Research Report:

Information Retrieval on the Semantic Web Using Ontology-based Visualization

Larry Reeve
for Dr. Han
INFO780, Spring 2004
Term Project

Table of Contents

Ontology and Semantic Web Overview

The Semantic Web extends the current World Wide Web by adding facilities for machine-processable descriptions of meaning. In order for semantic exchanges of information to take place, there needs to be agreement on how to model meaning. Ontologies are the mechanism for representing formal and shared domain descriptions (Geroimenko and Chen, 2002). Ontologies help both people and machines communicate more effectively by providing a common definition of a domain (van Harmelen et al., 2001). Ontologies can be generally defined as a “specification of a conceptualization” (Gruber, 1993). Ontologies are metadata that provide a controlled vocabulary of terms, where each term is defined explicitly so that it is machine-processable. Ontologies facilitate machine processing by allowing information to be annotated with metadata so that meaning can be determined. Ontologies are expected to play a central role in the development of the Semantic Web, and will be used for many different purposes, such as querying, presentation, and navigation (Fluit et al., 2003). Ontologies are similar to taxonomies in that classification is provided. However, ontologies also include information about the relationships among terms (concepts), which provides the basis for semantic reasoning (Seeling and Becks, 2003).

There is a substantial amount of work currently being devoted to issues surrounding the Semantic Web. In the United States, DARPA has provided $70 million in funding to develop the Semantic Web. In Europe, the European Union has focused part of its IST programme by allocating 55 million Euros to semantic-based knowledge systems (van Harmelen et al., 2001). The IST programme web site (http://www.cordis.lu/ist/so/knowledge/home.html) states its goal for semantic systems as “*To develop semantic-based and context-aware systems to acquire, organise, process, share and use the knowledge embedded in multimedia content. Research will aim to maximise automation of the complete knowledge lifecycle and achieve semantic interoperability between Web resources and services*”. Clearly there is worldwide support and work going on to develop and improve semantic-based systems. Implicit in both the European and U.S. systems is that ontologies will be a central part of the development of semantic-based systems (Fluit et al., 2003).

Ontologies on the Semantic Web can developed in many forms. They can be lightweight, which normally means they are simple keyword-hierarchies that consist of a set of classes representing concepts and as well as an expression of usually minimal relationships (Geroimenko and Chen, 2002). This type of ontology is also known as a taxonomy. For example, the Yahoo! Web site categorizes web sites based on an index of topics that is hierarchically organized. The web pages of a web site can be annotated using the Yahoo! categories. In this case, the Yahoo! category index serves as a lightweight ontology. Ontologies can also be more complex. They can consist of complex concept-hierarchies with properties, value-restrictions, and axiomatised relations between concepts (van Harmelen et al., 2001). However, it is expected that the ontologies on the Semantic Web will tend to be oriented more towards lightweight ontologies (Fluit et al., 2003). An obvious attraction of lightweight ontologies is their simplicity. Visualization techniques need to be developed and adapted that take advantage of the properties of lightweight ontologies. Since the sub-classes of a lightweight ontology tend to overlap, visualization can exploit this fact to show query results that may not have an exact match, but show the user what other constraints have been met. Such a feature allows the user to perform query relaxation during retrieval in order to better satisfy an information need.

There are currently many ontologies that have been developed and are stored in various public repositories. For example, the DARPA Agent Markup Language (DAML) web site (www.daml.org) currently lists 282 ontologies in its DAML Ontology Library. The diversity of ontologies stored at the DAML web site include an ontology for describing baseball teams, players, games and plays, an employment hierarchy for Carnegie Mellon University, and an ontology to capture the most popular coordinate systems used by GPS. Eventually it is expected that the specification of ontologies in domains will reach the point where the depth is great enough to support annotation of vast majority of the resources available on the Web (Garcia and Sicilia, 2003). In some domains, the specification may not be complete but is complete enough to be useful (lightweight), while in other domains the ontologies developed will need to be more fully specified in order to be considered useful. The important point is that ultimately the Semantic Web will allow information retrieval to transition from unstructured keyword-based models to richer logic-based annotations that provide a basis for reasoning (Garcia and Sicilia, 2003).

RDF and OWL W3C standards for the Semantic Web

In order to support ontologies on the Semantic Web, standards need to be developed to support interoperability between machines. There have been several de facto standards to date, but recent work by the W3C has formalized standards. The two key standards for Semantic Web technologies that have been defined by the W3C are RDF and OWL. Both of these technologies received final approval in February 2004. The Resource Description Framework (RDF) recommendation documents a language for representing information about resources on the Web (W3C RDF, 2004). The Web Ontology Language (OWL) recommendation is designed to let applications process the content of information (W3C OWL, 2004).

RDF is a data model for representing resources and their relations between them, that is, metadata about resources (W3C OWL, 2004). The data model is represented using XML. An RDF statement is a triplet composed of a subject, a predicate, and an object. For example, in the statement “index.html has a creator whose value is John Smith”, the subject is the URL for index.html, the predicate is “creator”, and the object is “John Smith”. Each RDF statement is modeled as a graph structure, where subjects and objects are nodes and the predicate is an arc. The example above can be represented in graph form as node(“index.html”) → predicate(“creator”) → node(“John Smith”). RDF allows the construction of simple properties about resources that are represented in a graph structure (W3C RDF, 2004). Such metadata can be helpful in IR by providing more details to a search engine other than keywords.

An example of a simple RDF description used for classification, taken from the Open Directory Project (ODP), is shown below. It specifies a list of two top-level categories, “Arts” and “Business”, in the ODP ontology (ODP, 2004).

