1. **INTRODUCTION TO SEMANTIC WEB**

The today's World Wide Web's content is designed for humans to read and understand, not for machines and computer programs to manipulate meaningfully. Computers can adeptly parse Web pages for layout and routine processing but, in general, machines have no reliable way to process the semantics. The Semantic Web will bring structure to the meaningful content of Web pages, where software agents roaming from page to page or from site to site can readily carry out automated sophisticated tasks for users.

 The World-wide web, a system of interlinked, hypertext documents accessed via the Internet, has transformed many areas of human endeavor. For example, scientific discovery is increasingly driven by our ability to share, integrate, and analyze data over the web. However, the current web falls significantly short of realizing its full potential as envisioned by its inventor Tim Berners- Lee. This is due to the fact that most of the information that is currently available on the web is designed for human consumption. The semantic web is aimed at transforming the web into an information space designed to support not only human-human communication, but also for human-machine and machine-machine communication. Semantic web is a key enabler of large scale distributed, integrative, collaborative e-science. Now, We may define the Semantic Web as according to Tim Berners-Lee, the inventor of World Wide Web is,

 "The extension of the current web in which information is given well-defined meaning, better enabling computers and humans to work in cooperation."

* 1. **WEB TO THE SEMANTIC WEB**

The World Wide Web has changed the way people communicate with each other and the way business is conducted. It lies at the heart of a revolution that is currently transforming the developed world toward a knowledge economy and, more broadly speaking, to a knowledge society. This development has also changed the way we think of computers. Originally they were used for computing numerical calculations. Currently their predominant use is for information processing, typical applications being data bases, text processing, and games. At present there is a transition of focus towards the view of computers as entry points to the information highways.

Most of today’s Web content is suitable for human consumption. Even Web content that is generated automatically from databases is usually presented without the original structural information found in databases. Typical uses of the Web today involve people’s seeking and making use of information, searching for and getting in touch with other people, reviewing catalogs of online stores and ordering products by filling out forms, and viewing adult material.

* 1. **SEMANTIC WEB SOLUTIONS**

The Semantic Web takes the solution further. It involves publishing in languages specifically designed for data: Resource Description Framework (RDF), Web Ontology Language (OWL), and Extensible Markup Language (XML). HTML describes documents and the links between them. RDF, OWL, and XML, by contrast, can describe arbitrary things such as people, meetings, or airplane parts. Tim Berners-Lee calls the resulting network of Linked Data the Giant Global Graph, in contrast to the HTML-based WorldWideWeb.

These technologies are combined in order to provide descriptions that supplement or replace the content of Web documents. Thus, content may manifest as descriptive data stored in Web-accessible databases, or as markup within documents (particularly, in Extensible HTML (XHTML) interspersed with XML, or, more often, purely in XML, with layout/rendering cues stored separately). The machine-readable descriptions enable content managers to add meaning to the content, i.e. to describe the structure of the knowledge we have about that content. In this way, a machine can process knowledge itself, instead of text, using processes similar to human deductive reasoning and inference, thereby obtaining more meaningful results and facilitating automated information gathering and research by computers.

An example of a tag that would be used in a non-semantic web page:

<item>cat</item>

Encoding similar information in a semantic web page might look like this:

<item rdf:about="http://dbpedia.org/resource/Cat">Cat</item>

1. **A LAYERED APPROACH TO THE SEMANTIC WEB**

The development of the Semantic Web proceeds in steps, each step building a layer on top of another. In building one layer of the Semantic Web on top of another, two principles should be followed:

* **Downward compatibility:** Agents fully aware of a layer should also be able to interpret and use information written at lower levels. For example, agents aware of the semantics of OWL can take full advantage of information written in RDF and RDF Schema.
* **Upward partial understanding:** On the other hand, agents fully aware of a layer should take at least partial advantage of information at higher levels. For example, an agent aware only of the RDF and RDF Schema semantics an interpret knowledge written in OWL partly, by disregarding those elements that go beyond RDF and RDF Schema.



Fig 2.1 : Layered Approach of Semantic Web

**2.1 THE TECHNOLOGIES**

The common use of the term Semantic Web is to identify a set of technologies, tools and standards which form the basic building blocks of a system that could support the vision of a Web imbued with meaning. The Semantic Web has been developing a layered architecture, which is often represented using a fig 2.1 first proposed by Tim Berners-Lee, with many variations since.

