

3 Ontology-based Information Visualization: Towards Semantic Web Applications

Christiaan Fluit, Marta Sabou and Frank van Harmelen

3.1 Introduction

The Semantic Web is an extension of the current World Wide Web, based on the idea of exchanging information with explicit, formal and machine-accessible descriptions of meaning. Providing information with such semantics will enable the construction of applications that have an increased awareness of what is contained in the information they process and that can therefore operate more accurately. This has the potential of improving the way we deal with information in the broadest sense possible, e.g. better search engines, mobile agents for various tasks, or even applications yet unheard of. Rather than being merely a vision, the Semantic Web has significant backing from various institutes such as DARPA, the European Union and the W3C, which all have formed a variety of Semantic Web activities.

In order to be able to exchange the semantics of information, one first needs to agree on how to explicitly model it. Ontologies are a mechanism for representing such formal and shared domain descriptions. They can be used to annotate data with labels (metadata) indicating their meaning, thereby making their semantics explicit and machine-accessible.

Many Semantic Web initiatives emphasize the capability of *machines* to exchange the meaning of information. Although their efforts will lead to an increased quality of the application's results, their user interfaces often take little or no advantage of the increased semantics. For example, an ontology-based search engine could use its ontology to enrich the presentation of the resulting list to the end user, e.g. by replacing the endless list of hits with a navigation structure based on the semantics of the hits.

Visualization is becoming increasingly important in Semantic Web tools. In particular, visualization is used in tools that support the development of ontologies, such as ontology extraction tools (OntoLift, Text-to-Onto) or ontology editors (Protege, OntoLift). The intended users of these tools are ontology engineers that need to get an insight in the complexity of the ontology. Therefore, these tools employ schema visualization techniques that primarily focus on the structure of the ontology, i.e. its concepts and their relationships. We presented a detailed overview of these tools in (Fluit et al., 2003).

The Cluster Map visualization technique, developed by the Dutch software vendor Aduna¹, bridges the gap between complex semantic structures and their simple, intuitive user-oriented presentation. It presents semantic data to end users that want to leverage the benefits of Semantic Web technology without being burdened with the

¹ <http://aduna.biz>

complexity of the underlying metadata. For end users, instance information is often more important than the structure of the ontology that is used to describe these instances. Accordingly, the Cluster Map technique focuses on visualizing instances and their classifications according to the concepts of the ontology.

We have reported in previous work (Fluit et al., 2002; Fluit et al., 2003) on case studies that exploit the expressive power of this technique. Since then, the growth of the Semantic Web has made it possible to take this technology a step further and integrate it in three different applications. Two of them are employed within Semantic Web research projects. The third is a commercial information retrieval application. These applications exhibit the characteristics of a typical Semantic Web tool: they provide easy (visual) access to a set of heterogeneous, distributed data sources and rely on Semantic Web encoding languages and storage facilities for the manipulation of the visualized data.

This chapter is centred around the description of these three applications. First, we will explain the contents of the Cluster Map visualization and the kind of ontologies it visualizes in section 3.2. Section 3.3 presents the three real-life applications that incorporate the visualization. These two sections lead to a discussion in section 3.4 on how the visualization can support several user tasks, such as analysis, search and exploration. Some considerations for future work and a summary conclude this chapter.

3.2 Cluster Map Basics

The Cluster Map is used to visualize light-weight ontologies that describe a domain through a set of classes (concepts) and their hierarchical relationships. Also known as taxonomies, such ontologies are frequently used in several domains (biology, chemistry, libraries) as classification systems. Information architects consider taxonomies as basic building blocks, representing the backbone of most web sites. Non-formal taxonomies are already widely used in web applications for product classification (e.g. Amazon) or web-directories (e.g. Yahoo, Open Directory Project (ODP)). Taxonomies are also part of Semantic Web standards such as RDF and Topic Maps.

Due to the specialization relationship that is encoded in the hierarchy, the set of objects in a subclass is a subset of its superclass. The set of subclasses of a class is *incomplete* when their union does not contain all the objects of the superclass. Classes that share instances are *overlapping* if no specialization relationship holds between them. These characteristics are very common for taxonomies. However, they are difficult to show satisfactorily in textual representations. Our visualization offers an alternative in this matter.

The Cluster Map visualizes the instances of a number of selected classes from a hierarchy, organized by their classifications. The Cluster Map in Figure 3.1 shows a collection of job offers organised according to a very simple ontology. Each small yellow sphere represents an instance (a job offer). The classes are represented as rounded rectangles, stating their names and cardinalities. Directed edges connect classes and point from specific to generic (e.g. *IT* is a subclass of *Job Vacancies*). Balloon-shaped edges connect instances to their most specific class(es). Instances

with the same class membership are grouped in *clusters*. Our example contains six clusters; two of them represent overlaps between classes.

