading and writing images. Standard C++ file input/output (I/0) functions were used to read and write text files that contain the collaboration context information.

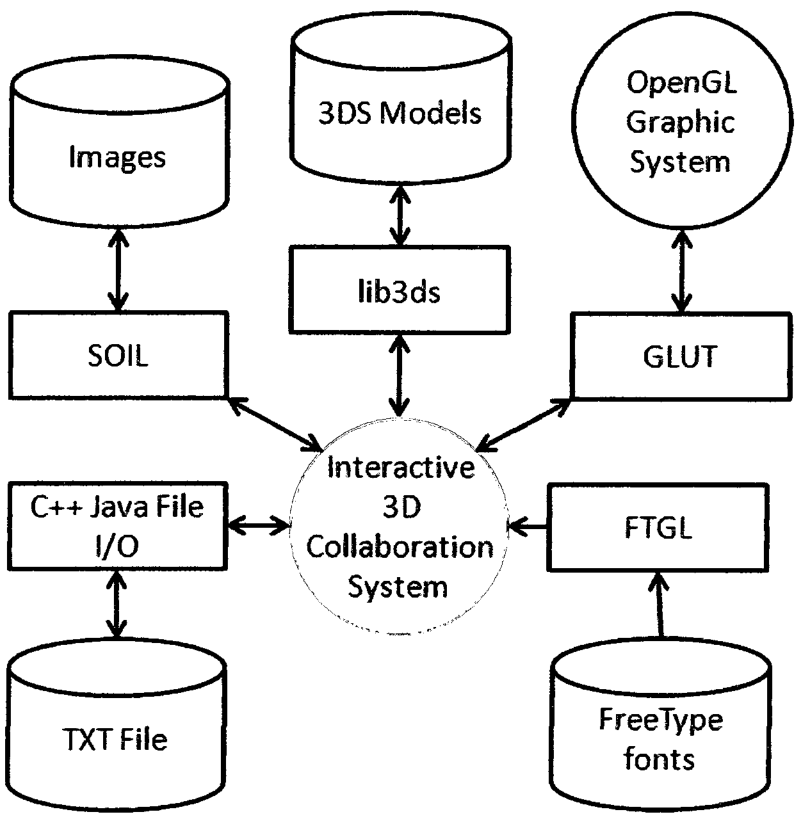

Figure 22 Interactive 3D Collaboration System Architecture

6.2.1 OpenGL 3D Visualisation System

A simple 3D visualization system, shown in figure 23, was developed to study the possibilities of integrating semantic web technologies with advanced computer graphics systems. The model's hierarchical structure is shown in a vertical tree. The 3D model can be manipulated using the mouse and the camera can be moved with the arrow keys. The keys F1-F4 are used to create and navigate tasks and the keys F5-F8 are used for views. The visualization context can be created manually, by specifying the 3ds model to load, creating visualization tasks and views. This information can be saved in a text file (F9) which can be encoded in OWL using the Java system described in section 6.1. Similarly, task and view information encoded in OWL can be encoded in a text format that can be read by the OpenGL system. Using OWL as a data exchange format enables system interoperability.

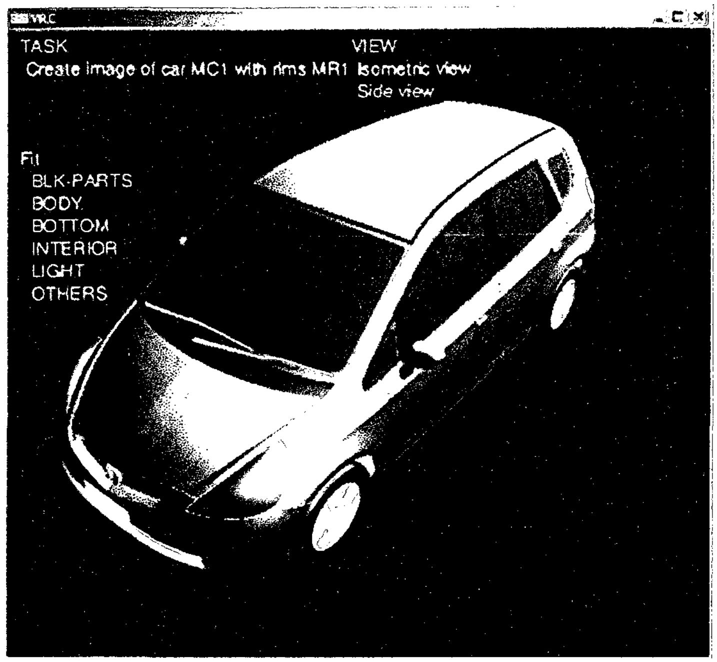

Figure 23 User Interface for the Interactive 3D Collaboration System

7 Applications of Semantic Web for PLM

The possibilities of semantic web technologies are infinite. The increasing number of research papers and books on the subject suggest that we have just begun to scratch the surface and that there is still much more that can be done.

Protégé 4.0 and TopBraid composer software applications were used to experiment with OWL DL ontologies for PLM. The inference engines Pellet and Racer were used to compute semantic web inferences; in other words, deduce knowledge from logical implications. The purpose was to demonstrate applications of semantic web-based systems for PLM. This section demonstrates applications for information retrieval, step by step process support, design and manufacturing support.

7.1 Information Retrieval

The ability to find the right information is essential for effective collaboration. In order to fully leverage the enterprise knowledge, it must be retrievable. Virtual assets that are lost in the sea of information are wasted assets with far reaching implications such as generating wasted time, wasted resources, increased risk of errors, increased costs, and missed opportunities. This section demonstrates some of the possibilities of description logic (DL) searches by using Protégé 4.0 and Pellet.

To show the capabilities of ontology-based searches, DL queries are used to retrieve specific PLM information. Figure 24 shows the results of the query for "potential PLM benefits". All of the concepts and individuals that are found satisfy the query axiom.

Therefore, if the information in the ontology is valid, all of results should be valid answers to the query. To further narrow the results, a more specific DL query can be constructed.

Furthermore, semantic links can provide additional relevant information, as shown in figure 25. Another example is shown in figure 26 where a search for "organizations that sell products for PLM" returns a list of all the PLM vendors that was encoded in OWL in section 6.1.1.