While necessarily a simplification which has to be used with some caution, it nevertheless gives reasonable conceptualizations of the various components of the Semantic Web. We describe briefly these layers.

* **Unicode and URI(Uniform Resource Locater) :** Unicode, the standard for computer character representation, and URIs, the standard for identifying and locating resources (such as pages on the Web), provide a baseline for representing

characters used in most of the languages in the world, and for identifying resources.

 <class-def>

 <class name=”plant”>

 <subclass-of>

 <NOT><class name=”animal”/></NOT>

 </subclass-of>

 </class-def>

 <class-def>

 <class name=”tree”/>

 <subclass-of>

 <class name=”plant”>

 </subclass-of>

 </class-def>

 <class-def>

 <class name=”branch”/>

 <slot-constraint>

<slot name=”is-part-of”/>

 <has-value>

 <class name=”tree”/>

 </has-value>

\ </slot-constraint>

 </class-def>

Fig 2.2. : Snap shot for XML code

* **XML (Extensible Markup Language):** A language that lets one write structured Web documents with a user-defined vocabulary fig 2.2. XML is particularly suitable for sending documents across the Web.
* **RDF (Resource Description Framework)** is a basic data model, like the entity-relationship model, for writing simple statements about Web objects (resources) fig 2.3 The RDF data model does not rely on XML, but RDF has an XML-based syntax. Therefore, in figure it is located on top of the XML layer.
* **RDF Schema** provides modeling primitives for organizing Web objects into hierarchies. Key primitives are classes and properties, subclass and sub property relationships, and domain and range restrictions. RDF Schema is based on RDF. RDF Schema can be viewed as a primitive language for writing ontology’s. But there is a need for more powerful ontology languages that expand RDF Schema and allow the representations of more complex relationships between Web objects.

hasName(‘http://www://www.w3.org/employee/id132’,”Jim Berners”).

authorOf(‘http://www.w3.org/employee/id132’,’http://www.books.org/ISBN0012515866’).

hasPrice(‘http://www.books.org/ISBN0625515861, “$62”).



Fig 2.3: RDF Example

* **Ontology Vocabulary** explicit formal specifications of the terms in the domain and relations among them has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts.
* **Logic layer** is used to enhance the ontology language further and to allow the writing of application-specific declarative knowledge.
* **Proof layer** involves the actual deductive process as well as the representation of proofs in Web languages (from lower levels) and proof validation.
* **Trust layer** will emerge through the use of digital signatures and other kinds of knowledge, based on recommendations by trusted agents or on rating and certification agencies and consumer bodies. Sometimes “Web of Trust” is used to indicate that trust will be organized in the same distributed and chaotic way as the WWW itself. Being located at the top of the pyramid, trust is a high-level and crucial concept, The Web will only achieve its full potential when users have trust in its operations (security) and in the quality of information provided.
1. **ONTOLOGY**

**3.1 DEFINITION**

Ontology is defined as “explicit specification of conceptualization” or it can be a formal conceptualization of a domain that is shared and reused across domains, tasks and group of people. Ontology is a model of the world, represented as a tangled tree of linked concepts. Ontology is used to capture knowledge about some domain of interest. Ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C).Basic structure of Ontology fig. 3.1 is formed by following components,



Fig 3.1: Ontology Structure

1. **• Classes :** OWL classes are interpreted as sets that contain individuals. They are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class fig 3.2. For example, the class Cat would contain all the individuals that are cats in our domain of interest. Classes may be organized into a super class-subclass hierarchy, which is also known as taxonomy. Subclasses specialize (‘are
2. subsumed by’) their super classes. For example consider the classes Animal and Cat – Cat might be a subclass of Animal (so Animal is the super class of Cat). This says that, ‘All cats are animals’, ‘All members of the class Cat are members of the class Animal’, ‘Being a Cat implies that you’re an Animal’, and ‘Cat is subsumed by Animal’.



Fig 3.2 : Representation of Classes (Containing Individuals)

1. **• Instances/ Individuals:** Individuals, represent objects in the domain that we are interested in. An important difference between Prot´eg´e and OWL is that OWL does not use the Unique Name Assumption (UNA). This means that two different names could actually refer to the same individual. For example, “Queen Elizabeth”, “The Queen” and “Elizabeth Windsor” might all refer to the same individual. In OWL, it must be explicitly stated that individuals are the same as each other, or different to each other — otherwise they might be the same as each other, or they might be different to each other. Fig. 3.3 shows a representation of some individuals in some domain.
2. 

