Ontology-based Information Visualisation

Frank van Harmelen

AIdministrator Nederland BV, Amersfoort,
http://www.aidministrator.nl

Vrije Universiteit Amsterdam
Frank.van.Harmelen@cs.vu.nl

Jeen Broekstra, Christiaan Fluit, Herko ter Horst
Arjohn Kampman, Jos van der Meer, Marta Sabou

AIdministrator Nederland BV, Amersfoort,
http://www.aidministrator.nl

Abstract

The main contribution of this paper is to show how visual representations of information can be based on ontological classifications of that information. We first discuss the central rôle of ontologies on the Semantic Web. We subsequently outline our general approach to the construction of ontology-based visualisations of data. This is followed by a set of examples of ontology-based visualisations which all differ in interesting respects. The paper concludes with a brief discussion of related work.

1 Introduction

1.1 The Context

We are on the brink of a new generation of World Wide Web (WWW) which, in his recent book Weaving the Web, Tim Berners-Lee calls the Semantic Web [1]. Unlike the existing WWW, where data is primarily intended for human consumption, the Semantic Web will provide data that is also machine processable. This will enable a wide range of intelligent services such as information brokers, search agents, information filters etc., a process that Berners-Lee describes as "Bringing the Web to its full potential". The importance of research in this area is indicated by the recently announced DAML initiative[1] in the USA, under whose aegis projects aimed at developing the Semantic Web will receive DARPA funding totalling $70 million [6]. In Europe, the newly launched Semantic Web activity of the European Union's IST programme illustrates the same importance[2]. Finally, the World Wide Web consortium has recently announced its Semantic Web activity to oversee and coordinate the development of standard technologies in this area[3].

1.2 Ontologies

The development of ontologies is seen as central in all of these efforts (they are mentioned as key technologies in all of the aforementioned Web-pages). Ontologies are meta-data, providing a controlled vocabulary of terms, each with an explicitly defined and machine processable semantics. By defining shared and common domain theories, ontologies help both people and machines to communicate more effectively. They will therefore have a crucial rôle in enabling content-based access, interoperability and communication across the Web. Examples of the use of ontologies to support content-based access and interoperability can already be seen in e.g., the American SHOE project[4] [5], in which HTML is being extended with ontology based semantic markup codes, and the European IST-project On-To-Knowledge[5] [4], in which ontologies are being used to facilitate access to large intranets.

Ontologies on the Semantic Web will be crucial to the development of Web applications such as e-commerce, providing users with much more sophisticated searching and browsing capabilities as well as support from intelligent agents such as shopbots (shopping "robots" that access vendor Web sites, compare prices etc.). Examples of the use of

[1] http://www.daml.org

[2] http://www.cordis.lu/ist/ka3/iaf/iii41obj.htm

[3] http://www.w3.org/2001/sw

[4] http://www.cs.umd.edu/projects/plus/SHOE

[5] http://www.ontoknowledge.org

0-7695-1195-3/01 $10.00 © 2001 IEEE

ontologies and taxonomies to support searching and browsing can already be seen at e.g., Yahoo Shopping[6] and Amazon.com[7]

Ontologies can be as simple as a keyword-hierarchy (sometimes known as "lightweight ontologies") or they can be complex concept-hierarchies with properties, value-restrictions, and axiomatised relations between the concepts (for instance as expressible in the language DAML+OIL[8], developed in a joint EU/US collaboration).

For the purposes of this paper however, we will restrict ontologies to simple concept-hierachies: each ontology is a simple subclass-hierarchy, where subclasses can be incomplete and overlapping. Thus, the only guarantee is that all subclasses are a subset of their superclass. Taken together, the subclasses may not completely cover their superclass; also, the subclasses may overlap, i.e. there may be some instances that belong to more than one of the subclasses. In fact, it will turn out that much of the power of our visualisation techniques derives from the potential overlap of the subclasses.