```
<RDF xmlns:r="http://www.w3.org/TR/RDF/"
     xmlns:d="http://purl.org/dc/elements/1.0/"
     xmlns="http://directory.mozilla.org/rdf">
  <Topic r:id="Top">
    <tag catid="1"/>
    <d:Title>Top</d:Title>
    <narrow r:resource="Top/Arts"/>
    <narrow r:resource="Top/Business"/>
  </Topic>
</RDF>
```

OWL provides a vocabulary for describing properties and classes and allows for greater expressive complexity than RDF alone. The vocabulary provided by OWL describes such items as relations between classes, cardinality, equality, richer typing of properties, characteristics of properties, and enumerated classes. OWL is comprised of three languages: OWL Lite for building classification hierarchies and simple constraints; OWL Description Logics (DL) provides all OWL features in addition to computational completeness (guaranteed computability of conclusions) as well as decidability (all computations will finish in finite time); and OWL Full provides all OWL features with no computational guarantees. All versions of OWL are considered extensions of RDF (W3C OWL, 2004).

An example OWL fragment, showing owl:Ontology usage, is shown below. It shows that the Class “car” is equivalent to the class “Automobile” (W3C-OWL, 2004). The extended vocabulary of OWL can be seen to be more expressive than the simple triplet structure of RDF. Equivalency expressions such as this example allow a generic logic engine to reason about “car” and “automobile” in different resources. The “car” class may be used in one resource while “automobile” class is used in another resource, yet the logic engine will identify them as equivalent. This is important in IR, since a category visualization method will be able to merge resources from multiple resources into the same category, rather than letting the user have to sift through a result set containing multiple categories that are semantically identical.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:first="http://www.w3.org/2002/03owlt/Ontology/premises001#"
    xml:base="http://www.w3.org/2002/03owlt/Ontology/premises001" >
   <owl:Ontology rdf:about="" />
   <owl:Class rdf:ID="Car">
     <owl:equivalentClass>
       <owl:Class rdf:ID="Automobile"/>
     </owl:equivalentClass>
  </owl:Class>
  <first:Car rdf:ID="car">
     <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
  </first:Car>
  <first:Automobile rdf:ID="auto">
     <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
  </first:Automobile>
</rdf:RDF>
```

Uses of ontology-based information visualization

The Semantic Web combined with ontologies can use visualization techniques in several different ways, but the visualization is dependent on characteristics of the ontologies used. In the Semantic Web, ontologies are expected to be light-weight, which means they are essentially a taxonomy with few logical relations between the ontology classes. The number of ontology instances is also expected to be large as compared to the number of classes in an ontology. Semantic Web ontologies are also expected to be incomplete and instances will have overlap between ontology classes. Incompleteness occurs when the union of all instances stored in a set of subclasses are contained in the superclass. Overlapping occurs when instances are shared by more than one class. Both of these characteristics are common to ontologies are not completely defined. Once the characteristics of an ontology are known, visualization techniques can be developed or adapted to support ontology-based visualization (Fluit et al., 2003).

There are several different ways visualization can be used with ontologies. These different ways are available because there are different life-cycle stages in the development and use of ontologies. There are three stages of an ontology life cycle: ontology development, ontology instantiation, and ontology deployment (Fluit et al., 2003). The Development stage is when the schema is first defined. Visualization is helpful to understand the concepts and relationships built into a schema, and is more beneficial as the complexity of a schema increases. The IsAViz tool is a well-known tool for visualizing RDF metadata. IsAViz is built on AT&T’s Graphviz graph visualization software library, and is a tool for browsing and authoring RDF documents. Protègè is another example of an ontology editor. It is produced as an open-source project by Stanford University, and has been extended to also support instance visualization (Mutton et al., 2003). RDF refers to the Resource Description Framework, which provides a light-weight ontology system to support the exchange of knowledge on the Web (W3C RDF, 2004). The Instantiation stage follows the schema development stage, and its primary purpose is to ensure that the population of an ontology using instances is what the ontology-designer expects. Instantiations of an ontology can be accomplished with the help of classifiers and other semi-automatic means. Visualization of an ontology instantiation can show summaries of how input data has been classified to help determine if it has been classified as the ontology developer expected. The Deployment stage uses visualization to help users of a system to effectively analyze, query, and navigate an ontology-based information space (Fluit et al., 2003). The focus on information visualization for this paper is on the last stage, deployment, where users can use information visualization techniques to search the information spaces of the Semantic Web and analyze the retrieved results.

In the Deployment stage, there are three primary areas where visualization is useful: Analysis, Querying, and Navigation (van Harmelen et al., 2001). A user performs a search on an information space, and depending on the information need, may want to use one or more of the areas to help fulfill the information need.

Analysis visualization is used to show an overview of a collection, and can also be used for data mining so that patterns can be discovered. Patterns may not reveal themselves using the traditional text display of information retrieval results. Analysis expects three parameters in order to be implemented: 1) a data set, 2) an ontology, and 3) classification (assignment of data-set objects to ontology classes). There are several strategies which have been identified for use in visual analysis (van Harmelen et al., 2001). ‘Analysis within a single domain’ allows a retrieved set of documents to be viewed using different perspectives. For example, Figures 1 and 2 show two cluster visualizations returned from a single query on job postings (van Harmelen et al., 2001). The results are grouped from two different perspectives, by economic sector (figure 1) and by region (figure 2). An information seeker can choose which perspective makes the most sense to use for their need. This is an interactive process that allows the user to pick the classes they are most interested in and then visualize how the instance documents between classes are related. If all classes are selected at once, a cluttered display may result. This is because not only are the classes shown, but also the overlap between classes is also visible. If there are a large number of classes, they may overwhelm the physical display space. This is not a problem for some visualization methods, such as the Hyperbolic Tree, which can display arbitrarily large graphs by moving the focus to different parts of the graph, but in ontology-based methods such as Cluster Maps, all classes are expected to be shown at the same time. The reason is to allow the user to gain an overview of the document space.