Fig 3.3: Representation of Individuals

1. • **Relationships/Properties:** Properties are binary relations on individuals - i.e. properties link two individuals together. For example, the property hasSibling might link the individual Matthew to the individual Gemma, or the property hasChild might link the individual Peter to the individual Matthew. Properties can have inverses. For example, the inverse of hasOwner is isOwnedBy. Properties can be limited to having a single value –i.e. to being functional. They can also be

either transitive or symmetric. Fig. 3.4 shows a representation of some properties linking some individuals together.



Fig 3.4 : Representation of Properties

1. • **Forms** are framework that is used to set the layout for the instances in ontology.
2. • **Constraints** are conditions that must be satisfied during the design. A property restriction is a special kind of class description. It defines an anonymous class, namely the set of class of all individuals that satisfy the restriction.In OWL properties are used to create restrictions. As the name may suggest, restrictions are used to restrict the individuals that belong to a class. Restrictions in OWL fall into three main categories:
3. a. Quantifier Restrictions
4. b. Cardinality Restrictions
5. c. hasValue Restrictions.

We will initially use quantifier restrictions. These types of restrictions are composed of a quantifier, a property, and filler. The two quantifiers that may be used are:

• The existential quantifier (), which can be read as at least one, or some.

• The universal quantifier (), which can be read as only.

Using these attribute we have designed the ontology for Geology domain that can be used for the searching purpose and work as knowledge base for semantic web. It includes a collection of domain-specific concepts, and is a system description which includes class-

subclass taxonomy, slots, forms, instances, relationships, constraints and performing query in knowledge base. The ontology design process is evolutionary in nature.

Ontology’s are classified in four groups, according to their dependency on a specific domain or point of view,

1. i) Top-level ontology’s describe very general concepts,
2. ii) Upper level ontology’s describe the vocabulary related to a generic domain.
3. iii) Domain ontology’s describe a domain or task.
4. iv) Application ontology’s are at the lowest level in inheritance view combines, integrates, and extends all sub ontology’s for the application.
5. **4. DESIGN PROCESS OF ONTOLOGY**

The ontology has been designed by the process depicted here. The various steps of process are shown in fig. 4.1.

1. • **Expert Analysis/ Domain Analysis:** First step in ontology design process is to analysis the domain for which we are going to design ontology. For analysis we need an expert of the particular domain having the knowledge about the knowledge representation for that domain. The expert will cover the following main issues regarding ontology: Ontology scope and Knowledge source. In our study scope of our geo ontology is to classify a satellite image with maximum accuracy.
2. • **Tool and Languages/ Design Structure:** The ontology development tools such as Protégé, SWOOP and many others are freely available. Protégé is one of the best choices for a free software ontology development platform. Several ontology languages are available like RDF, RDFS, DAML+OIL, OWL. OWL has three versions OWL lite, OWL DL, OWL Full. Each language have their own characteristics. We have made use of RDF/XML language for geo-ontology construction.



Fig. 4.1: Ontology Design Process

1. • **Ontology Design/ Creation:** Ontology design covers the design of framework for ontology by user, expert and designer to represent knowledge in efficient way.
2. • **Validation and Maintenance:** Ontology is checked for validation. It is checked to determine if it is balanced or not.
3. **5. ONTOLOGY LANGUAGES**

Ontology can be expressed in various formats. Each format has its limitations.The various representation formats are RDF/XML, Notation 3, TURTLE, OWL.

**5.1 OWL ONTOLOGY**

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies, and is endorsed by the World Wide Web Consortium. This family of languages is based on two (largely, but not entirely, compatible) semantics: OWL DL and OWL Lite semantics are based on Description Logics, which have attractive and well-understood computational properties, while OWL Full uses a novel semantic model intended to provide compatibility with RDF Schema. OWL ontologies are most commonly serialized using RDF/XML syntax. OWL is

considered one of the fundamental technologies underpinning the Semantic Web, and has attracted both academic and commercial interest.

Like Protégé OWL makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators - e.g. and, or and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Complex concepts can therefore be built up in definitions out of simpler concepts. Furthermore, the logical model allows the use of a reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent and can also recognize which concepts fit under which definitions. The reasoner can therefore help to maintain the hierarchy correctly. This is particularly useful when dealing with cases where classes can have more than one parent.