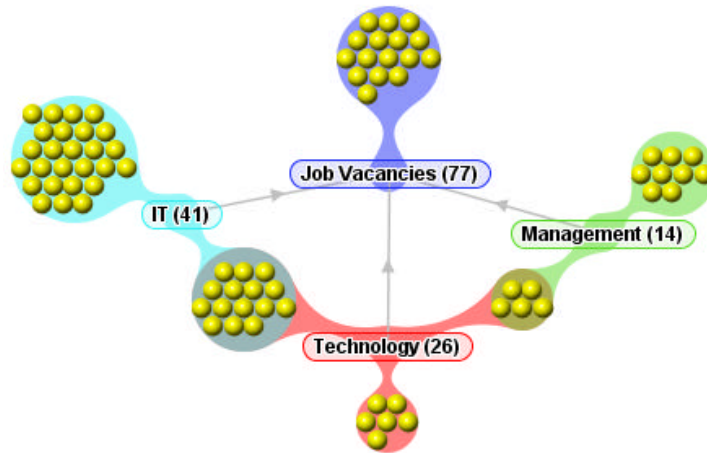


Figure 3.1 An example Cluster Map.

The organisation of the graph is computed using a variant of the well-known spring-embedder algorithm (Eades, 1984). On the one hand the class and cluster nodes repel each other. On the other hand the edges, connecting two classes or clusters to their classes, produce an attractive force between the connected nodes; i.e. they work as “springs”. The layout algorithm has been optimised for this particular kind of graph, both *qualitatively* (using the semantics of the graph to obtain a more insightful visualization) and *quantitatively* (obtaining a layout fast enough to permit interactive usage).

The added value of our visualization lies in its expressivity. The classes and their relationships (the vocabulary of the domain) are easy to detect. Also, it is immediately apparent which items belong to one or multiple classes, which classes overlap (e.g. *Technology* and *Management*) and which do not (*IT* and *Management*). The cardinality of classes and clusters is visible: the top class has the most objects, *Technology* shares more objects with *IT* than with *Management*. The subclasses of the root class are incomplete as their union does not cover the superclass: some members of *Job Vacancies* were not further classified.

Another interesting aspect of the visualization is that geometric closeness in the map is related to semantic closeness. This is a consequence of the graph layout algorithm. Classes are semantically close if they share many objects. Indeed, the more objects two classes share, the closer they are represented in the graph. Objects are semantically close if they belong to the same class(es). Indeed, objects with the same class membership are clustered.

The Cluster Map visualization is highly configurable. Several graph colour schemes are available, each tuned towards a different task, e.g. by displaying each class using a unique colour (see Figure 3.1) or by indicating cluster relevance using colour brightness (see Figure 3.5). Filters can reduce the complexity of the visualization, e.g. by removing all clusters representing the overlap of less or more than n classes, or all clusters with cardinality above or below a certain threshold. The scalability of the

visualization can be improved by displaying a cluster as a single graphical entity with its cardinality indicated, see (Fluit et al., 2003) for an example.

3.3 Applications

In this section, we describe three applications in which the Cluster Map is prominently used. The first two are results from research projects, the last one is a commercial off-the-shelf application.

3.3.1 The DOPE Browser

The exploration of large information spaces is a difficult task, especially if the user is not familiar with the terminology used to describe information. Conceptual models of a domain in terms of thesauri or ontologies can remedy this problem to some extent. In order to be useful, there is a need for interactive tools for exploring large information sets based on conceptual knowledge. The DOPE project² has developed a thesaurus-based browser that supports a mixed-initiative exploration of large online resources in the domain of drugs and diseases. The DOPE Browser provides support for thesaurus-based search and topic-based exploration of query results, and makes extensive use of Cluster Maps for both visualizing and exploring query results.

The DOPE Browser supports the user in navigating the information space by providing the following functions:

- ? Query Formulation: query interfaces that contain a graphical presentation of the available knowledge (ontology) make the query formulation task much easier as the user is already acquainted with the used terminology. The ideal situation would be that queries can be formulated visually.
- ? Query Result Presentation: inadequate presentation of query answers shadows the performance of many search engines as the user is overwhelmed with results without being able to pick the part that is of interest for him. Graphical presentations can explain the relation of the answer to the original terms of the query.
- ? Query Reformulation: Very often the user is not satisfied by the returned answer: there are either too many items or no items at all. Graphical presentation of results allows a user to perform query relaxation, query broadening and query narrowing.

Query Formulation

The DOPE Browser uses Elsevier's EMTREE thesaurus to suggest disambiguations of query terms. The EMTREE thesaurus contains about 48.000 preferred terms and 200.000 synonyms in the domain of drugs and diseases, organized in a multilevel hierarchy. The user can browse the hierarchy and select a term or type in a free term that is matched against EMTREE and a corresponding term is proposed. Figure 3.2 shows the state of the DOPE Browser after the following dialogues:

1. The user has first entered an initial search term (the term "aspirin" in the top-level entry-field).

² A collaboration between Elsevier, Collexis, Aduna and the Vrije Universiteit Amsterdam

- This search term was subsequently disambiguated to the EMTREE term “acetylsalicylic acid” by offering to the user different senses of the term that occur in the EMTREE thesaurus.