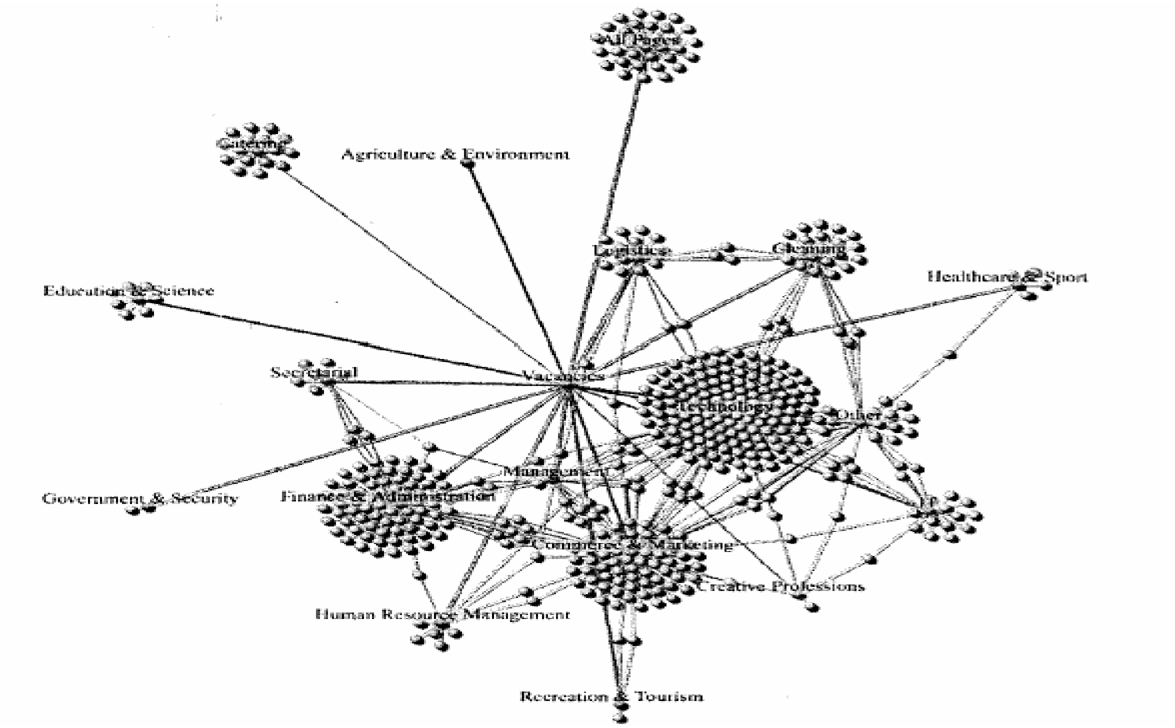

Figure 1: Data-set organized by economic sector (Van Harmelen et al., 2001)

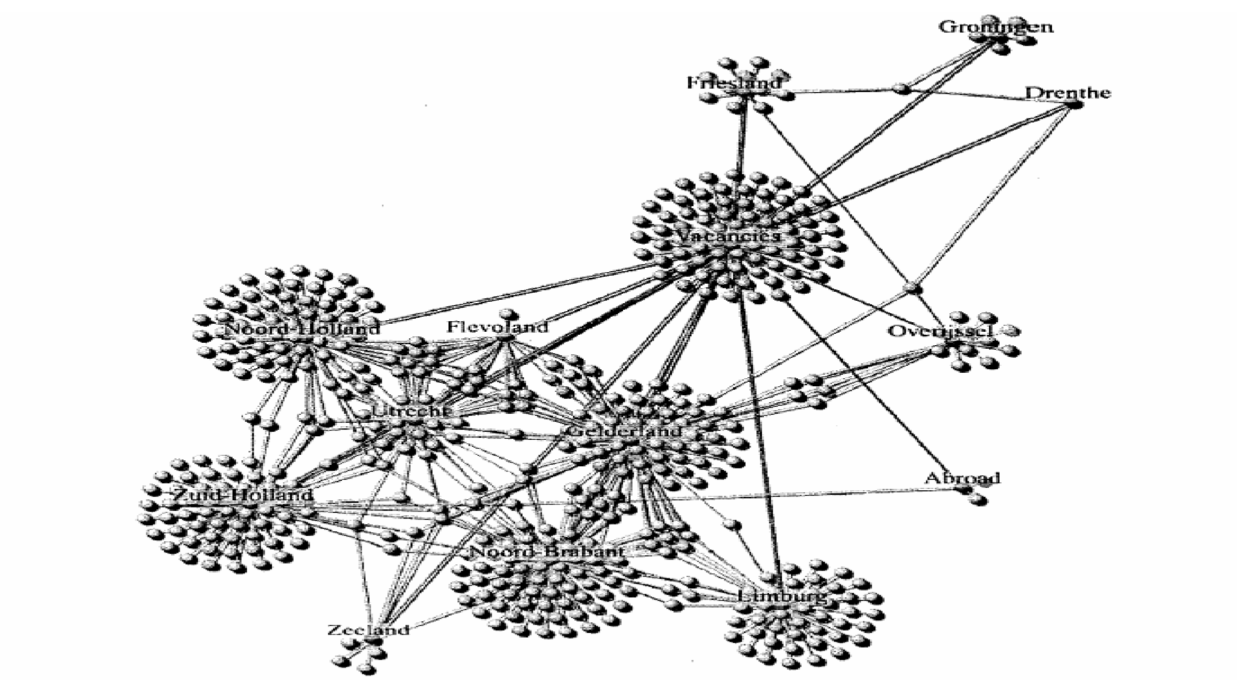

Figure 2: Same data-set as Figure 1, except it is organized by geographic region rather than economic sector (van Harmelen et al., 2001)

Another strategy than can be used is 'Comparison of different data sets'. This strategy allows two different data sets to be viewed using a single ontology. An example is the analysis of two banking web sites to see what differences can be found in terms of their business focus. Figure 3 shows two major Dutch banks' web sites. Although the terms are in Dutch, the term 'beleggen', meaning 'investment', appears isolated in the left-most Web site, while it appears connected to much more related product information in the right-most web site.