* + 1. **THE SPECIES OF OWL**

OWL ontology may be categorized into three species or sub-languages: OWL-Lite, OWL-DL and OWL- Full. A defining feature of each sub-language is its expressiveness. OWL-Lite is the least expressive sub-language. OWL-Full is the most expressive sub-language. The expressiveness of OWL-DL falls between that of OWL-Lite and OWL-Full. OWL-DL may be considered as an extension of OWL-Lite and OWL-Full an extension of OWL-DL.

* **OWL Lite** was originally intended to support those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It was hoped that it would be simpler to provide tool support for OWL Lite than its more expressive relatives, allowing quick migration path for systems utilizing thesauri and other taxonomies.
* **OWL DL OWL-DL** is much more expressive than OWL-Lite and is based on Description Logics (hence the suffix DL). Description Logics are a decidable fragment of First Order Logic2 and are therefore amenable to automated reasoning. It is therefore possible to automatically compute the classification hierarchy and check for inconsistencies in an ontology that conforms to OWL-DL.OWL DL was designed to provide the maximum expressiveness possible while retaining computational completeness (all conclusions are guaranteed to be computed), decidability (all computations will finish in finite time), and the availability of practical reasoning algorithms.
* **OWL Full** is based on a different semantics from OWL Lite or OWL DL, and was designed to preserve some compatibility with RDF Schema. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right; this is not permitted in OWL DL.

Each of these sublanguages is a syntactic extension of its simpler predecessor. The following set of relations hold. Their inverses do not.

1. • Every legal OWL Lite ontology is a legal OWL DL ontology.
2. • Every legal OWL DL ontology is a legal OWL Full ontology.
3. • Every valid OWL Lite conclusion is a valid OWL DL conclusion.
4. • Every valid OWL DL conclusion is a valid OWL Full conclusion.

**5.2 RDF/XML**

The **Resource Description Framework (RDF)** is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources; using a variety of syntax formats.

Basically speaking, the RDF data model is not different from classic conceptual modeling approaches such as Entity-Relationship or Class diagrams, as it is based upon the idea of making statements about resources, in particular, Web resources, in the form of subject-predicate-object expressions. These expressions are known as triples in RDF terminology fig 5.2. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object.

For example, one way to represent the notion "The sky has the color blue" in RDF is as the triple: a subject denoting "the sky", a predicate denoting "has the color", and an object denoting "blue". RDF is an abstract model with several serialization formats (i.e., file formats), and so the particular way in which a resource or triple is encoded varies from format to format.

This mechanism for describing resources is a major component in what is proposed by the W3C's Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity.

A collection of RDF statements intrinsically represents a labeled, directed multi-graph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model and other ontological models traditionally used in computing today. However, in practice, RDF data is often persisted in relational database or native representations also called Triple stores, or Quad stores if context (i.e. the named graph) is also persisted for

each RDF triple.As RDFS and OWL demonstrate, additional ontology languages can be built upon RDF.

**5.2.1 RDF BASIC IDEAS**

The fundamental concepts of RDF are resources, properties and statements.

* **Resources**

We can think of a resource as an object, a “thing” we want to talk about. Resources may be authors, books, publishers, places, people, hotels, rooms, search queries, and so on. Every resource has a URI, a Universal Resource Identifier. A URI can be a URL (Unified Resource Locator, or Web address) or some other kind of unique identifier; note that an identifier does not necessarily enable access to a resource. URI schemes have been defined not only for web-locations but also for such diverse objects as telephone numbers, ISBN numbers and geographic locations. There has been a long discussion about the

nature of URIs, even touching philosophical questions (for example, what is an appropriate unique identifier for a person?), but we will not go into into detail here. In general, we assume that a URI is the identifier of a Web resource.

* **Properties**

Properties are a special kind of resources; they describe relations between resources, for example “written by”, “age”, “title”, and so on. Properties in RDF are also identified by URIs (and in practice by URLs). This idea of using URIs to identify “things” and the relations between is quite important. This choice gives us in one stroke a global, worldwide, unique naming scheme. The use of such a scheme greatly reduces the homonym problem that has plagued distributed data representation until now.

* **Statements**

Statements assert the properties of resources. A statement is an object attribute-value triple, consisting of a resource, a property, and a value. Values can either be resources or literals.

Literals are atomic values (strings), the structure of which we do not discuss further. An example of a statement is

David Billington is the owner of the Web page http://www.cit.gu.edu.au/∼db.