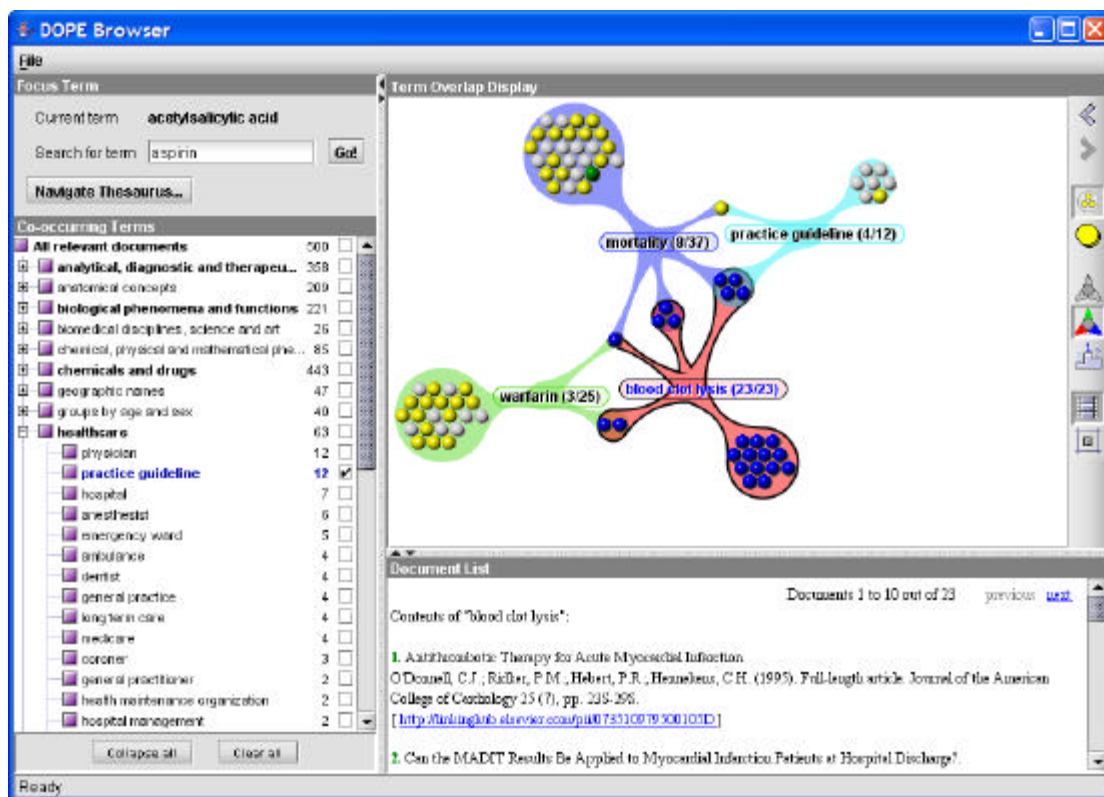


Figure 3.2 The DOPE Browser’s main screen.

Query Result presentation and exploration

The DOPE Browser presents a visual representation of terms and documents in a network where documents are connected to the terms they contain (compare (Hemmje et al., 1994) and (Korfhage, 1991)). The advantage of this representation is that it provides an overview over the result set rather than individual information for each document.

Again in terms of the example shown in Figure 3.2:

- The total number of hits for this term (500, in this example) is presented to the user in terms of which retrieved documents belong to which EMTREE classes (the hierarchical pane on the left). Note that this only shows the subset of EMTREE classes occurring in this document set.
- The user can then explore the structure of this result-set by selecting the EMTREE classes he is interested in.
- The relations between the selected document sets are displayed as a Cluster Map in the right-hand pane.

Steps 4 and 5 above enable an interactive and semantically rich exploration of the result set of a query. This presentation should be compared with the typical flat list of 500 answers returned by traditional retrieval engines. For example, in Figure 3.2, the user has checked the terms *mortality*, *practice guideline*, *blood clot lysis* and

warfarin. The visualization shows that within the set of documents about aspirin there is some significant overlap between the terms *blood clot lysis* and *mortality*, and that four of the *practice guidelines* documents relate to these two topics as well.

Various ways exist to further explore this graph. The user can click on a term or a cluster of articles to highlight their spheres and list the document metadata in the Document List panel at the lower right. Moving the mouse over the spheres reveals the same metadata in a tool tip. The visualizations can also be exported to a clickable image map that can be opened in a web browser.

A small-scale evaluation of this user interface was conducted at a Drug Discovery conference held in the US in 2003. Results are reported in (Stuckenschmidt et al., 2004).

3.3.2 Xarop/SWAP: Peer-to-peer knowledge management

The tremendous excitement generated by Napster, Gnutella, Kazaa and the like show the large potential of the peer-to-peer (P2P) paradigm. A case-study of the EU-funded SWAP consortium³ has exploited the P2P approach to solve a distributed knowledge management problem.

Knowledge management solutions relying on central repositories sometimes have not met expectations, since users often create knowledge ad-hoc using their individual vocabulary and using their own individual IT infrastructure (e.g., their laptop). (Bonifacio et al., 2002) provide an explanation for the failed cases. They argue that traditional knowledge management systems take on a traditional managerial control paradigm, while subjectivity and sociality are essential features of knowledge creation and sharing. From that point they overhaul the traditional architecture of knowledge management systems towards a more decentralized architecture.