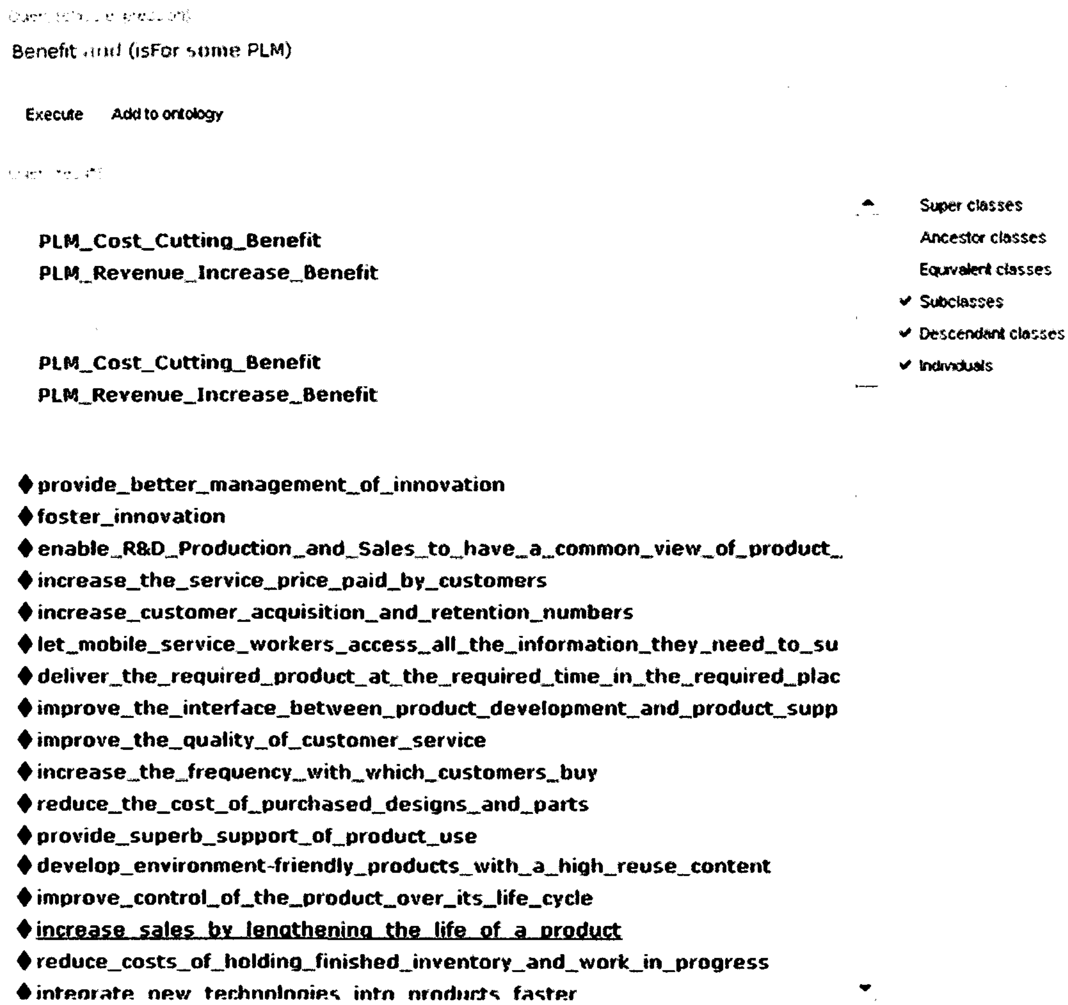

Figure 24 Searching for PLM Benefits using a DL Query in Protégé 4.0

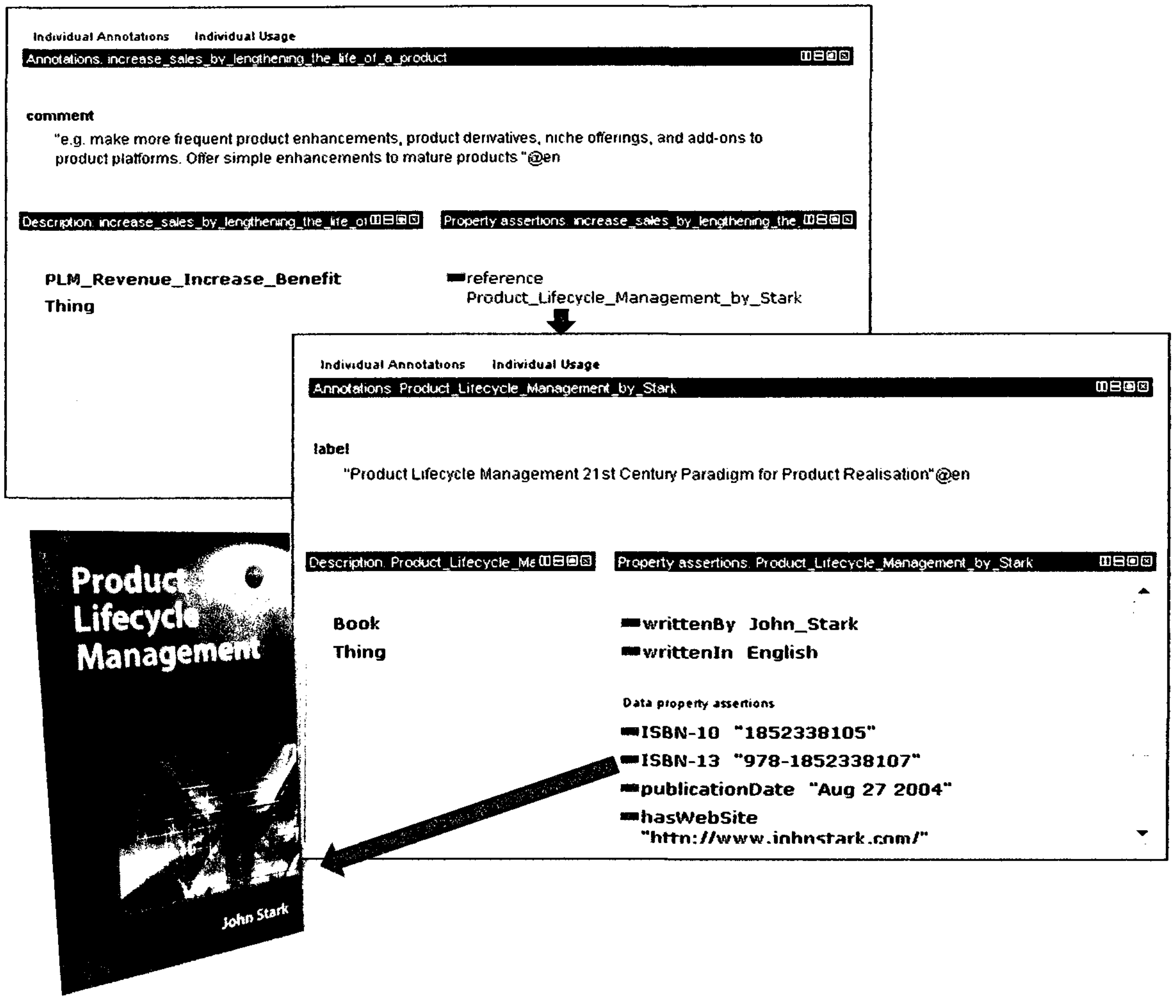

Figure 25 Semantic Description of a PLM Benefit Linked to Source

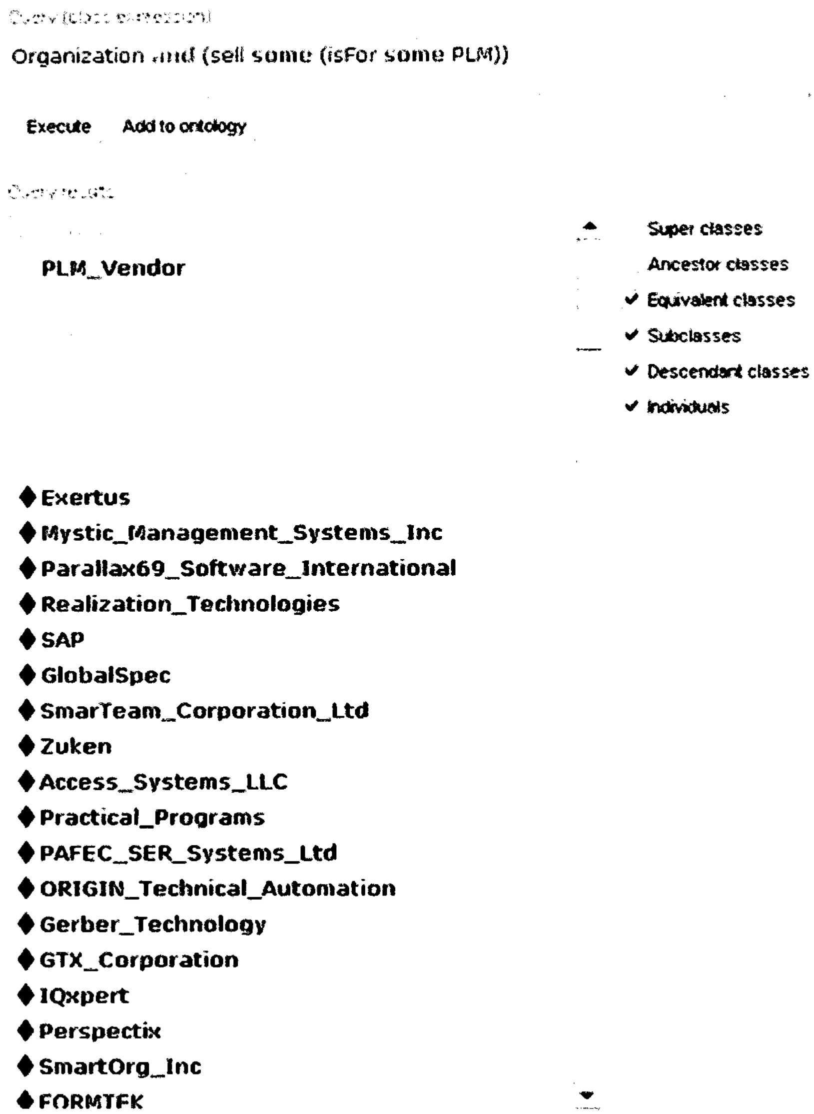

Figure 26 Searching for Companies that Sell Products for PLM with Protégé 4.0

7.1.1 Result Accuracy Compared to Popular Search Engines

The main advantage of semantic web searches is that they provide the right information, not ranked information. While ranked search engines such as Google and Bing are great to surf the internet, they are not always useful for finding precise information inside a specific domain. This is mainly due to the fact that most major topics on the internet have been searched and referenced to by millions of people. Search engines use this type of information to rank the most popular results first, providing more visibility to the most referenced results. Many enterprise environments have information encoded in a multitude of different formats, therefore current search engines can be relatively inefficient. To demonstrate the difference between right and ranked search result, the "vision of Concordia university of Montreal" was searched using Google Search, Microsoft Bing, and Protégé DL query powered by Pellet. Figure 27 shows that a Google search of the main keyword provide 25,700 results. Unfortunately, none of them seem to lead to the right information. Even more alarming is the fact that the first result is the vision and chronology of the Grey Nuns Mother House that was bought by Concordia. Since the information comes from the right source, it could be misinterpreted as the right information. In an attempt to obtain more accurate results, semantic relationships were added to the query of figure 28. Since Google Search is not designed to work with semantic relationships, the number of bad results doubles to 52,600. Even searching within the Concordia's domain using the Google powered search engine provides very poor results as shown in figure 29. Microsoft's new search engine Bing seams to provide similar results as shown in figure 30. Nevertheless, the 11th result does provide a linked to the right web page where the information can be retrieved. Note that searches usually provide links to some web pages where the information might be located and not the actual information. The user has to read through the web pages in order to determine if the information is actually there. In contrast, semantic searches can provide the right information directly. Figure 31 shows that Concordia's vision can be found using a simple DL query and that it has been documented in the Reaching Up, Reaching Out document.