The simplest way of interpreting this statement is to use the definition and consider the triple

( “David Billington”, http://www.mydomain.org/site-owner, http://www.cit.gu.edu.au/∼db).

We can think of this triple (x, P, y) as a logical formula P(x, y), where the binary predicate P relates the object x to the object y. In fact, RDF offers only binary predicates (properties) fig 5.2 Note that the property “site-owner” and one of the two objects are identified by URLs, whereas the other object is simply identified by a string.



Fig 5.2 : Graph Representation of Triple

A second view is graph-based. fig. 5.3 shows the graph corresponding to the preceding statement. It is a directed graph with labeled nodes and arcs; the arcs are directed from the resource (the subject of the statement) to the value (the object of the statement). This kind of graph is known in the Artificial Intelligence community as a semantic net .

As we already said, the value of a statement may be a resource. Therefore, it may be linked to other resources. Consider the following triples:

( http://www.cit.gu.edu.au/∼db, http://www.mydomain.org/siteowner,“David Billington”)

( “David Billington”, http://www.mydomain.org/phone, “3875507”)

(“DavidBillington”,http://www.mydomain.org/uses,http://www.cit.gu.edu.au/∼arock/defeasible/Defeasible.cgi)

(“www.cit.gu.edu.au/∼arock/defeasible/Defeasible.cgi”,http://www.mydomain.org/site-owner, “Andrew Rock”)

 

Fig 5.3: Semantic Net

Graphs are a powerful tool for human understanding. But the Semantic Web vision requires machine-accessible and machine-process able representations.

* + 1. **RDF/XML SYNTAX**

An RDF document consists of an rdf:RDF element, the content of which is a number of descriptions. For example, consider the domain of university courses and lecturers at VTU in the year 2001.

<rdf:RDF

xmlns:rdf=”[http://www.w3.org/1999/02/22-rdf-syntax-ns#](http://www.w3.org/1999/02/22-rdf-syntax-ns)”

xmlns:rdf=”[http://www.w3.org/2001/XLMSchema#](http://www.w3.org/2001/XLMSchema)”

xmlns:uni=”[http://www.mydomain.org/uni-ns#](http://www.mydomain.org/uni-ns)”>

<rdf:Description rdf:about=”949352”>

<uni:name>Grigoris Antoniou</uni:name>

<uni:title> Professor</uni:name>

</rdf:Description>

<rdf:Description rdf:about=”949318”>

<uni:name>David Billington</uni:name>

<uni:title> Associative Professor</uni:name>

<uni:age rdf:datatype=”&xsd:integer”>27</uni:age>

</rdf:Description>

<rdf:Description rdf:about=”CIT1111”>

<uni:courseNameDiscrete maths </uni:courseName>

<uni:isTaughtBye> David Billington </uni:isTaughtBye>

</rdf:Description>

<rdf:Description rdf:about=”CIT1112”>

<uni:courseName>Concrete maths </uni:courseName>

<uni:isTaughtBye> Grigoris Antoniou</uni:isTaughtBye>

</rdf:Description>

</rdf:RDF>

* **First,** the namespace mechanism of XML is used, but in an expanded way. In XML namespaces are only used for disambiguation purposes. In RDF external namespaces are expected to be RDF documents defining resources, which are then used in the importing RDF document. This mechanism allows the reuse of resources by other people who may decide to insert additional features into these resources. The result is the emergence of large, distributed collections of knowledge.
* **Second,** the rdf:about attribute of the element rdf:Description is strictly speaking equivalent meaning to that of an ID attribute, but it is often used to suggest that the object about which a statement is made has already been “defined” elsewhere. Formally speaking, a set of RDF statements together simply forms a large graph, relating things to other things through properties, and there is no such thing as “defining” an object in one place and referring to it elsewhere. Nevertheless, in the serialized XML syntax, it is sometimes useful (if only for human readability) to suggest that one location in the XML serialization is the “defining” location, while other locations state “additional” properties about an object that has been “defined” elsewhere. In fact the preceding example is slightly misleading. If we wanted to be

absolutely correct, we should replace all occurrences of course and staff ID’s, such as 949352 and CIT3112, by references to the external namespace, for example

<rdf:Description>

rdf:about=”http://www.mydomain.org/uni-ns/#CIT3112”

We have refrained from doing so to improve readability of our initial example because we are primarily interested here in the ideas of RDF. However, readers should be aware that this would be the precise way of writing a correct RDF document. The content of rdf:Description elements are called property elements. For example, in the description

<rdf:Description rdf:about=”CIT3116”>

<uni:courseName>Knowledge Representation</uni:courseName>

<uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>

</rdf:Description>

 The two elements uni:courseName and uni:isTaughtBy both define property-value pairs for CIT3116. The preceding description corresponds to two RDF statements.