The case study that was considered in the SWAP project is based in the tourism sector of the Balearic Islands. On the Balearic Islands, many players (both governmental, commercial and even academic) are working together to further the goals of the tourism industry, which is the major source of income for the Islands. The parties involved are very heterogeneous (government departments, tour operators, hotel chains, economics researchers), and geographically distributed (among the different Balearic Islands). Due to the different working areas and objectives of the collaborating organizations, it proved impossible to set up a centralized knowledge management system or even a completely centralized ontology. The case study partners asked explicitly for a system without a central server, where knowledge sharing is integrated into the normal work, but where very different kinds of information could be easily shared with others.

From a technical point of view, the different organizations can be seen as independently operating nodes within a knowledge network. Nodes can join or disconnect from the network at any moment and can live or act independently of the behavior of other nodes in the system. A node may perform several tasks. The most important one is that it acts as a peer in the network, so it can communicate with other nodes to achieve its goals. But apart from that it may act as an interface to interact

³ <http://swap.semanticweb.org>

with the human user of the network, or it may access knowledge sources to accomplish its tasks. One node may have one or more knowledge sources associated with it. These sources contain the information that a peer can make available to other peers. Examples are a user's file system and mail folders or a locally installed database.

In the Xarop system that was built according to the view above, nodes wrap knowledge from their local sources (files, e-mails, etc.). Nodes ask for and retrieve knowledge from their peers. For communicating knowledge, Xarop transmits RDF structures, which are used to convey conceptual structures (e.g., the definition of what an indicator for air travel is) as well as corresponding data (e.g., data about the number of arrivals by plane). For structured queries as well as for keyword queries, Xarop uses SeRQL, an SQL-like query language that allows for queries combining the conceptual structure of the information with keyword occurrences.

There are three modes of using the system:

- ? The basic use of the system is to search for information. One way of doing this is by manually browsing other peer folders. Xarop works in this scenario in a way similar to filesystems, with the exception of using P2P technology and unifying view from different sources (like different peers and from the other hand from folders / files and bookmarks from Web browser). (XXX Unclear description XXX)
- ? More advanced information search is possible using full-text search. This way users could easily find all documents containing words, eg. typing “tourism” would result in finding all documents containing the word “tourism” in its content.
- ? The most powerful mechanism is topic search. In this case, a user is interested in some topic and asks for all documents related to that topic (eg. tourist arrivals). The system allows this, relying on mechanisms for automatic assignment of documents into topics. This mechanism is multi-lingual, so the query would take into considerations documents in any of the supported languages – namely, English, Spanish or Catalan. As automatic mechanisms may not be perfectly accurate, the user is able to manually change document-topic alignments, and these alignments are considered when performing the search.

Figure 3.3 shows the main interface of Xarop, and it covers the following functionalities: sharing folders, full indexing of documents, automatic classification of documents under concepts of the local ontology, manual classification of documents under concepts of the local ontology using drag and drop, browsing folders, keyword-based search, conceptual search, presenting and visualizing results, configuring security access and downloading files.

The Cluster Map is embedded in a highly interactive GUI, which is designed for browsing-oriented exploration of the populated ontology. Users can browse shared folders of other peers at the upper left. Alternatively, they can enter keyword queries or select concept terms at the lower left and indicate whether these should be sent to the local node, the entire network or whether specific peers should be addressed. Note that, unlike some of the popular P2P file sharing applications, the specific peers

available on the network are prominently used in the interaction, thereby stimulating community awareness.

The results of these queries are shown as populated classes in the Cluster Map on the right. The software can animate the transition from one visualization to the next. Through interaction a user can also retrieve information about the specific documents that are contained in a class or cluster. Furthermore the visualization can be fine-tuned in several ways, in order to support certain tasks or improve scalability. In the context of Xarop it was important to account for the particularities of P2P systems. Hence, the results are marked with the peer name they are coming from. Results are added to the Cluster Map incrementally, since not all peers answer at the same time. New results are highlighted. The search can be stopped when the user is satisfied.