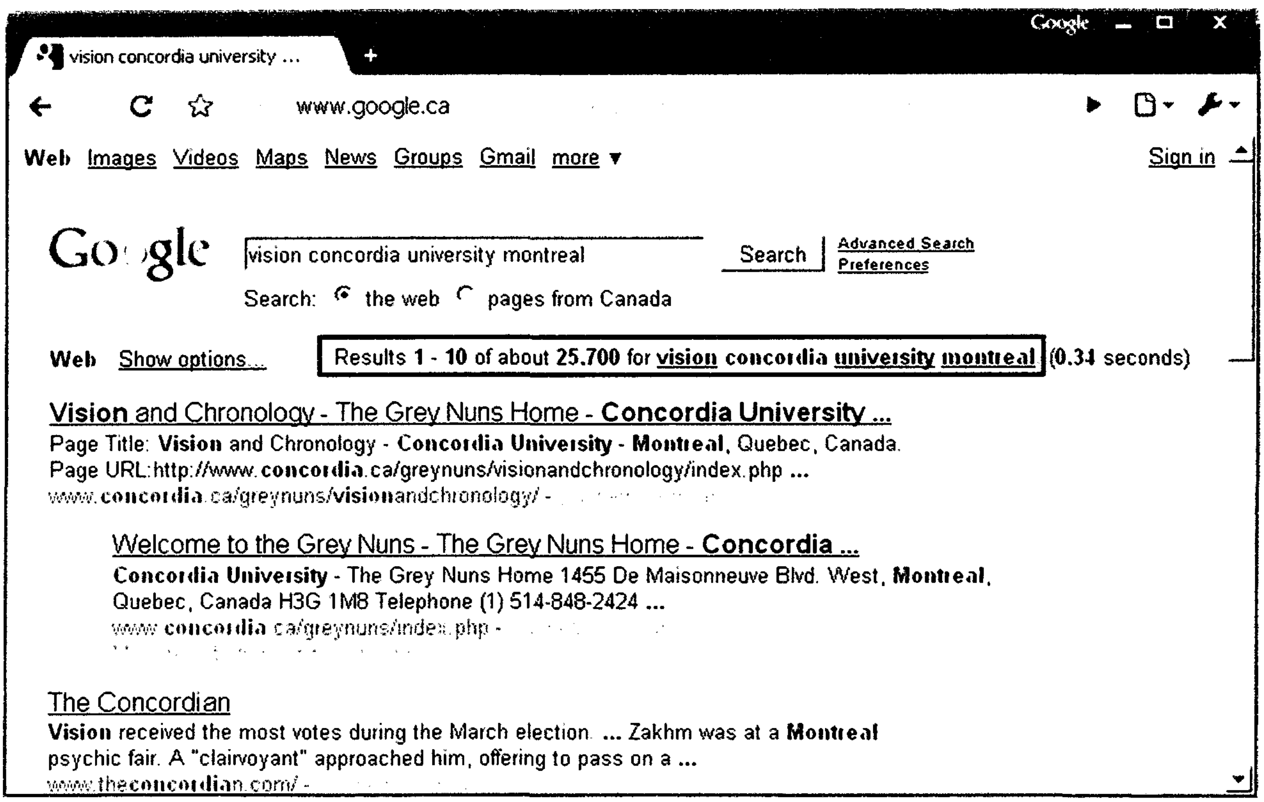

Figure 27 Google Search using Keywords

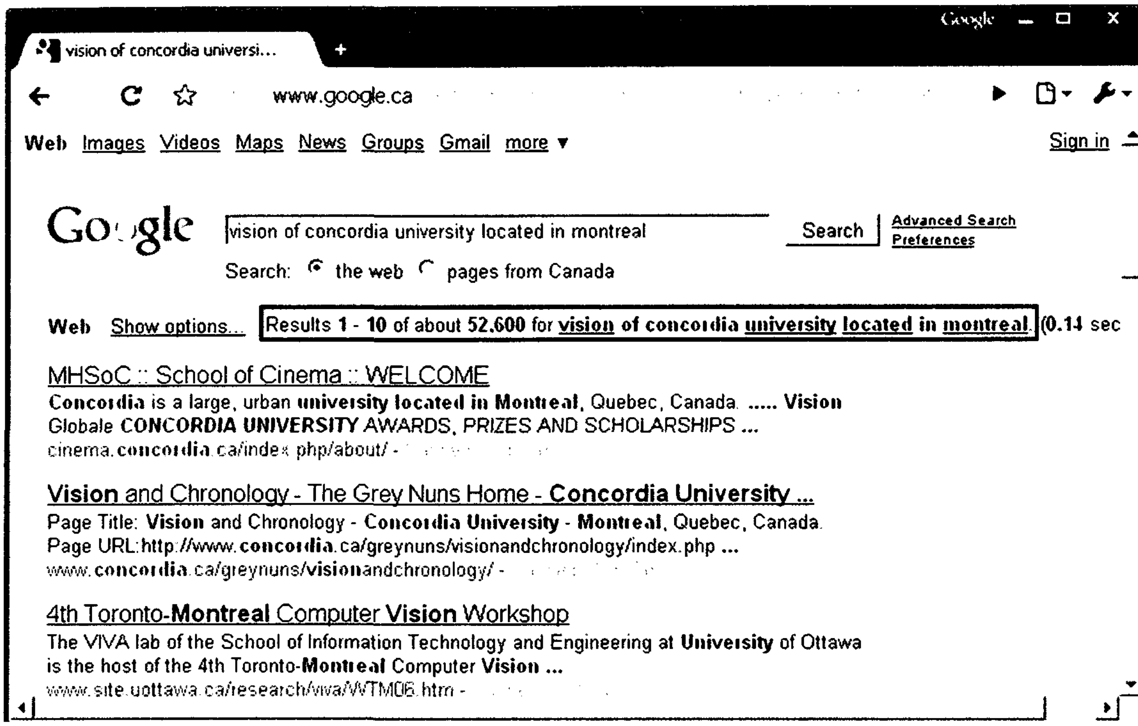

Figure 28 Google Search with Linked Keywords

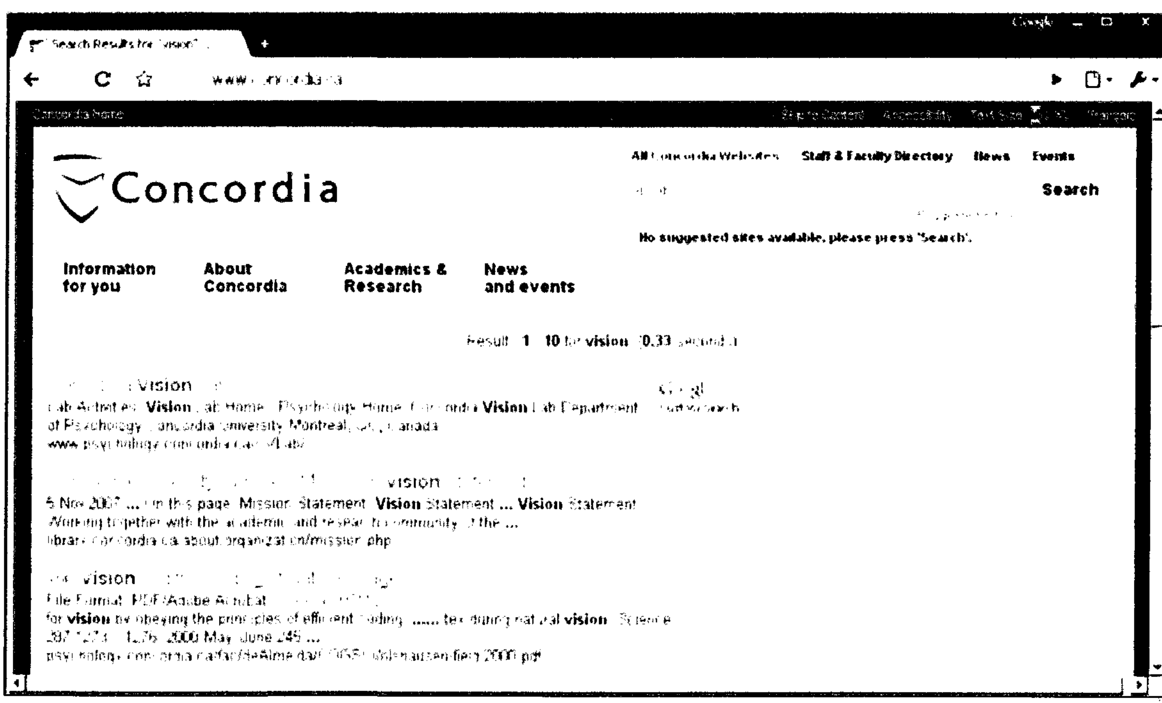

Figure 29 Searching Concordia's Google Powered Search

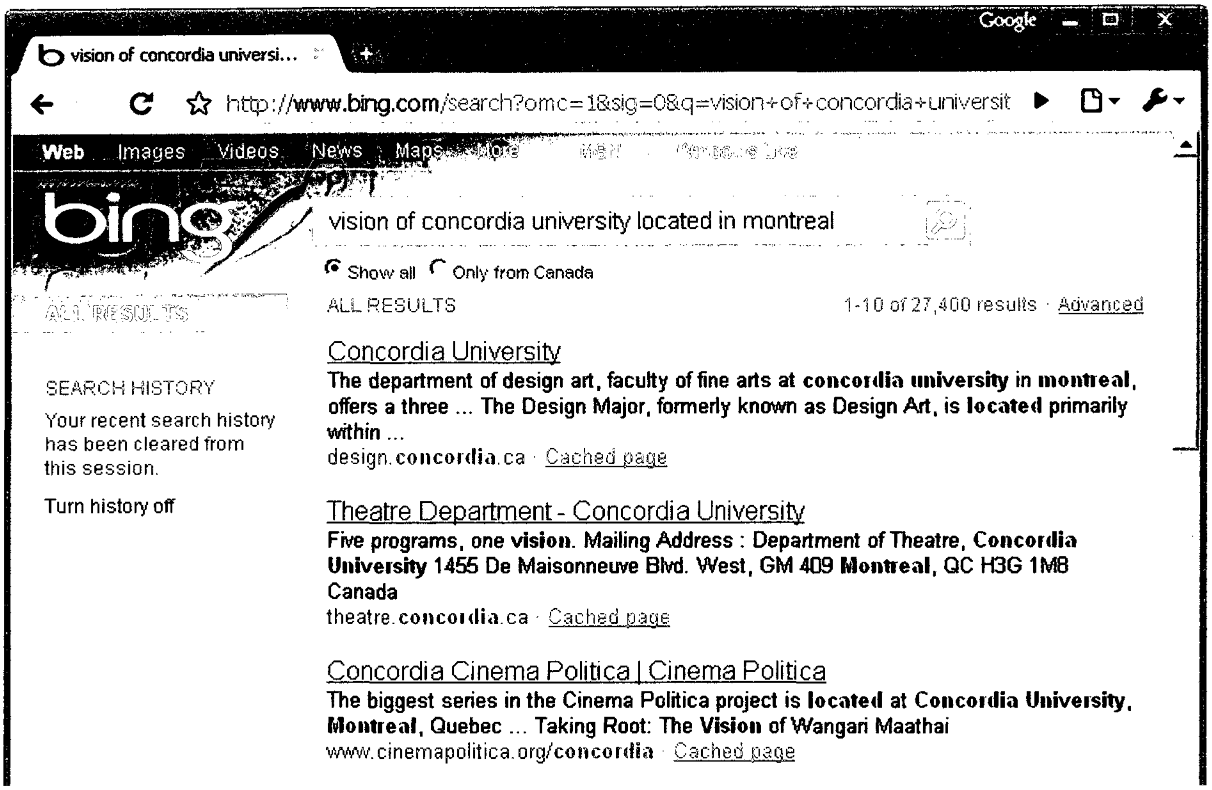

Figure 30 Searching with Bing

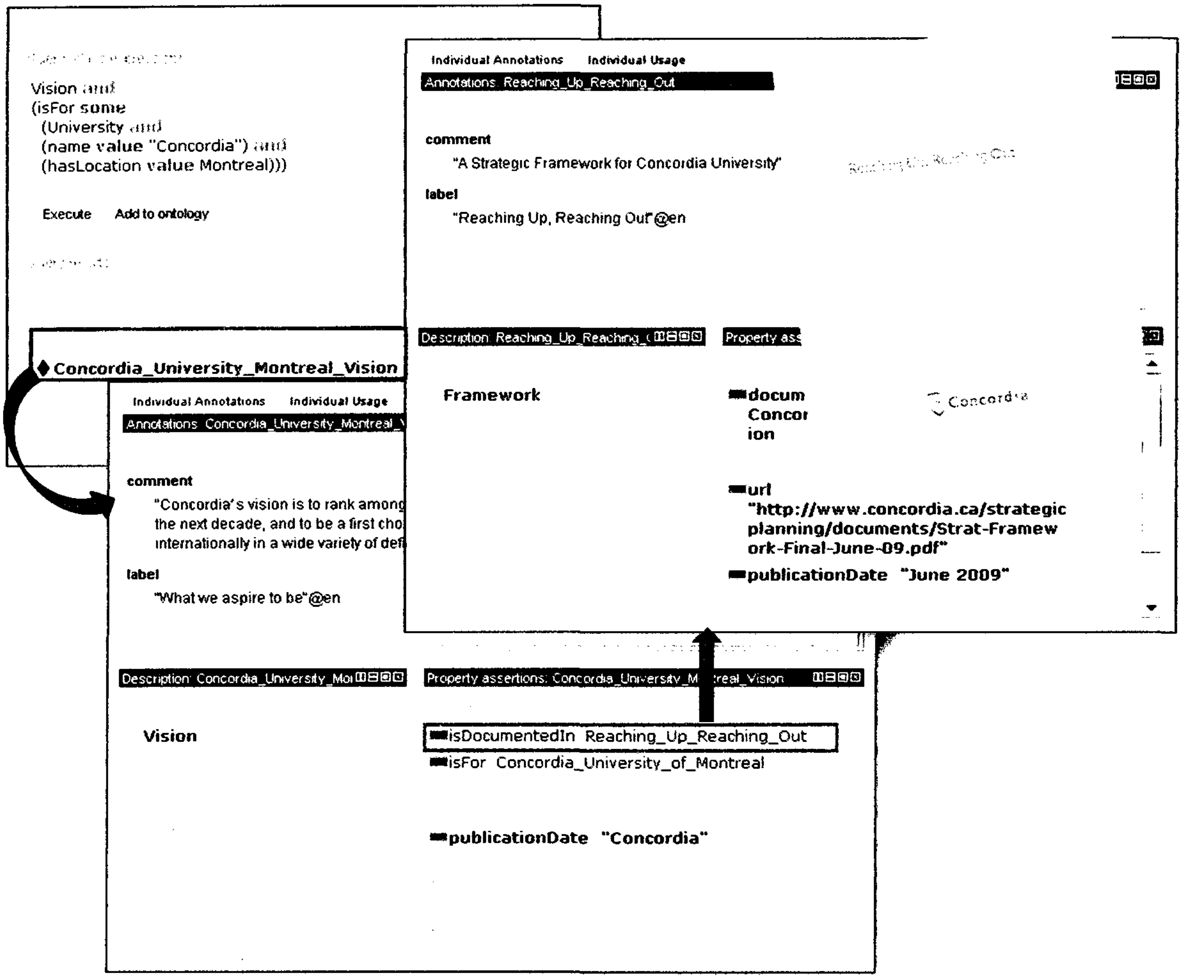

Figure 31 Retrieving the Information about Concordia's Vision

7.2 Step by Step Process Support

A good way for companies to leverage and improve their knowhow is to define standard processes. Knowledge and experience about activities can be used to define optimal processes which can ensure a certain level of quality, safety, and repeatability. Capturing the knowledge of experienced workers effectively is particularly important given the impending wave of baby boomers retiring. Semantic descriptions of processes can be used to offer step by step support within the context of the user.