* **Third,** the attribute rdf:datatype="&xsd;integer" is used to indicate the data type of the value of the age property. Even though the age property has been defined to have "&xsd;integer" as its range, it is still required to indicate the type of the value of this property each time it is used.

**5.2.3 THE rdf: RESOURCE ATTRIBUTE**

The preceding example was not satisfactory in one respect: the relationships between courses and lecturers were not formally defined but existed implicitly through the use of the same name. To a machine, the use of the same name may just be a coincidence: for example, the David Billington who teaches CIT3112 may not be the same person as the person with ID 949318 who happens to be called David Billington. What we need instead is a formal specification of the fact that, for example, the teacher of CIT1111 is the staff member with number 949318, whose name is David Billington. We can achieve this effect using an rdf:resource attribute:

<rdf:Description rdf:about=”CIT1111”>

<uni:courseName> Discrete Maths</uni:courseName>

<uni:isTaughtBy rdf:resource=”949318”>

</rdf:Description>

<rdf:Description rdf:about=”949318”/>

<uni:name> David Billington</uni:name>

<uni:title> Associate Professor </uni:title>

</rdf:Description>

We note that in case we had defined the resource of the staff member with ID number 939318 in the RDF document using the ID attribute instead of the about attribute, we would have had to use a # symbol in front of 949318 in the value of rdf:resource:

<rdf:Description rdf:about=”CIT1111”>

<uni:courseName>Discrete Maths</uni:courseName>

<uni:isTaughtBy rdf:resource=”#949318”/>

</rdf:Description>

<rdf:Description rdf:ID=”#949318”>

<uni:name> David Billington </uni:name>

<uni:title> Associate Professor</uni:title>

</rdf:Description>

The same is true for externally defined resources: For example, we refer to the externally defined resource CIT1111 by using http://www.mydomain.org/uni-ns/#CIT1111 as the value of rdf:about, where www.mydomain.org/uni-ns/ is the URI where the definition of CIT1111 is found. In other words, a description with an ID defines a fragment URI, which can be used to reference the defined description.

**6. ADVANTAGES OF SEMANTIC WEB**

The advantage of the Semantic Web allows much more advanced knowledge management systems:

* Knowledge will be organized in conceptual spaces according to its meaning.
* Automated tools will support maintenance by checking for inconsistencies and extracting new knowledge.
* Keyword-based search will be replaced by query answering: requested knowledge will be retrieved, extracted, and presented in a human friendly way.
* Query answering over several documents will be supported.
* Defining who may view certain parts of information (even parts of documents) will be possible.

**7. APPLICATIONS OF SEMANTIC WEB**

We have discussed four areas where the Semantic Web is most likely to make an impact: information management, digital libraries, virtual communities, and e-learning. To summarise:

* **Information Management:** The Semantic Web enhances the capabilities of those tools which form a familiar part of the current Web so that they can become useful information management tools in their own right. The Web is already an information source of choice for many learners and researchers. A more structured and directed approach to managing this information space, both within institutions and across the whole community, can make this information more useful, with less wasted effort, and more capacity to measure the quality of information. By making the annotation machine readable, it becomes accessible to automatic processing, carrying out many routine tasks which consume people’s time. A further impact is likely to be in the business of running education, allowing more efficient information flow around institutions.
* **Digital Libraries:** The impact on digital libraries, combined with the Open Access Initiative and the rise of open archiving is likely to be quite profound. Libraries become 'value-added' information annotators and collators rather than the archivists of externally published literature and the holders of the published output of institutions. The Semantic Web, although not a prerequisite or a motivator for this change is nevertheless likely to smooth its development. The tools are in place for sharing classification schemes and to allow the community to develop, deepen and share such schemes. The information infrastructure tools discussed above will have particular impact on the way students and researchers find information, so these tools may typically be provided and adapted by libraries who will tailor them to the needs of their own users. The Semantic Web, like the current Web, has the capacity of being an overwhelming place; libraries are well-placed to make sense of this for the HE and FE community.
* **Building communities and collaborations:** A major impact is likely to occur in the way that academic communities work together. The tools for forming virtual communities and sharing information across that community are simple and lightweight, and, if the

development of blogs and the use of RSS is an indication, can enhance the interaction of an interested community by an enormous amount. Providing a richer annotation structure to these can only enhance their usefulness, bringing them into the information infrastructure as well as providing a means of communication to people across the world. Support for virtual collaborations is a much larger issue, as it requires tighter control over resources and security. This is largely taking place in the Grid community and efforts to construct a Semantic Grid are already well underway, bringing the machine readable annotation to automate the discovery and negotiation of services onto the Grid.