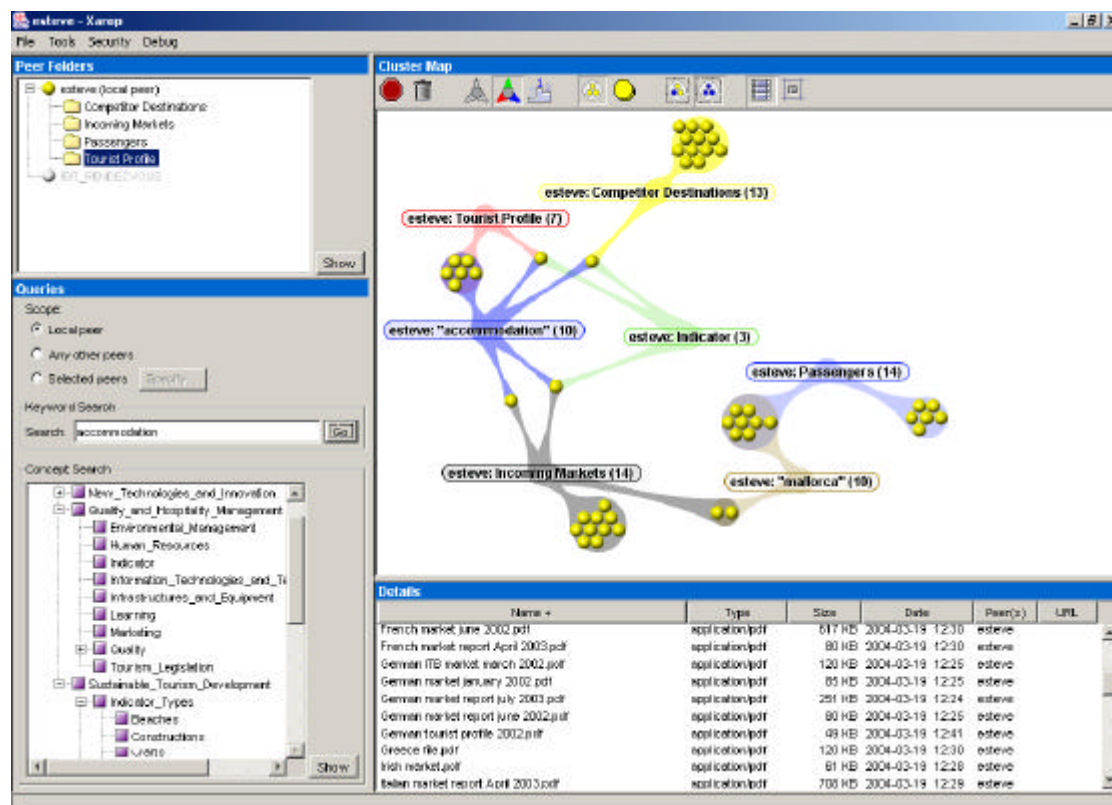


Figure 3.3 Xarop's main screen.

3.3.3 Aduna AutoFocus

The last system we discuss is AutoFocus, a commercial application developed at Aduna that brings semantic, multi-dimensional information visualization and navigation to everyone's desktop. It lets users oversee and access the overwhelming amount of information that they may have in their personal information sources.

AutoFocus discloses a user's personal information sources, such as files stored on a hard drive, emails, frequently used websites and intranets. Its interface lets the user navigate through the various types of metadata extracted from these sources and visualize their document sets.

More precisely, users explicitly define their sources in AutoFocus, e.g. by specifying one or more folders on the hard drive or in a mail box. The system will subsequently scan the files in these folders and determine attributes such as file type, size and date. Depending on the file type, the document text and any available metadata such as titles and authors may be extracted as well. All extracted information is indexed for fast retrieval in a local Sesame (Broekstra et al., 2002) RDF repository.

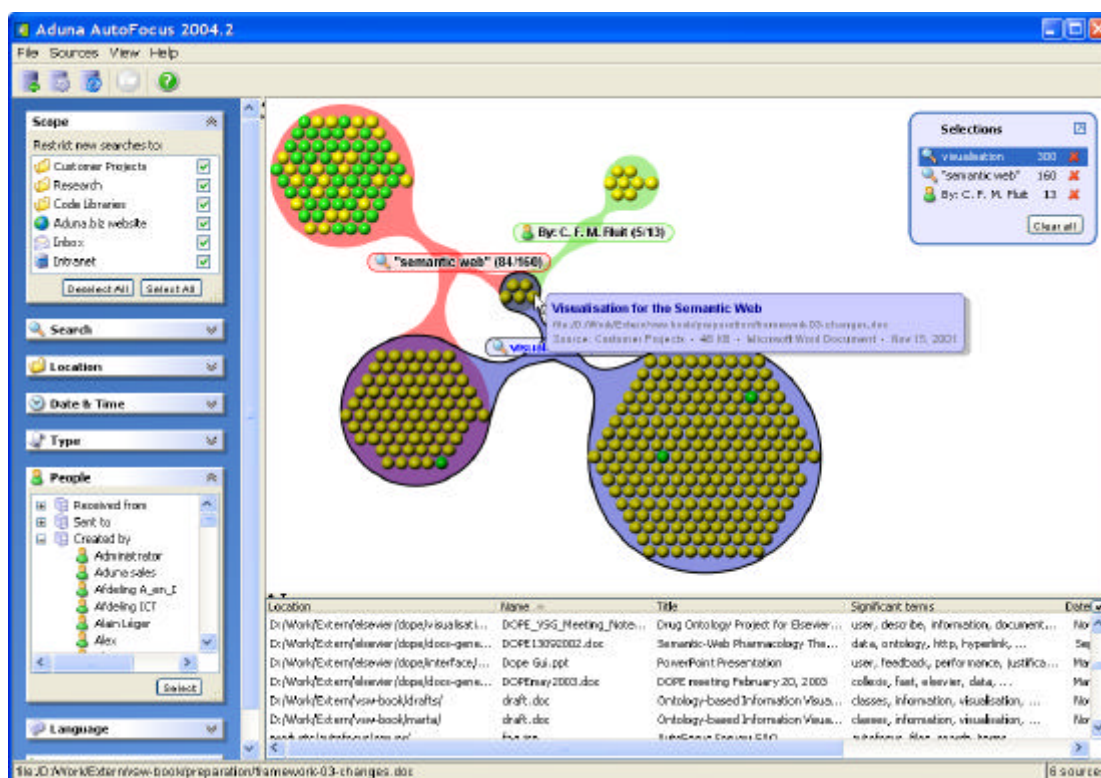


Figure 3.4 AutoFocus' main screen.