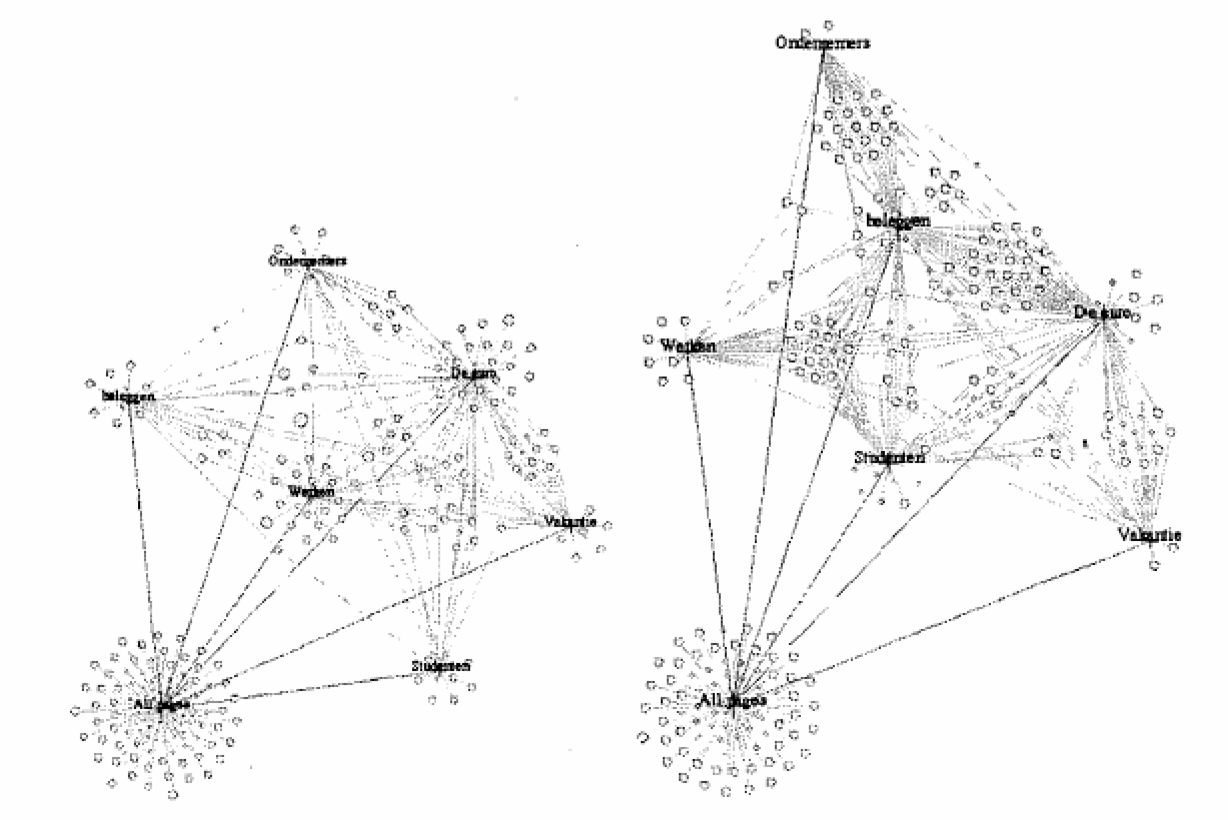

Figure 3: Two banking web sites analyzed using the same ontology (van Harmelen et al., 2001)

'Monitoring' is the third analysis strategy that can be used, and is used to show information changes as occur over a period of time. For example, a user can see how an information space has evolved over some period, and is useful to see how ontological classes are changing to include more documents, or are being de-emphasized to contain fewer documents. Figure 4 shows three ontological classes 'Insulation', 'Materials', and 'Tools' for a document system containing construction-related publications. In this example, it easily seen that there has been a sudden spike in publications concerning 'Materials' from period 1 to period 2, while the other classifications have not experienced as much of an increase. Also, this example shows the Cluster Map technique, in which overlaps between categories are shown. In the later periods shown in Figure 4, there is more overlap between the classifications, indicating that publications integrating information between the classifications is increasing each time period.