Real-time step by step process support is demonstrated using Protégé 4.0 in figure 32. An employee assigned to product planning can access the standard product planning process relevant to his domain automatically. Additional information about the process such as the specific steps, documentation, the person responsible, or the required tools can be easily retrieved by following the semantic links. Furthermore, different representations can be inferred and proposed to the user. In this example, the inferred Active_Flowchart_Drawing individual indicates that there exist a drawing application that can draw a flowchart for this process as shown in figure 32.

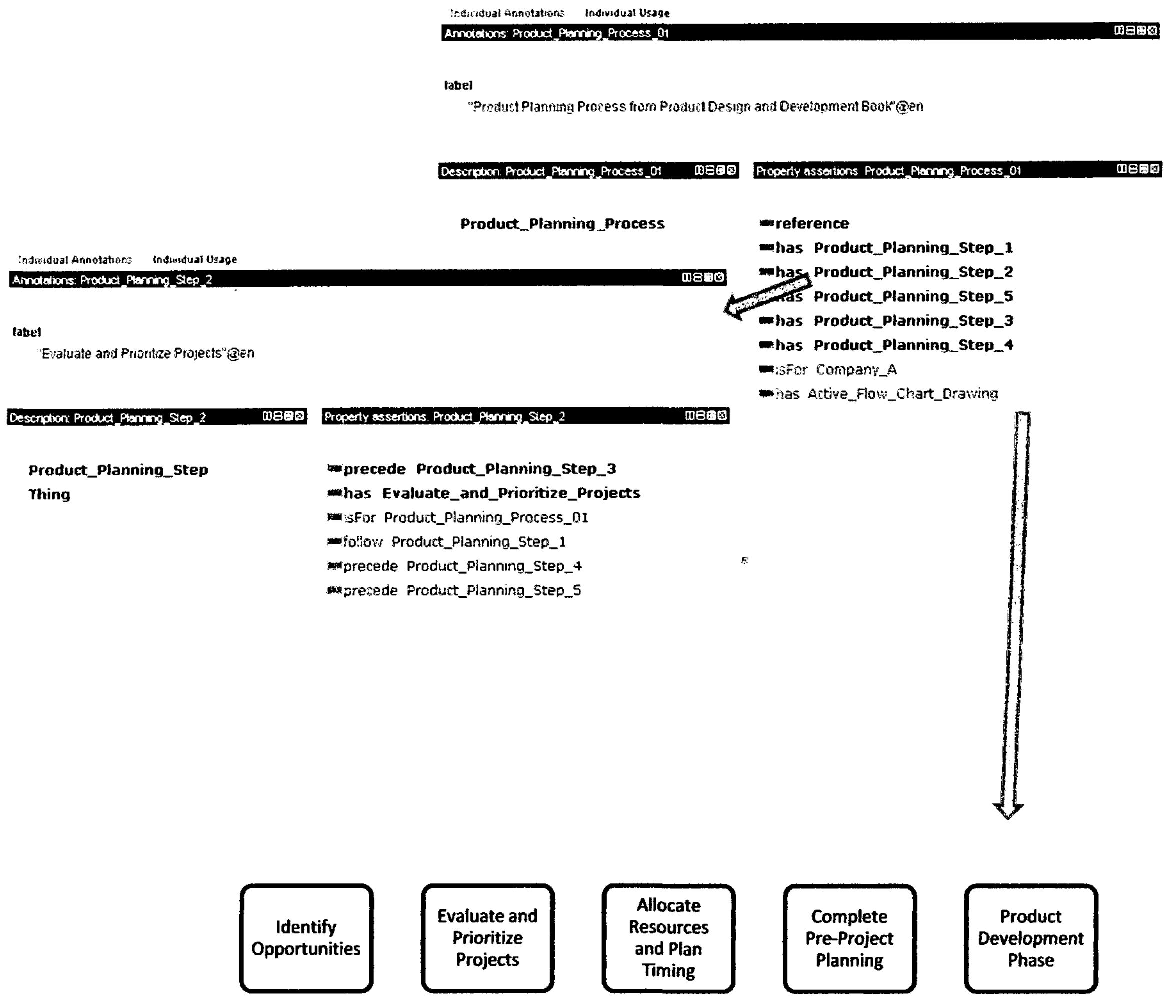

Figure 32 Product Planning Process

7.3 Design and Manufacturing Support

Semantic relationships between design, manufacturing, and material information can be captured and encoded in OWL. This information can be used to provide real-time design and manufacturing support. For example, when a designer chooses to use a certain material, some of the physical and manufacturing implications could be inferred using a semantic web reasoner such as Racer or Pellet.