1.3 Uses of ontology-based visualisation

In this paper, we illustrate how ontologies can be used as the basis for effectively visualising the contents of Web-resources for the purposes of navigating, searching, analysing and browsing such resources. We will first give a general outline of our approach to visualisation. This will be followed by a sequence of examples of this approach applied to various kinds of Web resources. Each of these examples uses the visualisation technology to support a different task:

- to do *data analysis*, either by organising a data-set under a single ontology, or by applying multiple ontologies to the same data-set (section 3),
- to do *data comparison* between related information resources by visualising them under the same ontology (section 4), and
- to do *query-relaxation*, i.e. to use ontologies to find near-hits to queries for which no exact answers exist (section 5).

2 Our approach

2.1 From information to visual representation

Ontologies can be used for many different purposes. The literature on Knowledge Representation contains research on the use of ontologies for data-interchange, for data-integration, for data-querying, for data-verification, etc. In this paper, we are concerned with the use of ontologies for data-*visualisation*.

In general, visualisation of information can be seen as a two-step process, as depicted in figure 1. In a first step, the information is transformed into some intermediate semantic structure. This structure organises the raw information into a meaningful structure. In this step, a variety of operations are typically applied to obtain this structure: selecting, transforming, filtering, classifying, merging, etc. In a second step, this semantic structure is then used as the basis for a visual representation. The first step can be characterised as deciding *what* to visualise, while the second step is concerned with *how* to visualise the result of the first step.

The figure depicts a very general process. Even processes such as constructing a lecture or designing a website are captured by this diagram. In each case, one would make different choices for the representation of the semantic structure and for the visual presentation. Obviously, the first choice limits the options for the second choice.

In the case of this paper, we apply this process to generating visualisations of Web resources. In that specific case, the raw data consists of traditional web-pages. These Webpages are classified into a predefined ontology which plays the role of the "semantic structure". The resulting classification of Web pages is the basis for generating the visual presentation.

We have written elsewhere on how we proceed to semi-automatically classify web-pages in different classes of a predefined ontology [9, 8], i.e. the first part of the above diagram. In this paper, we only discuss the second part of the diagram, namely which visual representations we generate from a dataset that is classified according to a given ontology.

2.2 Visualisation of ontological structures

Before we discuss the various visualisations in more detail, we briefly explain how to read the diagrams, and we will give a little background on how they have been generated (although the details of the algorithms are non-trivial, and beyond the scope of this paper). We use figure 2 as an example to illustrate our explanation.

Each text-label in the diagram corresponds to a class in the ontology (e.g. "secretarial"). Each circle corresponds to a data-element (an instance) that has been classified in one or more ontological classes. A line between an instance (i.e. a circle) and a class (i.e. a text-label) indicates that the instance is a *member of the class*; a line between two classes indicates that one is a *subclass* of the other (for instance the class of "secretarial" jobs is a subset of the most general class "vacancies").

[6] http://shopping.yahoo.com
[7] http://www.amazon.com
[8] http://www.daml.org/2001/03/daml+oil-index.html

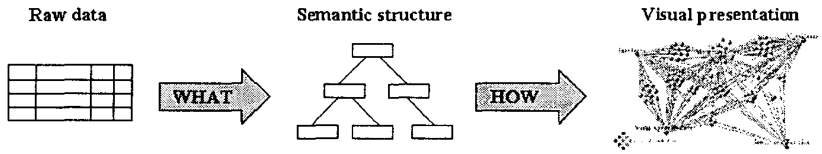

Figure 1. General approach to generating visual presentations

Notice that an instance may belong to multiple classes, because we allow overlapping classes in our ontologies. The organisation of the diagram is computed fully automatically by a variant of a "spring-embedder"-algorithm (originally proposed in [3]): On the one hand, all nodes (classes and instances) repell each other (and therefore try to fill the plane), while all edges (between two classes or between an instance and class) produce an attracting force (i.e. the edges work as "springs").

The central aim of the layout-algorithm is that objects which are *"semantically close"* in the ontology should also be graphically close in the diagram. To this end, we use the following simple but effective definition of semantic closeness[9]:

- Two classes are semantically close when they share many instances;
- Two instances are semantically close when they belong to the same classes.

For example, in figure 2 the classes "finance" and "commerce" share a number of instances, and are therefore semantically close.

The forces in the spring-embedder algorithm have been chosen such that the resulting layout of the objects captures their "semantic closeness". The layout-algorithm computes a stable configuration of the objects under these forces.