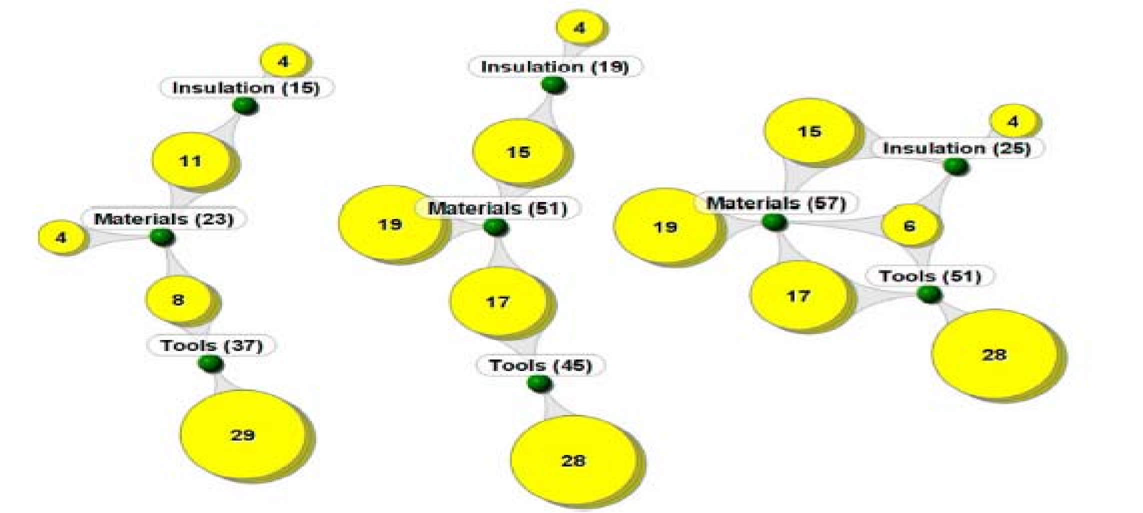

Figure 4: Three ontology classes changing over time (Fluit et al., 2003)

Query visualization uses an ontology to help construct and refine a query that is likely to satisfy a searcher's information need, as well as organize and display the query results. Fluit et al. (2003) define four stages in the query task, and three are used in visualization directly: 'Query Formulation', 'Review of Results', and 'Refinement'.

In Query Formulation, users of search tools typically express a query using search terms that may return relevant documents. Amanda Spink has studied users and their usage of search terms on the Web. She has found that less than 10% of searches use advanced search terms, such as Boolean operators and term relevance feedback. When users do use such advanced features, they often misuse them; for example, Spink found that the usage for Boolean operators is incorrect 50% of the time (Jansen et al., 2000). To overcome such query failures, ontology-based visualization can be used to assist in query construction. Fluit et al. (2003) demonstrate a user interface where users are first presented with a list of the classes in an ontology. This list essentially provides the terms that users can search with, eliminating the need for users to guess what terms or categories are stored in an information repository. Users select the classes they are interested in and then are presented with a visualization showing their results.

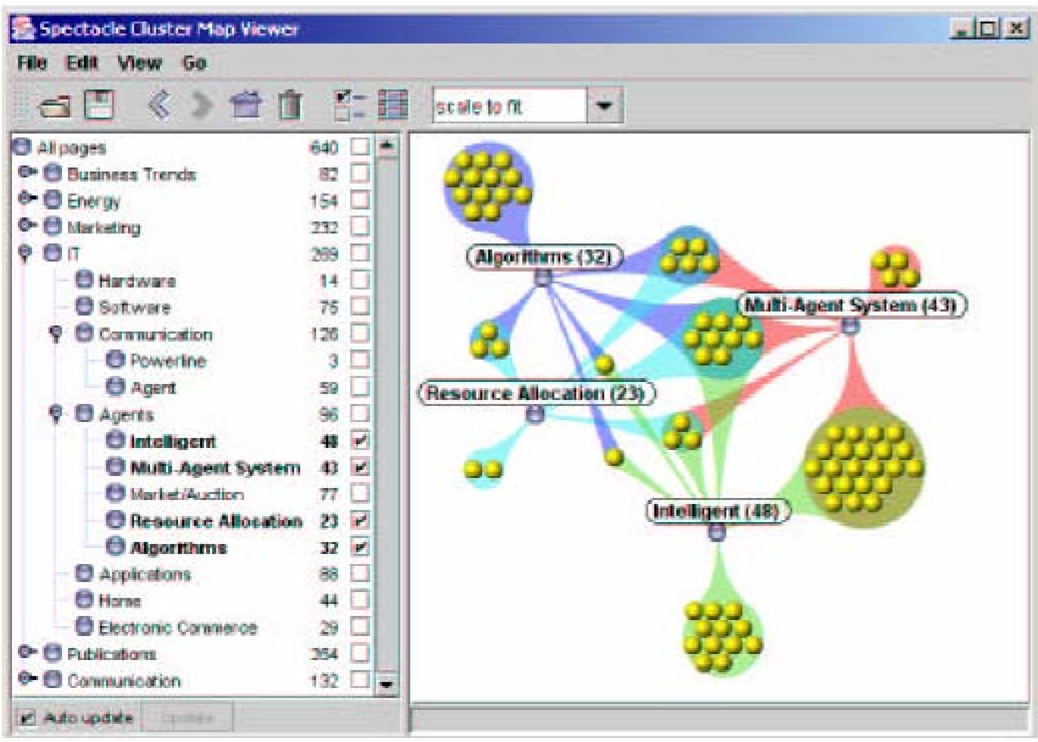

Figure 5: Ontological class selection and result list display for a user query (Fluit et al., 2003)

In the ‘Review of Results’ stage, the query results are placed into clusters based on their ontological class. Users review the visualization to gain an overall understanding of the result set, and can then drilldown further into a particular cluster, which may be an ontological class, or a cluster of documents that overlaps one or more ontological classes. This is possible because the ontology-based visualization is generated using the union of all ontological classes selected by the user, as well as all possible intersections of the selected classes (Fluit et al., 2003).