Figure 3.4 shows the main screen, where the user has defined six different information sources. At the left all the available metadata facets are shown. Each window allows the user to express a query using a facet-specific interface. The uppermost window is an exception since it restricts query evaluation to the selected sources. The query results are shown as populated classes in the Cluster Map at the right. In this example we see the results of three queries: a keyword query for “visualization”, a phrase search for “semantic web” and an author query for “C. F. M. Fluit”. The latter query is matched against the author metadata such as can be found in e.g. MS Office, PDF and HTML files.

The development of AutoFocus has been influenced in numerous ways by the systems developed within the DOPE and SWAP projects described above. For example, the AutoFocus architecture mimics the DOPE architecture in that it uses virtual Sesame repositories to store and provide access to the indexed text and metadata of each information source, while below the surface a number of different storage formats is used, each optimized for a specific type of data.

Using Sesame as the single point of access in the back-end has also led to the development of another product, Aduna Metadata Server, which indexes information

sources and provides programmatic access to its metadata using standard Sesame access mechanisms (typically XML over HTTP, but others are possible as well). This way the result of an indexation process, which can be expensive to establish and complex to maintain, can be shared in enterprise environments, where shared network drives and intranet servers are in common use.

The use of the Sesame API provides great flexibility in creating customer solutions. For example, an Aduna Metadata Server can easily be adapted or even replaced by a tailor-made Sesame server that discloses the documents in a document management system. AutoFocus clients can easily query these servers without adaptations. On the other hand, AutoFocus can just as easily be replaced by a custom-made client, as long as it uses Sesame's APIs to query the servers. The only assumption (for now) is that the information in the repositories conform to the AutoFocus RDF schema.

The SWAP project provided an environment for us to experiment with handling asynchronous query responses in a highly interactive user interface. This is a real issue in P2P applications, where you never know when and if a response will arrive. The same is true to a lesser extent for AutoFocus when the user makes use of Metadata Server repositories. Typically a trade-off needs to be made between updating the interface with new results as fast as possible and buffering incoming results to prevent overwhelming the user with changes.

Finally, both the DOPE and the SWAP projects resulted in extensive experiences with designing and implementing a user interface for browsing faceted metadata with information visualization support.

Recently, we have seen an increase in the number of applications entering the desktop search market. It is being targeted by major technology suppliers such as Google and Apple as well as a number of smaller and lesser known companies. Relatively few of these applications use information visualization techniques, with Groxis being a well-known exception. The use of Semantic Web standards in this area is still limited to research applications. With AutoFocus we believe that we have made an application that stands out of the competition by the use of unique information visualization techniques and Semantic Web standards that provide a basis for sharing and interoperability in enterprise environments.

An evaluation copy of AutoFocus can be downloaded from <http://aduna.biz>.

3.4 Uses of Ontology-based Visualization

So far we have demonstrated the use of our visualization in real life applications. In what follows we will discuss how our maps support three generic information seeking tasks: *data analysis*, *querying* and *exploration*.

Data Analysis

A user may want to analyse a data set in order to get a better insight in its characteristics, to get a global understanding of the collection and to discover unexpected patterns.

The *static version* of the Cluster Map (i.e. the image itself) already contains

information that can be used for such purposes. The classes and their hierarchy provide an understanding of the domain of the data set. The way instances are classified results in class characteristics such as incompleteness and overlaps, showing class relations at the instance level. The cardinality of classes and clusters supports a quantitative analysis. Interpreting this information, one can already derive some domain specific observations.

The *strategy* used to obtain a static Cluster Map plays an important role. Different strategies support different analysis scenarios, as shown in (Fluit et al., 2002):

1. *Analysis within a single domain* In this case a data set is visualised from one or more perspectives, giving insight in the collection. One of our case studies in the above mentioned paper investigated characteristics of a set of job offers by visualising it according to perspectives such as region and relevant economic sector.
2. *Comparison of different data-sets* Data sets can be compared by visualising them using the same ontology. In (Fluit et al., 2002) we have compared two banks by imposing the same ontology on their web sites and analysing the two visualisations.
3. *Monitoring* Analysing a data set at different points in time provides insight in the way it evolves. For example one could monitor the site of a bank over time and see how its activities evolve (expand/interconnect/...).

Viewers that incorporate the Cluster Map (e.g. the DOPE browser or Xarop) turn analysis into an interactive and explorative process. The overview, zoom and filter facilities offered by the user interface qualitatively enhance the analysis process.

Querying

The goal of a query task is to find a narrow set of items in a large collection that satisfy a well-understood information need (Marchionini, 1995). We will show the benefits of using the Cluster Map viewer in the four stages of the search task (Shneiderman, 1998).

Query formulation – the step of expressing an information need through a query is a difficult task. Users encounter difficulties when having to provide terms that best describe their information needs (vocabulary problem). Furthermore, combining these terms in simple logical expressions using “AND” and “OR” is even more complicated. See (Shneiderman, 1996) for a demonstration of this problem.