We discuss different examples of such visual representations in the next few sections.

3 One dataset, multiple ontologies

3.1 Jobs organised by economic sector

The first example of an ontology-based visualisation is shown in figure 2. This figure was generated from the contents of the web-site of a major Dutch job-agency. Each circle represents a job offering by the agency (represented as a web-page on the agency's web-site). The ontology classifies the different jobs by economic sector: agricultural, technical, financial, etc.

[9]We only give the intuitive definition, but this can of course be formalised. This formal definition is omitted from this paper

This visualisation can be effectively used for *data-analysis*. A few observations follow immediately from this visualisation.

First of all, it is immediately clear which classes in the ontology are larger than others: many jobs are available in the technical sector, while very few are available in sport, for example.

Secondly, and more importantly, the visualisation also makes it immediately clear which jobs are classified in multiple ontological classes (i.e. in multiple economic sectors). For example, there are quite a few jobs from the administrative sector which are also classified as either management, marketing or secretarial. Such multiple classifications make it immediately clear to users which jobs are or are not relevant to their interests.

Similarly, it is also directly clear which ontological classes (i.e. which economic sectors) do not overlap at all in the given dataset. Classes which have a large overlap will tend to be located closer together in the diagram. As a result, we can see immediately (and perhaps surprisingly), that the classes sport and recreation do not contain any joint elements. If they would have had joint elements, these classes would not have been placed at opposite ends of the diagram.

Finally, it is worth emphasising the compactness of this representation. Figure 2 displays 484 jobs in a single figure. In a traditional text-based presentation, that would have corresponded with 24 pages of 20 items per page! Just as the traditional text-based representations are used as the basis for navigating by including clickable links in the text, images such as figure 2 can be used for navigating by producing the figures as clickable image maps.

3.2 Jobs organised by region

Figure 3 shows the same dataset as figure 2 , but this time organised in a different ontology. The ontology of figure 3 is based on geographical region (in fact, it simply lists all 12 provinces of The Netherlands plus jobs abroad). The figure shows a radically different organisation of the same dataset, according to this different ontology. Again, it is clear that there are many more jobs on offer in some provinces than in others. It is also clear which jobs are located in multiple provinces (for example work on different locations). It