In the ‘Refinement’ stage, users are not entirely satisfied with the result set of a query. This may happen if an ontological class that is valued highly by the user returns is minimized in the visualization because other classes are more dominant. There are also two extreme cases where dissatisfaction can result: under-specification and over-specification. In under-specification the result set is too large, so users will need to choose a sub-classes of an ontology to restrict the number of items returned. In over-specification the result set is empty, that is, no items satisfy all criteria, but visualization can show items that satisfy at least some of the criteria, in effect, performing query relaxation automatically for the user.

Query navigation is used to help users navigate through information spaces and query result sets. Examples of navigation have been implemented in research systems in two different ways: automatic and semi-automatic. In semi-automatic mode, users invoke at will a visualization showing the ontology (classes and relationships) as well as ontology instances. The visualization is not the primary interface to the system. Users select the class they are interested in and a textual list of documents in the selected class is presented. The visualization based on the ontology serves as a global map for a document system. Users can view the available ontology classes to get a global view of the documents available, and can then drilldown into a particular ontology class to retrieve all of the ontology instances within the class. In the automatic method, the ontology-based visualization is the primary interface for a system. Users navigate through ontology classes and sub-classes using hierarchical browsing. Users first start with the major ontology classes and then select ontology sub-classes to choose more specific information areas (Geroimenko & Chen, 2002).

Existing information visualization techniques and Cluster Map

There are several visualization techniques that have been applied to web resource visualization. Common examples and techniques include the Hyperbolic Tree for displaying hierarchical trees, The Brain for visualizing arbitrary-sized graphs, and Kohonen Self-Organizing Maps (Fluit et al., 2003). These methods are characterized by the fact they are largely focused on navigation, showing aspects of the information infrastructure rather than the information (instance data) itself (Fluit et al., 2003). These existing techniques can be adapted to show ontology metadata as well as ontology instances, but the focus is usually on syntactic, rather than semantic, structures. For example, links between resources are often used, which do show some semantics by way of relationships, but these links are usually implicit rather than explicit and well-defined (Fluit et al., 2003). For example, links between web pages are implicit, while ontology definitions of resource linkages are explicit.

van Harmelen et al. (2001) identify several issues with these existing techniques that make them sub-optimal for displaying ontological data. The Brain, which shows graphs, shows the local neighborhood of a graph and is therefore difficult for a user to maintain global orientation. The Brain also is designed to display arbitrary graphs and makes no assumptions on what the graph represents. A visualization method designed for ontologies would take advantage of the information stored in the ontology to enhance the graph display. Hyperbolic Trees also provide general-purpose visualizations of graphs and do not make use of the information defined in an ontology. The focus is on some parts of the tree while excluding others. This allows very large trees to be displayed, but the use cannot gain a global view when the tree is larger than the display. Kohonen maps show organization of ontological class instances, but do not show the overlap that occurs when instances belong to more than one class. For well-defined ontologies, this may not be an issue since an instance is more likely to belong to a single class, but for ontologies that are not completely-defined, an instance is more likely belong to more than one class.

The Cluster Map visualization method was designed to address some of the shortcomings of existing visualization techniques when applying both ontologies and their instances. The Cluster Map provides instances as part of the displayed graph. Documents belonging to a query are shown within the graphics itself, as opposed to other visualization techniques, which will often display the instance data (documents) separately in a text list, usually below the graphic. Figure 5 shows a sample Cluster Map retrieval display. The Cluster Map is displaying 1) the ontological classes that the retrieved instance documents belong to, as shown by the larger bubbles that have also have text labels indicating the class name as well as the class cardinality, 2) the instance documents, represented by balls within the bubbles, and 3) overlapping instance documents, as shown by the smaller bubbles interconnected between two larger bubbles.

The display of the Cluster Map classes is done using a variant of the spring-embedder algorithm, and so the display of class and overlapping classes is not arbitrary. The spring-embedder algorithm works by defining nodes that repel one another and edges that attract one another. The opposing and attracting forces will generate a stable configuration that reflects the semantic closeness of ontology classes and class instances. Semantic closeness is defined as 1) two classes are close when they share many instances, and 2) two instances are close when they belong to the same class (van Harmelen et al., 2001).

A large assumption made by the developers of Cluster Map is that ontologies on the Semantic Web will be light-weight ontologies. Light-weight ontologies describe a domain using classes (concepts) and hierarchical relationships between classes (Fluit et al., 2003). The light-weight ontology can also be described as a taxonomy or classification system. Two major characteristics of such light-weight ontologies are that they are incomplete and are likely to have overlap. Overlap is when instances belong to two or more classes and there is no specialization relationship between them. Incompleteness occurs when a set of subclasses of a particular class does not contain all of the instances of the class. The assumption that lightweight ontologies would be largely used on the Semantic Web is based largely on the Cluster Map developers' experiences to date with a variety of Semantic Web applications (Fluit et al., 2003). Examples of existing lightweight ontologies are the web directories of Yahoo! (http://www.yahoo.org/) and the Open Directory Project (http://www.dmoz.org). The opposite of a lightweight ontology would be a well-defined ontology that includes complex concept hierarchies, properties, value restrictions, and relations between concepts (van Harmelen, 2001). A further assumption for Cluster Map is that the number of instances will be very large as compared to the number of classes (Fluit et al., 2003).

The advantages of Cluster Map over existing visualization methods, such as the Hyperbolic Tree, are 1) all of the classes and instances are displayed at one time, so that the user has global view of the entire document space; 2) non-tree like hierarchies can be displayed, not just graph structures; and 3) overlap between different classes is exploited.