An experimental PLM ontology for a company that design and manufacture tables was developed using TopBraid Composer. Figure 33 shows an OpenGL 3D rendering of two possible table designs with support information for each design. The support information panels are taken from the TobBraid graphical interface. The asserted properties are shown in rectangular boxes with white backgrounds. The inferred properties that were computed by Pellet have a light blue background.

Manufacturing processes, machine requirements, and tooling necessities are inferred from basic shape and material information. Also, expected behavior of the product can be deduced from material specifications. Using product design and manufacturing domain ontologies, it is possible to make logical inferences that can be used to develop better products.

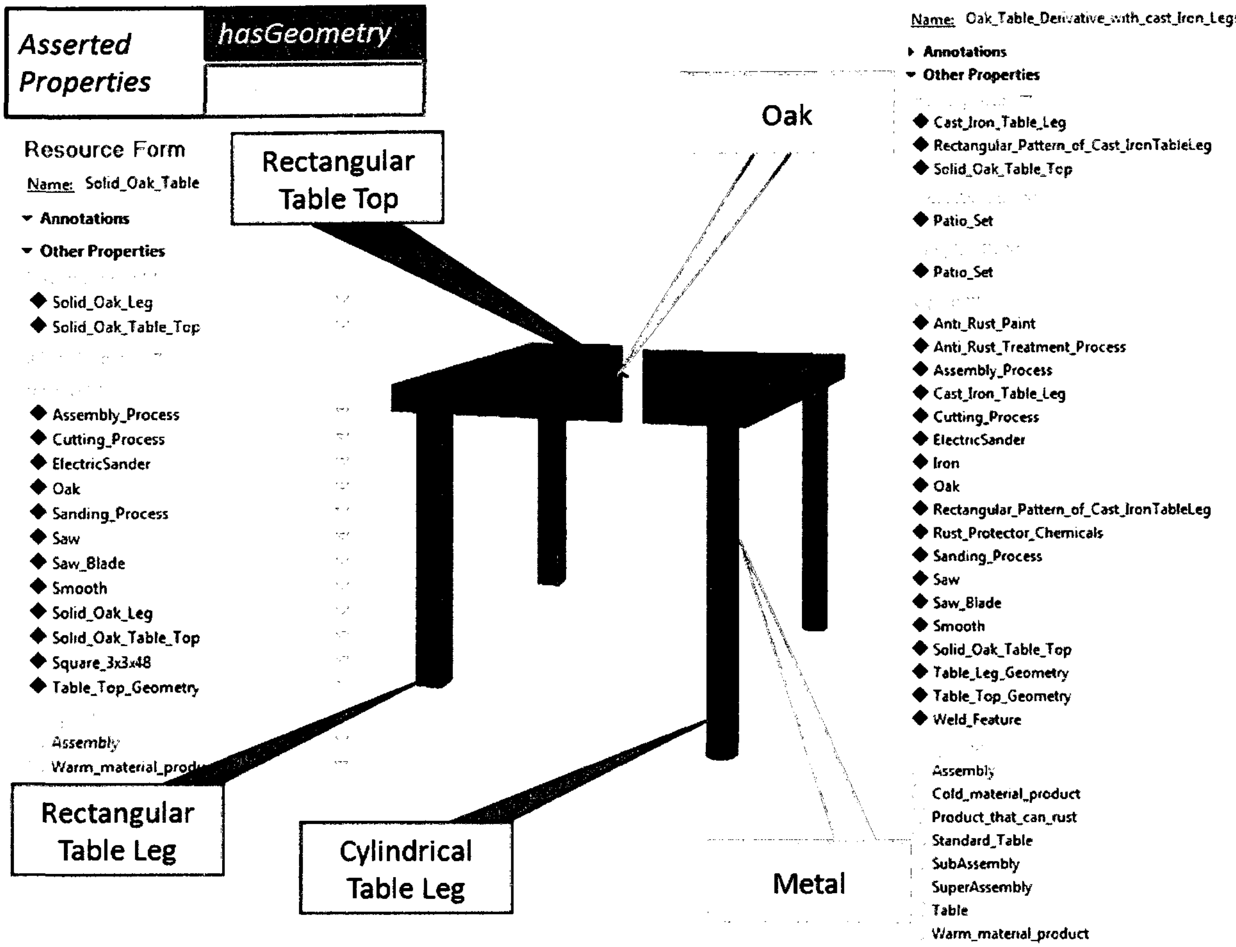

Figure 33 Design and Manufacturing Information for an Oak Table

8 Protected Remote Rendering Collaboration

Software solutions now offer the possibility of virtual reality collaboration. The virtual product mockup can be visualized, analyzed, optimized, and understood. Virtual reality collaboration environments are great enablers for global collaboration. Unfortunately, the high risk of losing intellectual property when collaborating within immersive environments prevents widespread adoption of VR collaboration. Information leaks are a major concern to most companies, especially when they are involved in high technology product development. A novel 3D collaboration approach that protects 3D virtual assets via remote rendering and the service of a trusted shared mask broker has been developed. The novelty of the proposed approach is that companies can collaborate effectively on shared digital mockups without having to share their 3D models. Semantic web technologies are used to provide the views, 2D images, that show the required information for specific tasks, thereby eliminating the need for navigation. Collaborators only exchange the necessary graphical information to create accurate renderings that can be used for engineering collaboration. This section describes the technique, the semantic descriptions, and the automated process for remote rendering collaboration.

8.1 Contribution of Literature Review

The remote rendering system presented by Koller, et al. (2004) is extended with the ability to combine images from different sources. Mikovec et al. (2006) used a set of scalable vector graphic images (SVG) and semantic descriptions of the 3D environment to facilitate interaction and navigation in a mobile environment.

The proposed approach offers the ability to collaborate interactively in a 3D environment by only sharing the required information. Unlike other security techniques based on reducing the level of detail and modifying the geometry such as the one described by Cera, et. al (2004), this approach will not result in engineering errors due to misrepresentation. A 3D model for a circular part that is simplified to a level where it looks like a hexagon can seriously impede engineering collaboration.

8.2 Remote Rendering Collaboration Technique

The problem of deciding which pixel of the rendered image needs to be shared can be solved through the use of a trusted broker that will compare the depth of every pixel and return a mask that defines which pixels need to be shared. This approach is illustrated in figure 34 where a car manufacturer, company A, would like to offer their customer the option of custom rims from company B. In order to ensure the look, fitting, and tolerances, some critical views of the digital mockup are exchanged. The illustration shows the different phases of the collaboration process. Both companies send a depth map, which represents the depth of every pixel in the picture to a trusted broker. The trusted broker compares depth maps and generates a shared mask for each company. The shared masks are sent back to the respective companies who use it to remove unnecessary visual information and create their part of the shared image.

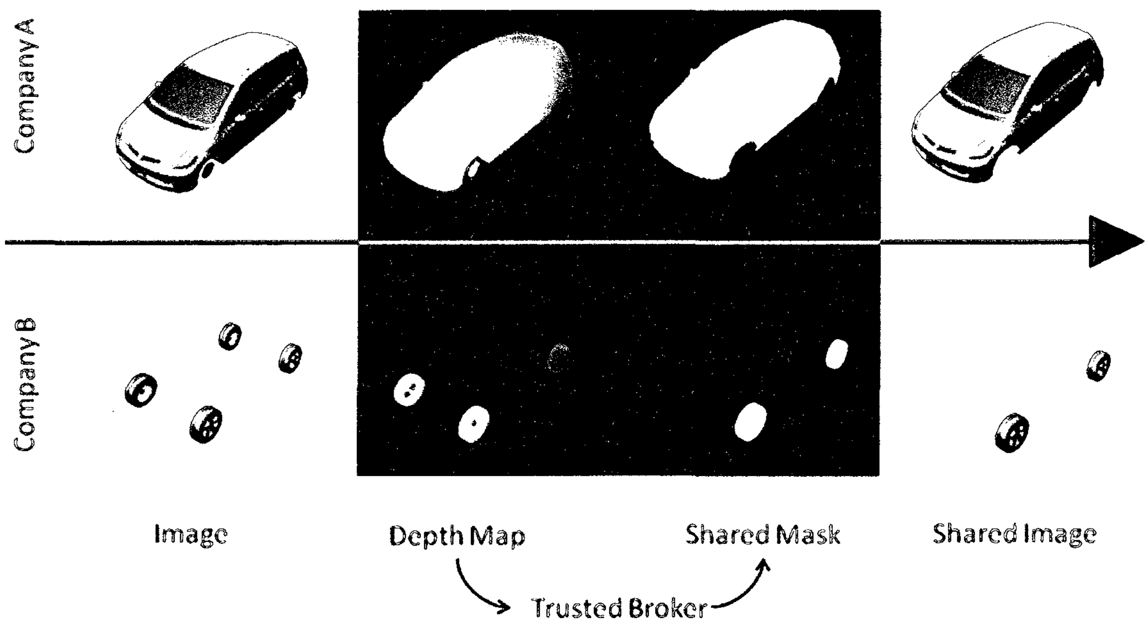

Figure 34 Using a Trusted Broker to Generate Shared Masks.