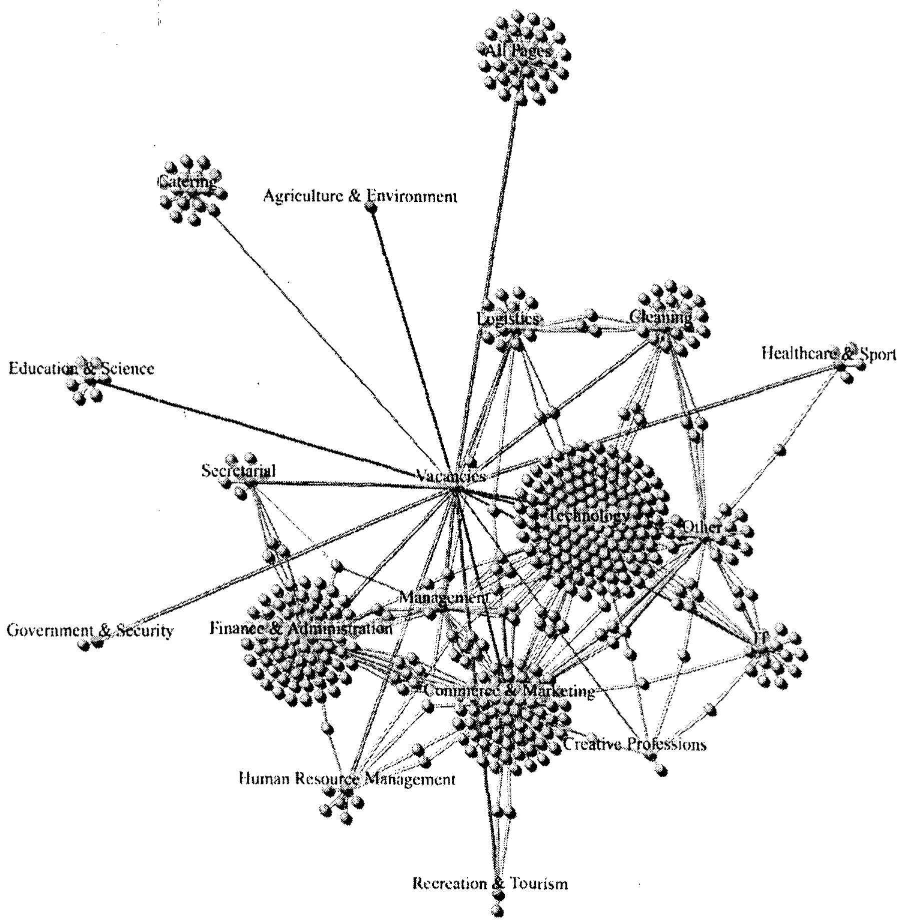

Figure 2. Jobs organised by economic sector

is amusing to note that the layout algorithm has more or less reconstructed the geography of the Netherlands. This is of course because provinces that are geographically clause in reality are likely to share jobs, and are therefore likely to appear next to each other in the visualisation.

3.3 Combining multiple ontologies

Figure 4 shows again the map from figure 2 (jobs organised by economic sector), but this time shows the results of a query (darker circles are those instances satisfying the query). The crucial point here is that, while the layout is based on the economic-sector ontology, the query was based on the geography of the jobs: "give me all jobs in either Utrecht or Noord-Holland". Thus, the query was phrased using one ontology, and the results were displayed in a map based on another ontology. Rather than simply getting a list of the few dozen jobs that satisfied the query, the user now sees the resulting jobs organised in a meaningful way, namely by economic sector. (As a minor point: dark circles are those jobs which are in both Utrecht and North-Holland, while the lighter circles are in either Utrecht or North-Holland, but not in both).

This example shows that ontology-based visualisations can be effectively used to *display query-results* over information sources.

4 One ontology, multiple datasets

Whereas in the previous section, we used two different ontologies to organise the same data-set, it is of course also possible to display multiple datasets using a single ontology. An example of this is shown in figure 5. This shows the top-level pages on the Web-site of two major Dutch banks, both organised with the same ontology. Such a visualisation can