In both the DOPE browser and the Xarop client the classes that describe the domain of a data set are explicitly shown, making the vocabulary choice much easier. There is no need to specify Boolean expressions as they are already visible on the map.

Initiation of action – Classes selected in the left panel become terms of a query. The search is launched at a mouse-click.

Review of results – the results of the query are graphically presented to the user. For n selected classes the following is shown:

1. the union of the classes (disjunction of *all* query terms)
2. all intersections of the selected classes (conjunction of *some* query terms)

3. (as a particular case of 2), the intersection of all classes (conjunction of *all* query terms) – if it exists.

Note that the results of simple Boolean expressions are intuitively shown in the map. If the user wants a disjunction of the terms he will analyse all the presented objects. As an added value he will see how the corresponding classes overlap. A more interesting (and probably more frequent) case is when users want the conjunction of the terms. In that scenario, two extreme situations can happen:

1. the result set is too large (under-specification)
2. the result set is empty (over-specification)

If the result set is empty, the user can at least find objects that partially satisfy the query. The left side of Figure 3.5 illustrates a case when the result set to a four term query is empty: there are no holiday destinations in the analysed data set that are located in *Loire*, have 2 rooms that accommodate 4 persons and provide a 3 star quality. The degree of relevance of certain clusters is suggested by their colour: the more relevant the darker the shade of the colour. It is evident from the visualisation that there are two destinations potentially interesting if the customer dropped one requirement (quality of accommodation or location). This is a form of *query relaxation*.

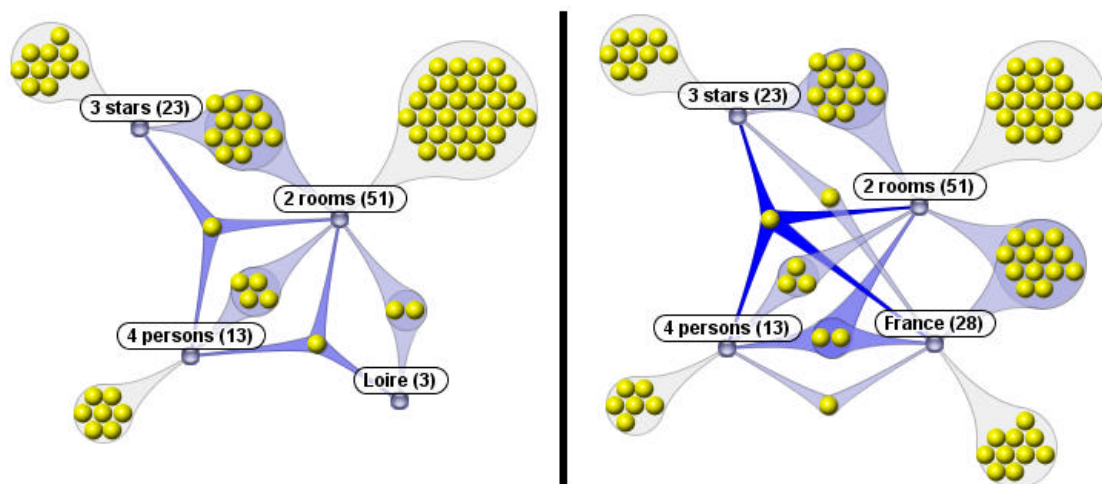


Figure 3.5 Query refinement.

Refinement – according to the conclusions of the result interpretation, the user can narrow or broaden the scope of his search by refining the query.

If the result set is too large, the user can replace some classes with more specific subclasses. Imagine that in the travel agent scenario the customer would like to go to *France* and that the system returns a huge cluster that satisfies all the criteria. Instead of looking at each returned object, the customer can *refine* his query to use a more specific class, for example *Loire*. This would narrow the scope of the query and return a smaller set of options.

At the other extreme, in case of an empty set, some classes can be replaced by their superclass. For example the search for a vacation in *Loire* was empty for the given settings, however one destination was found in *France*, as shown at the right side of Figure 3.5. Note that both narrowing and broadening the scope of the query are

possible due to the ontological nature of the domain description. The viewer facilitates choosing more specific or more general classes.

Exploration

We define the exploration task as the process of finding information that is loosely relevant to the user's current activities.

It differs from the querying task in that no specific question needs to be answered. Instead, the user wants to know about relevant information at a more global level, e.g. to see what information is available. Exploration differs from general analysis in that the issue is not to oversee the entire data set in a holistic way but only inspect those parts relevant to the user's current activities.

User studies have shown that exploration support is one of the biggest achievements of the Cluster Map visualization. For example, the informal DOPE usability testing (Stuckenschmidt et al., 2004) suggested a strong applicability for scientific literature surveys for research and educational purposes.

Exploration is strongly related to serendipity, usually defined as the making of fortunate discoveries by accident. Whereas serendipity is a welcome quality of every interface and during any task, we see exploration as a task where the user purposefully browses an information space to find such information.