Conclusion and Future Work

The Cluster Map visualization method provides a way to take advantage of ontologies when presenting retrieval results, and this method is an improvement over existing general-purpose graph visualization methods. The Cluster Map, however, can also be improved.

There are several areas where the Cluster Map has weaknesses. It relies heavily on light-weight ontologies, which may limit its useful in complex ontologies. It also expects the number of classes within an ontology to be small as compared to the number of instances. The implication is that some classes will be densely populated rather than further specialized. Related to this, Cluster Map has difficulty scaling to large number of instances within a class. The display shows a bubble (class) containing balls (documents). If the number of documents within a class is very high (say several hundred), then the display becomes overrun with large bubbles containing many balls. A way to get around this scalability issue is to display the cardinality of the class without the ball display. The cardinality is already displayed, so removing the ball and scaling the bubble appropriately is a solution. However, this removes a desirable feature of Cluster Map: showing the instance documents for immediate access. If the documents are not shown graphically, then they will need to be presented textually, probably as a list. Seeling et al. (2003) designs a system that uses this approach.

Seeling has also identified a further problem with Cluster Map: there is no way to view document similarity. Seeling writes that the Cluster Map visualization capabilities for document analysis are subordinate to navigation and querying. The document similarity is usually displayed in other IR systems by giving its ranking. A query can generate a text-based result list of documents that are ordered by their ranking. The ranking can be determined several ways, but a common way in keyword-based search is by using search term frequency. The higher the frequency of the search terms within a document, the higher the document ranking with a result set. Seeling addresses the document similarity problem by generating a document map visualization which appears very close to a Kohonen map. When the user selects a particular class, the documents in that class are highlighted against all documents in the document space. Users can then visually see the class document instances as well as how well the documents are related to one another. Figure 6 shows a sample visualization using this technique.

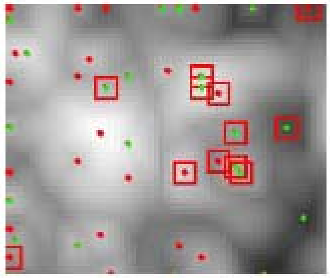

Figure 6: Document space showing document similarity for a particular ontology domain (Seeling et al., 2003).

While showing document similarity using this visualization may be accurate, it does not seem as intuitive as the Cluster Map display. An extension can be made to the Cluster Map display to show document similarity while at the same time preserving the advantages of Cluster Map. In a class display (shown by a bubble), there can be embedded concentric circles showing similarity. Similar documents can be grouped together. In a typical IR keyword search, the top ranked documents can be shown in one circle, the next third documents in another circle, and so forth; in ontology-based usage, each circle can represent a set of documents that match the ontology class in varying degrees. The concentric circles can continue to an unlimited nesting. This type of display is termed a volvox display, named by McCain after a similarly-shaped microorganism (Small, 1999). An example of the volvox display, from a commercial search and visualization application called Grokker, is given in Figure 7. One needs to envision the Cluster Map display of Figure 5 and replace the bubbles with the volvox display shown in Figure 7. An example of just such a display might appear as in Figure 8, which should be compared with the standard Cluster Map display shown in Figure 5. The advantage of merging the Cluster Map and Volvox display techniques at the class level is that document similarity can be easily shown. The Cluster Map visualization method currently has no facility for displaying document similarity.

The Cluster Map can also extend the volvox display so that top-level classes are shown and sub-classes are contained within the bubble. As the users drill-down into the circles in each subclasses, eventually they would discover the document instances within each subclass. The advantage here is that it would allow Cluster Map to scale to handle many class instances instead of just displaying class cardinality when the number of instances in a class is too large to physically display.

In addressing either issue, scalability or document similarity, it appears that the volvox display might be a useful evolution to the Cluster Map display to address some of its weaknesses.

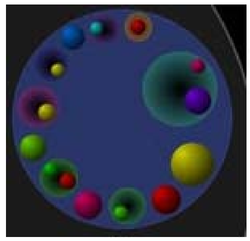

Figure 7: Volvox display (Groxis, 2004).

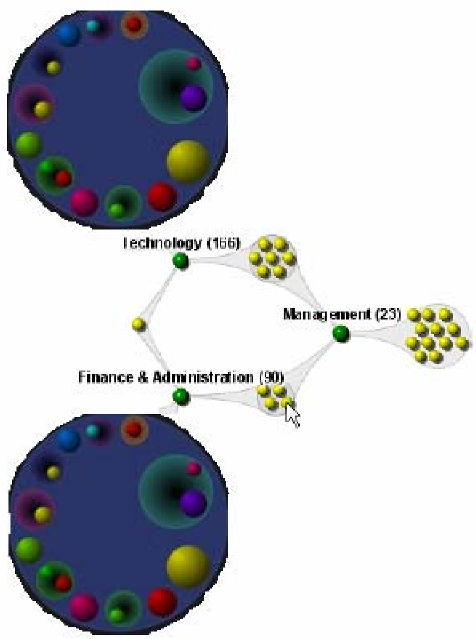

Figure 8: Possible document similarity display using a volvox enhancement.