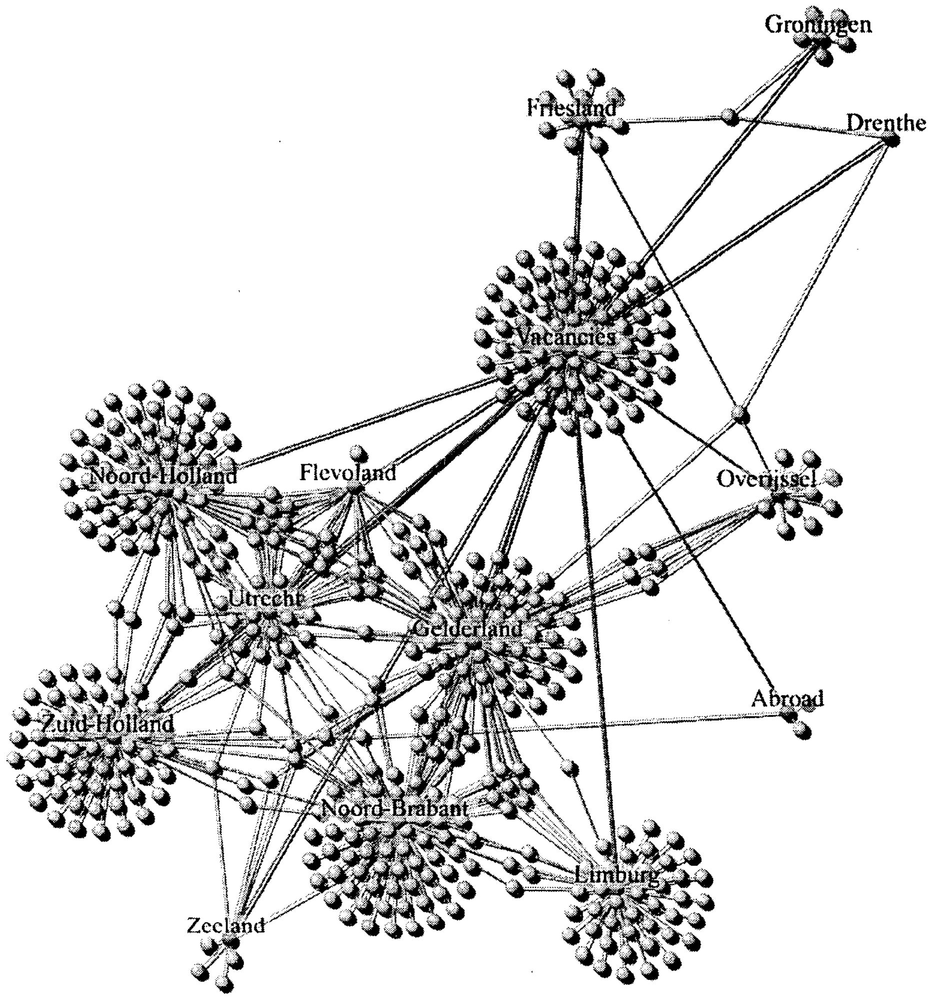

Figure 3. Jobs organised by region

be used to *analyse the differences* between the two data-sets. For example, notice the location of the pages on "beleggen" (investing in shares) in both images. In the image on the right, these pages are tightly integrated with the pages from other clusters, showing that the banks offerings on on share-investment products mention many of their other products and services. In the image on the left, share-investment pages are rather isolated from the other clusters.

Incidentally, both images contain a nice example of the usefulness of overlapping classes: both banks offer rather many pages on the Euro. Due to the spring-layout algorithm, it is immediately clear which of these web-pages are concerned with the Euro on holidays ("vakantie"), and which of these web-pages are dealing with other matters such as investment ("beleggen") or businesses ("ondernemen").

5 Visualising query relaxation

Figure 6 shows an altogether different use of ontology-based visualisation. It is an example taken from an e-commerce site selling computer printers. The ontology that is driving this website describes various properties of printers such as speed (in pages-per-minute, ppm), resolution (in dots-per-inch, dpi), price (in Deutsch Mark DM), whether the printers are ink-jet or laser, color or black-and-white, and the make of the printer (HP, Compaq, etc). Customer to the site areasked to specify their preferences for a printer in terms of these properties.