* **E-Learning:** All of the above can influence e-learning. However, we should also consider specifically, support for the presentation and delivery of course materials and for assisting and assessing students. Again, the impact of the Semantic Web is likely to mean that these can be more closely tailored to the needs of the user, with a choice of learning objects mediated through selection mechanisms. The Semantic Web can provide context and co-ordination, with workflow tools providing a supporting infrastructure.

**8. CONCLUSION**

The goal of the Semantic Web initiative is as broad as that of the Web to create a universal medium for the exchange of data. It is envisaged to smoothly interconnect personal information management, enterprise application integration, and the global sharing of commercial, scientific and cultural data. Facilities to put machine-understandable data on the Web are quickly becoming a high priority for many organizations, individuals and communities.

The Web can reach its full potential only if it becomes a place where data can be shared and processed by automated tools as well as by people. For the Web to scale, tomorrow's programs must be able to share and process data even when these programs have been designed totally independently. The Semantic Web Activity is an initiative of the World Wide Web Consortium (W3C) designed to provide a leadership role in defining this Web. The Activity develops open specifications for those technologies that are ready for large scale deployment, and identifies, through open source advanced development, the infrastructure components that will be necessary to scale in the Web in the future.

The principal technologies of the Semantic Web fit into a set of layered specifications. The current components are the Resource Description Framework (RDF) Core Model, the RDF Schema language, the Web Ontology language (OWL), and the Simple Knowledge Organization System (SKOS). Building on these core components is a standardized query language, SPARQL (pronounced "sparkle"), enabling querying decentralized collections of RDF data. The POWDER recommendations provide technologies to find resource descriptions for specific resources on the Web; descriptions which can be ‘joined’ to other RDF data. The GRDDL and RDF a Recommendations aim at creating bridges between the RDF model and various XML formats, like XHTML. Finally, the goal of the R2RML language (under development) is to provide standard language to map relational data and relational database schemas to RDF and OWL.

**REFERENCES**

 1. Annotea project homepage: http://www.w3.org/2001/Annotea/ [last accessed 25/04/05]

1. *G. Antoniou, F. van Harmelen* (2004). A Semantic Web Primer. MIT Press.

2 *G. Beged-Dov et al*. (2000). RDF Site Summary (RSS) 1.0. Available

 from: http://purl.org/rss/1.0/spec [last accessed 25/04/05]

1. *Dave Beckett* (2004). RDF/XML Syntax Specification (Revised). W3C

Recommendatio 10th February 2004.

1. *Dave Beckett's* Resource Description Framework (RDF) Resource Guide: available at: http://www.ilrt.bris.ac.uk/discovery/rdf/resources/ [last accessed 25/04/05]
2. *T. Berners-Lee* (1989). Information Management: A Proposal. CERN. Available at: http://www.w3.org/History/1989/proposal.html [last accessed 25/04/05]
3. *T. Berners-Lee* (1998). Semantic Web Road Map. W3C. Available at: http://www.w3.org/DesignIssues/Semantic.html [last accessed 25/04/05]
4. *T. Berners-Lee, J. Hendler, and O. Lassila* (2001). The Semantic Web. Scientific American. Available at: http://www.sciam.com/print\_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21 [last accessed 25/04/05]
5. Business Process Execution Language for Web Services (BPEL4WS) homepage: http://www-128.ibm.com/developerworks/library/ws-bpel/ [last accessed 25/04/05]
6. *J. Bradshaw et al.* (2003). Representation and reasoning about DAML-based policy and domain services in KAoS. In: J. Rosenschein, M. Wooldridge, Proc. of the 2nd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems. ACM Press, pp. 835–842.
7. *Brickley and R.V. Guha* (2000). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Candidate recommendation, 27th March 2000. Available at: http://www.w3.org/TR/2000/CR-rdf-schema-20000327/ [last accessed 25/04/05]