8.2.1 Step 1: Render Images

Once the 3D model is loaded into the scene, the system iterates through the required views and renders the 3D model. Images are rendered using standard OpenGL functionalities. The OpenGL smooth shading model is used to produce a realistic image with smooth transition between the flat tessellations of the 3D model. The rendered image is saved as a .bmp file. Different shading models could be implemented to produce different rendering styles such as black and white with hidden lines for rendering technical drawings.

8.2.2 Step 2: Extract Depth Maps

A depth map is used to represent the depth value of every pixel in a rendered image. It is computed automatically when a rendered image is saved. This information is required to know which pixels are seen in the complete assembly. The depth map for an orthogonal view of company A's model is shown in figure 35. This implementation only uses one color channel which results in a grayscale image representation of the depth. The depth of every pixel is extracted from the OpenGL's depth buffer and saved as a grayscale image with an integer value between 0 and 255. White has a value of 255 and is the closest, while black has a value of 0 and is the farthest. The source code used to produce depth maps is presented in appendix A. To produce more accurate depth maps, more color channels can be used. For example using three color channels would provide a range of 16,777,216 (256^3) possible depth values.

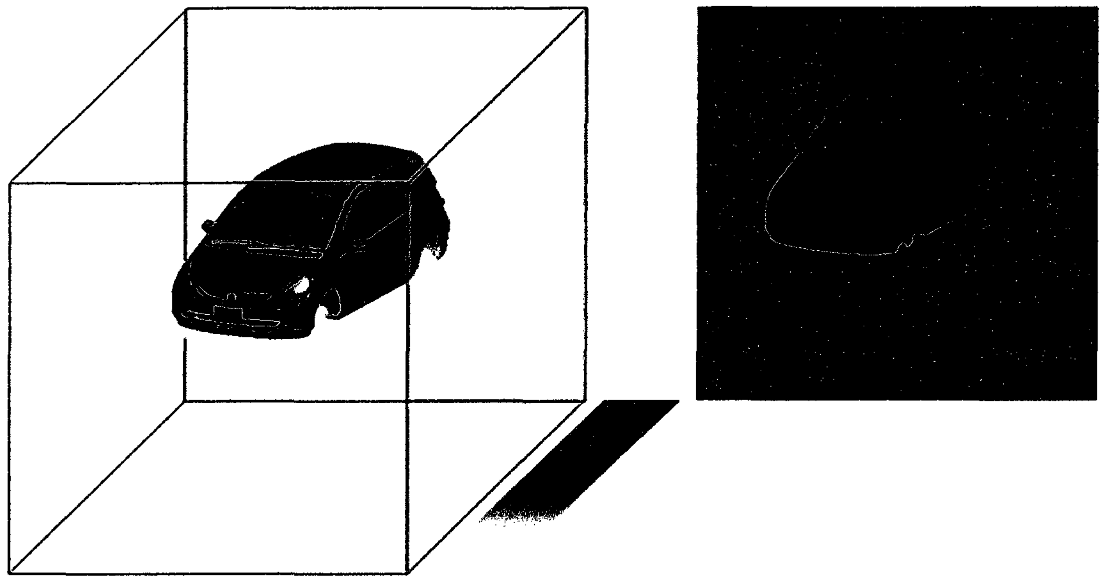

Figure 35 Orthogonal Projection Depth Map for of a Car Body

8.2.3 Step 3: Compute Shared Masks

The shared masks are used to identify which pixels are not needed for the final shared images. Shared masks are computed by comparing the depth map of every collaborator in order to identify which pixels are in the line of sight for the view. The operation of comparing the depth maps of different collaborators can be done by a trusted external party called the trusted broker. This computer agent compares the depth value of every pixel and generates a black and white mask for each collaborator. The GLSL fragment program used to generate the shared masks is presented in appendix B.

8.2.4 Step 4: Compute Shared Images

Shared images consist of image layers that represent the portion of the complete image for each collaborator. Similar to a jigsaw puzzle, when all of the shared images are combined together, the complete image appears. Shared images are created by removing the masked areas identified in the shared mask from the initial rendered image. The GLSL fragment program used to generate the shared images is presented in appendix C.

8.2.5 Step 5: Compute Complete Images

The final step consists of combining the colored pixels of each shared image. Since each shared image shows a different region, combining the shared images creates the complete image. Figure 36 shows the complete image for the example used to demonstrate the remote rendering collaboration technique. The GLSL fragment program used to combine the shared images is presented in appendix D.

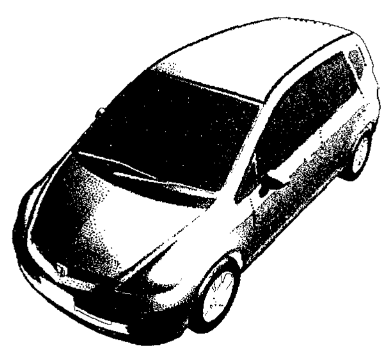

Figure 36 Complete Image

8.3 Semantic Description of Image based Collaboration

A semantic description of the collaborative environment can be used to increase the effectiveness of the remote rendering collaboration technique. While 2D images can help protect 3D virtual assets, the issue of which image to look at can complicate the collaboration process. This valuable information can be encoded into an ontology as shown in figure 37.

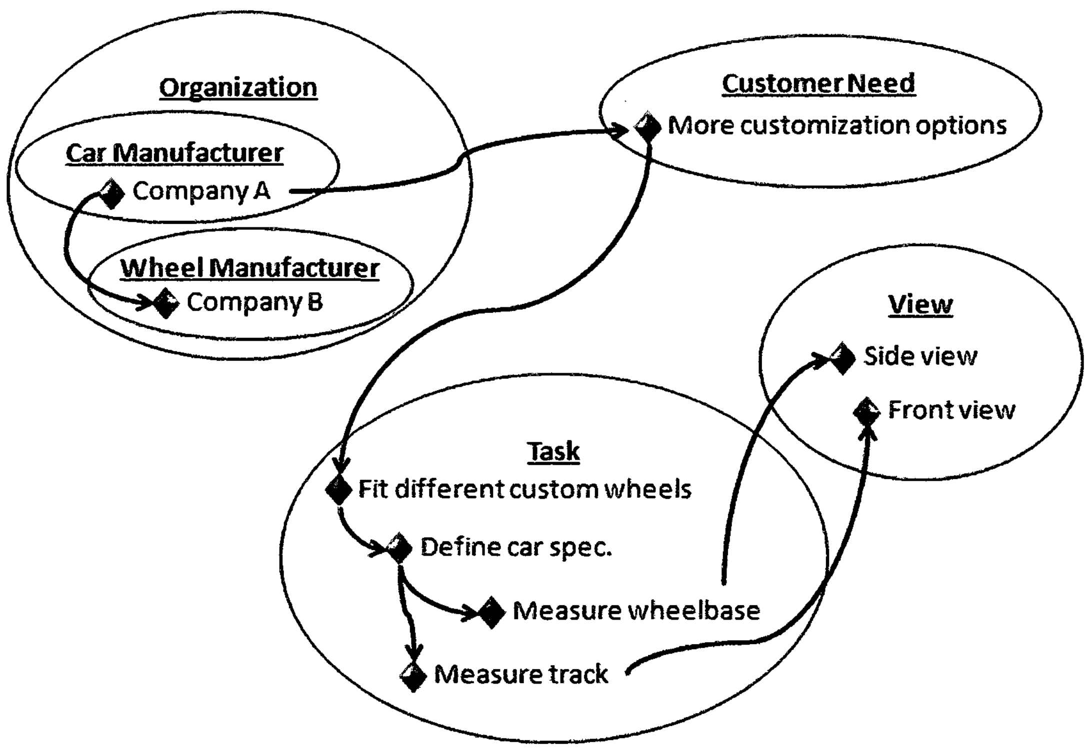

Figure 37 Illustration of Semantic Relationships for Image based Collaboration

The semantic relationships between tasks and views can be used to load or request the views required for a specific task. The ontology shown in figure 37 is encoded in an OWL ontology. The Java automation system can read the ontology and generate text files that contain the 3D collaboration context. For example, if company A identified a customer requirement for more customization options, their system could propose the task of fitting different custom wheels and collaborating with wheel manufacturers one of which is company B. Because semantic web ontologies are used to record the information about the collaboration process, different ontology based techniques could be implemented to provide additional functionality and security.