A major problem with e-commerce is that often no single

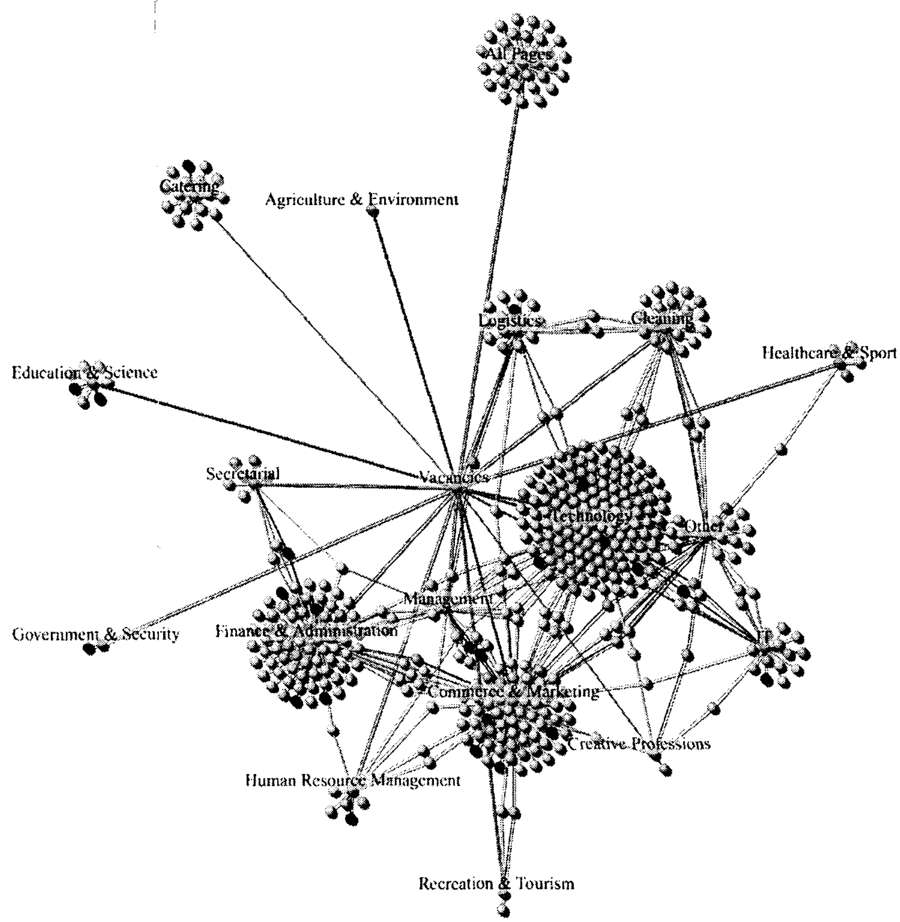

Figure 4. Combining two ontologies

product exists which exactly satisfies the wishes of the customer (high-resolution, high-speed, high-quality, low-price printers simply do not exist☺). This will return an empty answer in response to the customer's query. In the absence of a qualified sales-person, there is a serious risk that the customer will abandon the site at this point, thinking that no interesting products are available. What a qualified sales-person in a shop would do at this point is to analyse the customers requirements, and to explain to the customer that although no printer completely satisfies the requirements, there are printers which satisfy almost all of the requirements, and which are therefore still interesting options for the customer.

The visualisation in figure 6 does exactly that. It shows that no product exists that matches the given query (an ink-jet color printer by Compaq less then DM 700 and faster then 12 ppm) but that a matching product does exist if we drop the requirement that it must be from Compaq. Even better, we also see that 4 printers exist that match all of the criteria except that they will be slower then 12ppm. Lighter shades indicate products that match fewer of the customers requirements.

What is happening here is that the ontology is being used for *query-relaxation*: when no perfect answer exists, the semantic structure of the ontology enables the system to return answers that are close the required answer. Furthermore, the system is also able to show in which respect these answers deviate from the requirements.

Notice that in figure 6 the query and its outcome are actually determining the layout of the diagram. This is different from figure 4, where the the query is only used to highlight instances in a layout which is determined by another ontology (namely economic sector).