Serendipitous and explorative interfaces can be realized in many ways. Of course any system that shows documents based on a user query is to some extent serendipitous as it may show a useful document that the user did not know about or forgot about. However, our applications go further than this. When displaying several populated classes in a Cluster Map, users often encounter unsuspected clusterings of the document. For example, overlaps may exist between classes that the user expected to be disjoint, or there may be "missing" overlaps, e.g. when a user expected a certain class to completely include all instances of another class and there are several instances that fall out of this subsumption.

Our AutoFocus application goes even one step further and suggests new search terms when the user clicks on a class or cluster. Using a statistical algorithm the application tries to find informative terms that occur in the selected set of documents and presents them to the user. The benefit of this is two-fold:

1. The user quickly learns about the rough contents of the document set without having to browse through a long result list and inspect every single document. This helps him building an abstract view of the information.
2. The user is guided in his exploration by being offered search terms that help him to drill down in this document set.

Explorative interfaces turn out to be an effective instrument in battling the information overload that knowledge workers have to face every day, because it diminishes the chance of missing valuable information. With such interfaces they are better informed about the information that is at their disposal.

3.5 Future Work

Future developments mainly concern the visualization itself and AutoFocus, as currently these are still continuously being refined. We will mainly focus on those improvements that are relevant to information visualization and Semantic Web research.

First, we want to investigate the use of more expressive ontological structures. RDF Schema and related standards like OWL allow for much more expressive models than the hierarchical, populated classification trees visualized by the Cluster Map. High on our priority list are arbitrary binary relations between documents, since these occur frequently in the data sets we encounter. Examples are hypertext links between webpages and aggregated information objects such as emails containing attachments and zip files. Such improvements impact the query formulation user interface as well as the display of results in the Cluster Map.

In several cases we have observed that the inclusion of a time dimension in the display of information may be useful. An example is the publication date of a document. Visualizing how the population of a set of classes changes over time can greatly improve insight into how the entire document set evolves over time.

Finally, we envision several AutoFocus extensions that improve its general usefulness. One example is the ability to let the user define its own taxonomy, with each class in the taxonomy being defined by a filter on the document characteristics or, when sufficient, by simply listing all the relevant documents. Similar functionality can already be found in certain mail readers, where they define virtual mail folders, and MP3 collection managers, where they define dynamic playlists. In our context, this concept can be extended to arbitrary document sets. Additionally, since AutoFocus and Metadata Server share the same Sesame-based back-end, this can provide a basis towards a P2P version of AutoFocus. Only the typical P2P communication layer has to be provided, a query answering mechanism is already in place. These two extensions may reinforce each other, as the possibility to share (populated) taxonomies may make their creation and maintenance well worth the effort.

3.6 Summary

This chapter has demonstrated an elegant way to visually represent ontological data. We have described how the Cluster Map visualization can use ontologies to create expressive information visualizations, with the attractive property that classes and objects that are semantically related are also spatially close in the visualization.

Another key aspect of the visualization is that it focuses on visualizing instances rather than ontological models, thereby making it very useful for information retrieval purposes.

A number of applications developed in the last years have been described that prominently incorporate the Cluster Map visualization. Based on these descriptions, we could distinguish a number of generic information retrieval tasks that are well supported by the visualization.

These applications prove the usability and usefulness of the Cluster Map in real life scenarios. Furthermore, these applications show the applicability of the visualization in Semantic Web-based environments, where light-weight ontologies are playing a crucial role in organizing and accessing heterogenous and decentralized information sources.

3.7 References

Bonifacio M, Bouquet P, Traverso P (2002) Enabling distributed knowledge management: Managerial and technological implications. *Novatica and Informatik/Informatique III* (2002).

Broekstra J, Van Harmelen F, Kampman A (2002) Sesame: a generic architecture for storing and querying RDF and RDF Schema. *Proceedings of the First International Semantic Web Conference (ISWC'02)*.

Eades P (1984) A heuristic for graph drawing. *Congressus Numerantium* 42: 149-160.

Fluit C, Sabou M, Van Harmelen F (2002) *Ontology-based information visualisation, Visualising the Semantic Web*, Springer-Verlag.

Fluit C, Sabou M, Van Harmelen F (2003) Supporting user tasks through visualisation of light-weight ontologies, *Ontology Handbook*, Springer-Verlag.

Hemmje M, Kunkel C, Willett A (1994) Lyberworld – a visualization user interface supporting fulltext retrieval. *Proceedings of the 17th Annual International ACM/SIGIR Conference*, pp 249–259.

Korfhage R (1991) To see or not to see – is that the query? *Proceedings of the 14th Annual International ACM/SIGIR Conference*, pp 134–141.

Marchionini G (1995) *Information seeking in electronic environments*. Cambridge University Press, Cambridge, UK.

Shneiderman B (1996) The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings of the 1996 IEEE symposium on visual languages (VL '96)*, pp. 336-343.

Shneiderman B (1998) *Designing the user interface*. Addison-Wesley, Menlo Park, pp. 509-551.

Stuckenschmidt H, De Waard A, Bhogal R, Fluit C, Kampman A, Van Buel J, Van Mulligen E, Broekstra J, Crowlesmith I, Van Harmelen F, Scerri T (2004) A topic-based browser for large online resources. *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW'04)*.