8.4 Automated Remote Rendering Collaboration

The decision to accept the exchange of information can be automated using the service of a reasoner that computes logical clashes with the security policy of the organization. This approach is illustrated with a unified modeling language (UML) sequence diagram shown in figure 38. The diagram shows the sequence of operations that must be performed by the different collaborators in order to automate the remote rendering collaboration process. First, the request is sent to all the collaborators. If the image request is accepted, all parties send their depth maps to a trusted broker. The trusted broker is a computer agent that compares the depth map of all collaborators and sends back a different shared mask to each collaborator showing which parts of their image will be visible. Every party sends back their part of the picture to create the complete shared image for the requestor.

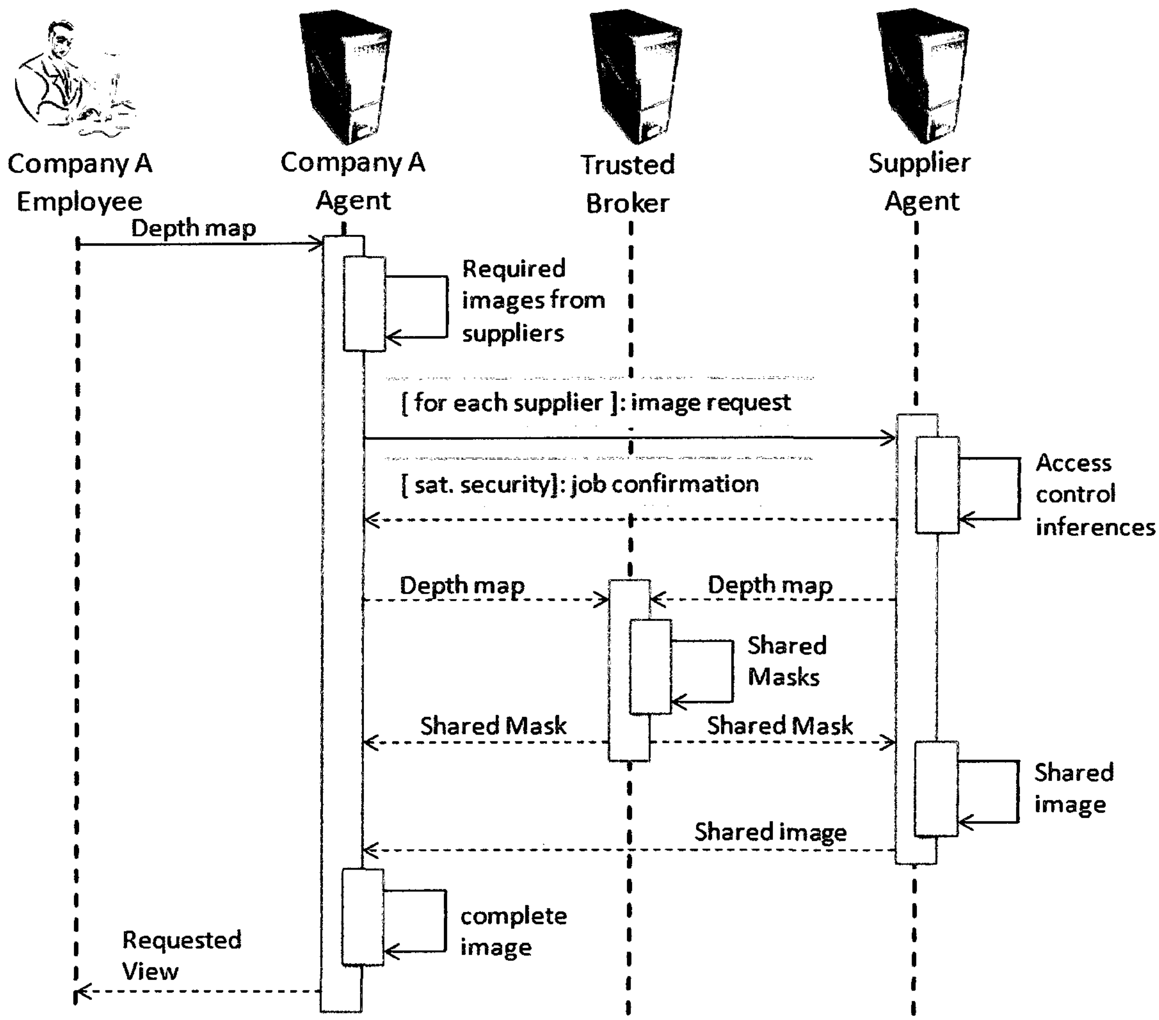

Figure 38 Sequence Diagram: Successful Automated Remote Rendering Collaboration

9 Conclusion

The global market of today's economy requires companies to be more responsive and competitive. Currently factors such as proprietary data formats and vocabularies impede the realization of PLM. The semantic web can be used to facilitate the use of data by providing common data formats and languages that enable open and efficient collaboration. The semantics and syntaxes of W3C languages such as URI, XML, RDF, OWL, SPARQL, and SWRL form the foundation for the semantic web. The use of these languages to describe the environment enables computational analysis (reasoning) that can lead to logical conclusions as demonstrated in the section about applications of semantic web for PLM.

Shared semantic web vocabularies can be used to break down some collaboration barriers between people, machines, and information. Agents and information can be described using semantic descriptions that are written with a shared semantic vocabulary and encoded in an open format. Semantic descriptions are analogous to an identity card that is readable by any person or machine.

A general framework for developing the foundation for a PLM ontology was defined using three phases; identify PLM requirements, define PLM concepts and relationships, and gather information. This framework could help companies develop ontologies that meet their custom needs by focusing on thorough analysis of common questions that employees have within the context of their position. Ensuring a uniform understanding of common terms provides a platform for companywide and even global collaboration.

The semantic web-based PLM experimental system was implemented through the integration of many enabling software technologies such as OWL API, Protégé, Racer, Pellet, Java 2D, and OpenGL. A system to automate information parsing and encoding was developed in Java. Furthermore, a Java 2D user interface was created using force-based algorithms to generate custom graphs. A 3D visualization platform for remote rendering collaboration was developed using C++ and advanced OpenGL graphics.

A novel 3D collaboration approach that protects 3D virtual assets via remote rendering and the service of a trusted shared mask broker was presented. The novelty of the proposed approach is that companies can collaborate on shared digital mockups without having to share their 3D models. Instead, only the "right" views for specific tasks are automatically exchanged, thereby eliminating the need for 3D navigation.

The full implications of semantic web technologies have yet to be revealed. What is certain is that collaboration and communication across language and system barriers are providing optimal conditions for the realization of a holistic approach to PLM.

References

Abdul-Ghafour, S., Ghodous, P., Shariat, B., & Perna, E. (2007). A Common Design-Features Ontology for Product Data Semantics Interoperability. *IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 443-446). Washington, DC, USA: IEEE Computer Society.

Aberdeen Group. (2007). *Spend Analysis: Working Too Hard for the Money.* Aberdeen Group.

Aberdeen Group. (2003). *The Spending Analysis Benchmark Report: Dissecting a Corporate Epidemic.*

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (2003). *The Description Logic Handbook Theory, Implementation and Applications.* Cambridge: Cambridge University Press.

Berners-Lee, T., Fielding, R., & Masinter, L. (2005, January). *Uniform Resource Identifier (URI): Generic Syntax.* Retrieved March 2009, from labs.apache.org: http://labs.apache.org/webarch/uri/rfc/rfc3986.html

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). Retrieved from Scientific American: http://www.scientificamerican.com/article.cfm?id=the-semantic-web&sc=I100322

Brachman, R., & Levesque, H. (2004). *Knowledge Representation and Reasoning* . San Francisco: Morgan Kaufmann publications.

Bratt, S. (2007). *Semantic Web: Linked Data on the Web.* Retrieved from W3C: http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#%2824%29

Brickley, D., & Guha, R. (2004, February 10). *RDF Vocabulary Description Language 1.0: RDF Schema.* Retrieved March 18, 2009, from w3.org: http://www.w3.org/TR/rdf-schema/

Cera, C. D., Kim, T., Han, J., & Regli, W. C. (2004). Role-based viewing envelopes for information protection in collaborative modeling. *Computer-Aided Design 36* , 873–886.

Chen, T.-Y. (2008). Knowledge sharing in virtual enterprises via an ontology-based access control approach. *Computers in Industry 59* , 502–519.

Cheung, W. M., & Maropoulos, P. G. (n.d.). A Novel Knowledge Management Methodology To Support Collaborative Product Development. *Digital Enterprise Technology* .

Cho, J., Han, S., & Kim, H. (2006). Meta-ontology for automated information integration of parts libraries. *Computer-Aided Design 38* , 713–725.

Colvin, & Geoff. (2006, January 11). *The 100 Best Companies to Work For 2006.* Retrieved May 08, 2009, from CNNMoney Fortune: http://money.cnn.com/magazines/fortune/fortune_archive/2006/01/23/8366990/index.htm

Denis, L., Gardan, Y., & Perrin, E. (2003). A Framework for a Distributed CAD System. *Computer-Aided Design* , 761-773.

Fabien, G. (2001). Engineering an Ontology for a Multi-Agents Corporate. *Eigth International Symposium on the Management of Industrial and Corporate Knowledge.*

Fox, M. (1992). The TOVE Project: A Common-sense Model of the Enterprise. *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems* (pp. 25-34). Berlin: Springer-Verlag.

Fruchterman, T. M., & Reingold, E. M. (1991). Graph Drawing by Force-directed Placement. *SOFTWARE—PRACTICE AND EXPERIENCE* (pp. 1129-1164). John Wiley & Sons.