The example from figure 6 is based on a flat set of re-

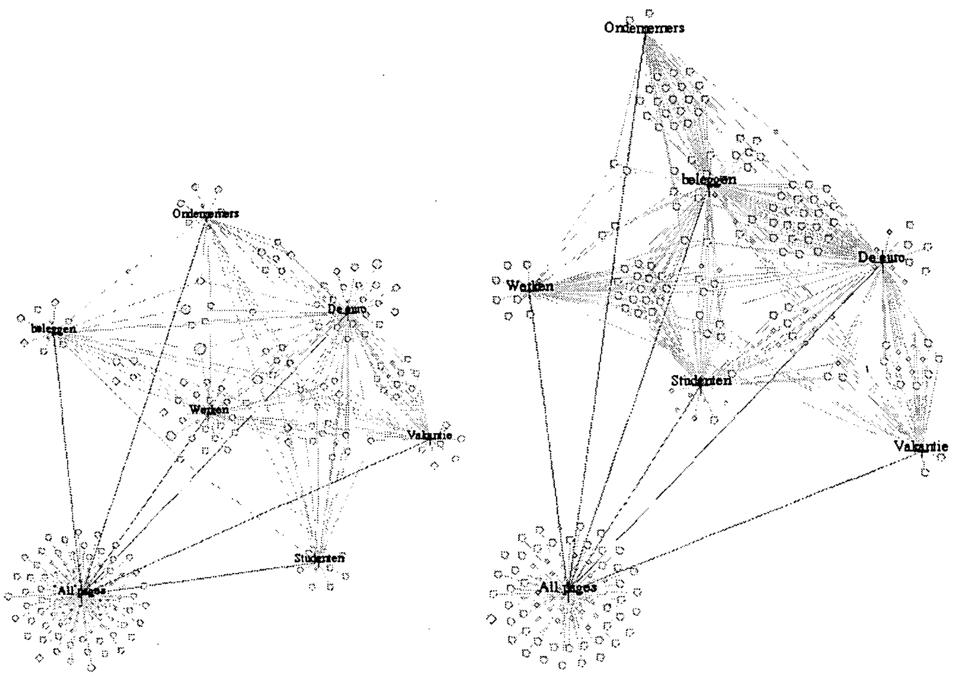

Figure 5. Two datasets with the same ontology

quirements. The possibilities of this visualisation become more interesting when the requirements are organised in a proper hierarchy, for example by putting "speed" and "resolution" together in a superclass "printer quality", while "ink" and "color" would go together in a superclass "printer type". It is then possible to do query relaxation based on the hierarchical structure of the ontology. This will allows users to choose if they would rather choose a different printer type, or compromise on printer quality instead. In fact, the query-relaxation shown in figure 6 could also have been done in a simple relational database-schema. It is only with such hierarchical structures that the additional value of ontologies become apparent.

6 Related work

Quite a few surveys on visualisation-techniques for web-resources are available (for example [7]). The Atlas of Cyberspaces[10] provides one of the best on-line overviews of visualisations. This paper is not the place to give an exhaustive overview and analysis of the plethora of different approaches to visualising web-resources, and we will only discuss two that are most closely related to our work. These are The Brain and the Hyperbolic Tree by Inxight, because these visualisations have the same nature as the technology described in this paper.

6.1 The Brain: only local neighbourhood, not global overview

The main difference with The Brain[11] is that The Brain only shows the local neighbourhood in a large network of relations. The visualisation changes while the user navigates in the large network. The advantage is that this allows in principle to display an arbitrarily large network. Experience shows that our technologies can display hundreds of nodes before screen real-estate overflows. However, the disadvantage of The Brain's approach is that the constantly changing nature of the display makes it hard for the user to orient themselves, and to obtain a sense of global location.

6.2 Hyberbolic Trees: only trees, not arbitrary hierarchies

Another well-known visualisation technique is the Hyperbolic Tree Server[12] by Inxight (exploiting Xerox technology). It emphasises focussing techniques to focus on