Evolutional improvements to the Cluster Map are useful, but it is important to realize that the Cluster Map and its evolution is not the final method in ontology-based visualization for IR. Mutton and Golbeck (2003) point out that while the Cluster Map is useful for navigating search results and displaying a clear and intuitive graphic depicting relationships between instance data and their class memberships, grouping by classes is not always desirable. There are alternate ways of viewing ontology-based data rather than by class membership. For example, Mutton and Golbeck (2003) suggest that if you are working with data about people, projects, and organization-produced papers, it might be better to visualize the people and papers together in order to see their interaction. More generally, classes (concepts) related to one another through properties and sub-classes can be clustered using these semantic links. The Cluster Map is useful for categorizing the results of an IR query using lightweight ontology information. Additional visualization methods are also possible for ontologies which are more complex and more well-defined than simple taxonomies. Figure 9 shows a graph structure of semantic data, where the chains of structures show sequentially linked concepts, as well as centers of related concepts. The graph also uses the spring-embedded algorithm to place semantically-related data together (Mutton and Golbeck, 2003). Such a graph structure may more completely show a complex ontology, but does not seem as immediately intuitive as the Cluster Map display. The direction of visualization of the Semantic Web and retrieval results will depend on the complexity of the ontologies generated by users and organizations. If the Cluster Map developers are correct in their assumption that most Semantic Web ontologies will be lightweight (essentially taxonomies), then the Cluster Map and its variants will be particular useful. If, on the other hand, most ontologies are designed as well-defined, complete ontologies, then the visualization method proposed by Golbeck and Mutton may become more commonplace.

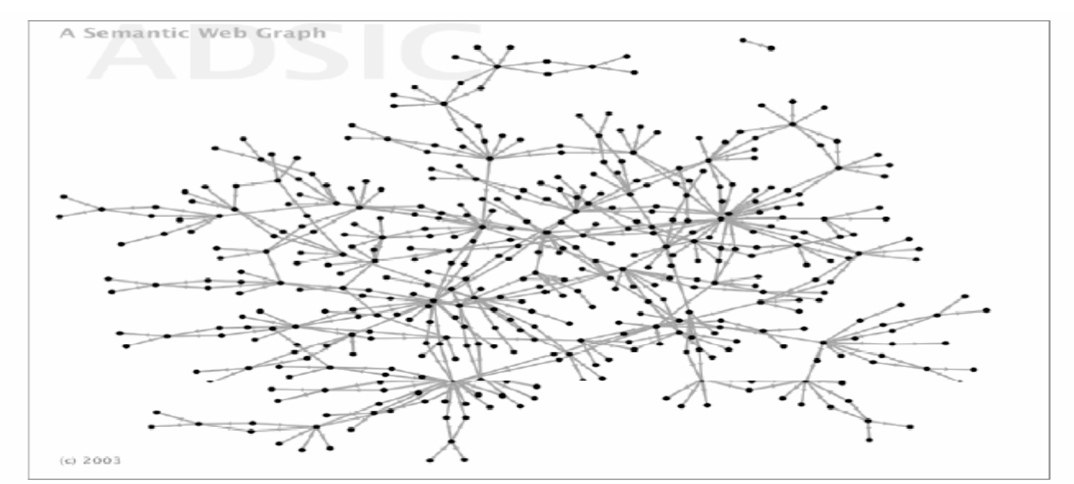

Figure 9: Graph display of semantic data (Mutton and Golbeck, 2003).

<u>References</u>:

Card, S., Mackinlay, J., Shneiderman, B. (eds) (1999). <u>Readings in Information Visualization: Using Vision to Think</u>. Morgan Kaufmann, San Francisco, ISBN 1-55860-533-9.

*Fluit, C., Sabou, M., van Harmelen, F. (2003). Supporting User Tasks through Visualisation of Light-weight Ontologies. <u>Handbook on Ontologies in Information Systems</u>. Springer-Verlag.

*Garcia, E., Sicilia, M. (2003). User Interface Tactics in Ontology-Based Information Seeking. *PyschNology Journal*, 1(3), 242-255.

*Geroimenko, V., Chen, C. (eds) (2002). Ontology-based Information Visualization. <u>Visualising the Semantic Web</u>. Springer Verlag, London, ISBN 1-85233-576-9.

Groxis 2004. Groxis web site. Retrieved May 23, 2004, from http://www.groxis.com/service/grok/g_prod_grok_screens.html.

Gruber, T.R., (1993). A translation approach to portable ontologies. *Knowledge Acquisition, 5*(2), 199-220.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real Users, and real needs: A study and analysis of user queries on the web. Information Processing and Management. 36(2), 207-227.

*Mutton, P.; Golbeck, J. (2003). Visualization of Semantic Metadata and Ontologies. *Proceedings of Seventh International Conference on Information Visualization*, 300-305.

Open Directory Project (2004). Retrieved May 23, 2004, from <u>http://www.</u>dmoz.org.

*Seeling, C., Becks, A. (2003). Exploiting Metadata for Ontology-Based Visual Exploration of Weakly Structured Text Documents. *Proceedings of Seventh International Conference on Information Visualization*, 652-657.

Small, H. (1999). Visualizing science by citation mapping. *Journal of the American Society for Information Science, 50*(9), 799-813.

*van Harmelen, F.; Broekstra, J.; Fluit, C.; ter Horst, H.; Kampman, A.; van der Meer, J.; Sabou, M. (2001). Ontology-based Information Visualization. *Proceedings of Fifth International Conference on Information Visualisation Conference*, 546–554.

<u>World Wide Web Consortium</u> (W3C-OWL) 2004. W3C Web-Ontology (WebOnt) Working Group web site. Retrieved April 17, 2004, from <u>http://www.w3c.org/2001/sw/WebOnt/</u>.

<u>World Wide Web Consortium</u> (W3C-RDF) 2004. W3C Resource Description Framework (RDF) web site. Retrieved April 17, 2004, from <u>http://www.w3c.org/RDF</u>.

* indicates primary paper used in research