Fuh, J., & Li, W. (2005). Advances in collaborative CAD: the-state-of-the art. *Computer-Aided Design 37* , 571–581.

Gandon, F. L., & Sadeh, N. M. (2004). Semantic web technologies to reconcile privacy and context awareness . *Elsevier* .

Gössling, A., & Wollschlaeger, M. (2008). Integrating Information over the Life-cycle ofManufacturing Equipment by Assigning Semantics. *IEEE* .

Gruber, T. R., Vemuri, S., & Rice, J. (1997). Model-Based Virtual Document Generation. *nternational Journal of Human-Computer Studies* .

Gruninger, M., & Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing.* Montreal.

Haarslev, V., & Moller, R. (2003). Racer: A Core Inference Engine for the Semantic Web. *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003)*, (pp. 27-36). Sanibel Island.

Horridge, M., Bechhofer, S., & Noppens, O. (2007). Igniting the OWL 1.1 Touch Paper: The OWL API. *3rd OWL Experienced and Directions Workshop.* Innsbruck.

Horrocks, I., Patel-Schneider, P., F. Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004, May 21). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* Retrieved March 18, 2009, from w3.org: http://www.w3.org/Submission/SWRL/

IDC. (March 2008). *The Diverse and Exploding Digital Universe.* IDC.

Kim, K.-Y., Manley, D. G., & Yang, H. (2006). Ontology-based assembly design and information sharing for collaborative product development. *Computer-Aided Design , 38,* 1233–1250.

Kim, T., Cera, C. D., C., R. W., Choo, H., & Han, J. (2006). Multi-Level modeling and access control for data sharing in collaborative design. *Advanced Engineering Informatics* , 47-57.

Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004). The Protege OWL Plugin: An Open Development Environment for Semantic Web Applications. *International Semantic Web Conference.*

Koller, D., Turitzin, M., Levoy, M., Tarini, M., Croccia, G., Cignoni, P., et al. (2004). Protected Interactive 3D Graphics Via Remote Rendering. *Association for Computing Machinery* , 695-703.

Kwak, J.-A., & Yong, H.-S. (2008). An Approach to Ontology-based Semantic Integration for PLM Object. *IEEE International Workshop on Semantic Computing and Application* , 19-26.

Lin, H., & Harding, J. (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. *Computers in Industry , 58,* 428–437.

Manola, F., Miller, E., & McBride, B. M. (2004, February 10). *RDF Primer.* Retrieved March 18, 2009, from w3.org: http://www.w3.org/TR/rdf-primer/#intro

McGuinness, D. L., & Harmelen, F. v. (2004, February 10). *w3.org.* Retrieved March 18, 2009, from OWL Web Ontology Language Overview: http://www.w3.org/TR/owl-features/

Mikovec, Z., Cmolik, L., Kopsa, J., & Slavik, P. (2006). Beyond traditional interaction in a mobile environment: New approach to 3D scene rendering. *Computers & Graphics 30 , 30,* 714–726.

Mo, J., & Zhou, M. (2003). Tools and methods for managing intangible assets of virtual enterprise. *Computers in Industry 51* , 197-210.

Niles, I., & Pease, A. (2001). Towards a Standard Upper Ontology. *Formal Ontology in Information Systems FOIS.* ACM.

OntologyPortal. (n.d.). Retrieved July 25, 2009, from ontologyportal.org: http://www.ontologyportal.org/

Prud'hommeaux, E., & Seaborne, A. (2008, January 15). *SPARQL Query Language for RDF.* Retrieved March 18, 2009, from w3.org: http://www.w3.org/TR/rdf-sparql-query/

Qiu, Z. Q., Kok, K., Wong, Y., & Fuh, J. Y. (2007). Role-based 3D visualisation for asynchronous PLM collaboration. *Computers in Industry 58* , 747–755.

Qiu, Z., Wong, Y., Fuh, J., Chen, Y., Zhou, Z., Li, W., et al. (2003). Geometric model simplification for distributed CAD. *Computer-Aided Design* , 809-819.

Ru-zhi, X., Xiao-liang, D., Feng, Y., & Pei-guang, L. (2009). Research on the Construction Method of Emergency Plan Ontology Based-on OWL. *International Symposium on Web Information Systems and Applications* (pp. 019-023). Nanchang: ACADEMY PUBLISHER.

Sheth, J. C. (2006). *Semantic Web Services, Processes and Applications.* Springer.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). *Pellet: A Practical OWL-DL Reasoner.*

Soanes, C., & Stevenson, A. (2005). *The Oxford Dictionary of English (revised edition).* Oxford University Press.

Stark, J. (2004). *Product Lifecycle Management: 21st century Paradigm for Product Realisation.* London: Springer.

Staub, S., Fischer, M., Kunz, J., Ishii, K., & Paulson, B. (2003). A feature ontology to support construction cost estimating. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* , 133–154.

Tham, K. D., & Fox, M. S. (1994). A cost ontology for enterprise modelling. *Enabling Technologies: Infrastructure for Collaborative Enterprises.* Morgantown,WV: IEEE.

Unicode, Inc. (2008, Sep 17). *About the Unicode Standard.* Retrieved March 2009, from unicode.org: http://www.unicode.org/standard/standard.html

Vasconcelos, J., Kimble, C., & Gouveia, F. R. (2000). A design for a Group Memory System using Ontologies. *Proceedings of 5th UKAIS.* Cardiff: McGraw Hill.

W3C. (2009, 04 16). *Extensible Markup Language (XML).* Retrieved 07 26, 2009, from Ubiquitous Web Domain: http://www.w3.org/XML/

WC3. (2008, October). *Extensible Markup Language (XML).* Retrieved March 2009, from w3.org: http://www.w3.org/XML/

Zhang, G.-Q., Zhang, G.-Q., & Yang, Q.-F. (2008). Evolution of the Internet and its cores. *New Journal of Physics 10* .

Appendix A C++ Code for Creating a Depth Map using OpenGL

```
//Pixel coordinates
int i, j;
//Pointer to pixel data
unsigned char *pdata;
//Resize pixel data to fit the window size
pdata = (unsigned char*)malloc(gl_width*gl_height);
//Write the GL_DEPTH_COMPONENT to pdata
glReadPixels (0,0,gl_width,
gl_height,GL_DEPTH_COMPONENT,GL_UNSIGNED_BYTE,pdata);
//Invert the image and the depth color
for(j = 0; j*2 < gl_height; ++j )
{
      int index1 = j * gl_width;
      int index2 = (gl_height - 1 - j) * gl_width;
      for(i = gl_width; i > 0; --i )
      {
            unsigned char temp = pdata[index1];

            pdata[index1] = abs(pdata[index2]-255);
            pdata[index2] = abs(temp-255);
            ++index1;
            ++index2;
      }
}
//Save the pixel data, pdata, in a BMP file
SOIL_save_image(depthFile,SOIL_SAVE_TYPE_BMP,gl_width,
gl_height,1,pdata);
```

Appendix B GLSL Fragment Program to Render the Shared Mask

```
//Shared Mask Fragment Shader

//This program is used to create shared mask images
//The depth values of two depth map is compared for each pixel

//Variable to access the depth values of two depth maps
uniform sampler2D D1;
uniform sampler2D D2;

//Coordinate of a pixel
varying vec2 texCoord;

void main(void)
{

	//Final color
	vec3 color;
	//Depth value of depth map 1
	vec3 depth1 = vec3(texture2D(D1, texCoord));
	//Depth value of depth map 2
	vec3 depth2 = vec3(texture2D(D2, texCoord));

	//Compare the depth values
	//Set the color to white if the depth value of image1 is lower
	if (depth1.x<depth2.x){
		color = vec3(1.0,1.0,1.0);

	}
	//Set the color to black otherwise
	else {
		color = vec3(0.0,0.0,0.0);

	}

	//Set the color of the fragment
	gl_FragColor = vec4(color,1.0);


}
```

Appendix C GLSL Fragment Program to Render the Shared Image

```
//Shared Mask Fragment Shader

//This program is used to create shared images.
//The pixels that are outside of the shared mask are discarded.


//Variable to access an image
uniform sampler2D P1;
//Variable to access a shared mask
uniform sampler2D M1;

//Coordinate of a pixel
varying vec2 texCoord;

void main(void)
{

      //Shared mask value
      vec3 mask = vec3(texture2D(M1, texCoord));

      //Check the shared mask value
      //Discard the pixel,if the color of the shared mask is equal to 0
      if ((mask.x + mask.y + mask.z)==0.0)
            discard;
      //Set the color of the fragment equal to the original color
      else
            gl_FragColor = vec4(texture2D(P1, texCoord));


}
```

Appendix D GLSL Fragment Program for the Complete Image

```
//Complete Image Fragment Shader

//This program is used to create the complete image
//The pixels from two shared image are added together


//Variable to access two shared images
uniform sampler2D P1;
uniform sampler2D P2;

//Coordinate of the pixel
varying vec2 texCoord;

void main(void)
{

	//Final color
	vec3 color;

	//Shared image 1 pixel color
	vec3 image1 = vec3(texture2D(P1, texCoord));

	//Check the shared image pixel color
	//If the color is black set the pixel color to image 2
	//Else, use the pixel color of image 1
	if ((mask.x + mask.y + mask.z)==0.0)
		vec3 color = vec3(texture2D(P1, texCoord));
	else
		vec3 color = image1;

	gl_FragColor = vec4(color);


}
```