[10] http://www.cybergeography.org/atlas/web_sites.html

[11] http://www.thebrain.com

[12] http://www.inxight.com/products_wb/ht_server

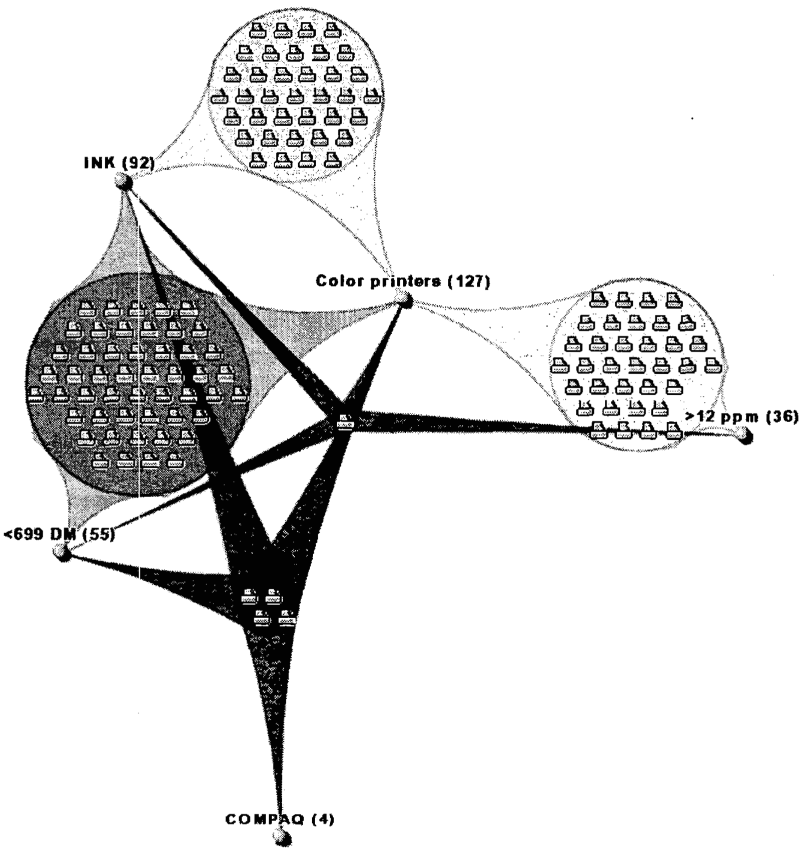

Figure 6. Displaying query relaxation

some parts of a large data-set at the cost of other parts of the data-set. This is closely related to a number of other visualisations (some of which directly rely on the Inxight technology), such as the fish-eye view of Mapuccino by IBM Haifa Research Lab[13], the display of ontologies in the OnToBroker project[14] [2] and the Astra Site Manager[15] among others. The main limitation of all of these is that they assume that the visualised data-structure is a tree (i.e. subclasses are non-overlapping and exhaustive). The main power of the approach illustrated in this paper is that it can deal with non-tree-like hierarchies, and that it actually exploits the overlap between different subtypes to organise the diagrams in a meaningful way.

6.3 The Brain and Hyperbolic Trees: general graphs, not ontologies

A final import difference is that the both the Brain and the Inxight system visualise a graph structure without any assumption on what this structure represents. As far as these systems are concerned, the graph might be a hypertext-structure, a class-hierarchy, a semantic network, or anything else. Our technique, on the other hand, is based on the knowledge that the hierarchy it visualises is an ontology, and this knowledge is exploited in the visualisation. For example, the forces which determine the spring-embedder algorithm are chosen with this in mind.

7 Conclusion

In this paper we have shown that ontologies are a very useful semantic structure as the basis for visual presentations of web-resources. Since ontologies are expected to

[13] http://www.alphaworks.ibm.com/tech/mapuccino
[14] http://ontobroker.semanticweb.org
[15] http://www-heva.mercuryinteractive.com

play a central role in the infrastructure of the Semantic Web, we expect that such ontology-based visualisations will become an important tool for navigation, searching and query-answering on the Semantic Web.

We have briefly described how ontologies can be used in fully automatic spring-embedder layout algorithms, with the attractive result that classes and instances which are semantically close in the ontology will also appear spatially close in the visualisation. We have shown a number of examples of how medium-sized data-sets (up to a few hundred items) can be displayed in a meaningful way using this technology.

Acknowledgement

Figure 6 is based on a domain model constructed by Semantic Edge in Berlin.

References

[1] T. Berners-Lee. *Weaving the Web.* Harper, San Francisco, 1999.

[2] S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In R. M. et al., editor, *Semantic Issues in Multimedia Systems, Proceedings of DS-8*, pages 351–369, Boston, 1999. Kluwer Academic Publisher.

[3] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[4] D. Fensel, F. van Harmelen, M. Klein, H. Akkermans, J. Broekstra, C. Fluit, J. van der Meer, H.-P. Schnurr, R. Studer, J. Hughes, U. Krohn, J. Davies, R. Engels, B. Bremdal, F. Ygge, T. Lau, B. Novotny, U. Reimer, and I. Horrocks. On-to-knowledge: Ontology-based tools for knowledge management. In *eBusiness and eWork*, Madrid, October 2000.

[5] J. Heflin and J. Hendler. Dynamic ontologies on the web. In *Proceedings of 17th National Conference on Artificial Intelligence (AAAI-2000)*, 2000.

[6] J.Hendler and D. McGuinness. The darpa agent markup language. *IEEE Intelligent Systems*, 15(6):72–73, Nov/Dec 2000.

[7] D. Stenmark. To search is great, to find is greater: a study of visualisation tools for the web. http://w3.informatik.gu.se/ dixi/publ/mdi.htm.

[8] F. van Harmelen, A. Kampman, H. Stuckenschmidt, and T. Vogele. Knowledge-based meta-data validation: Analyzing a web-based information system. In K. Greve, editor, *Fourtheenth International Symposium Informatics for Environmental Protection*. German Computer Society, 2000.

[9] F. van Harmelen and J. van der Meer. Webmaster: Knowledge-based verification of web-pages. In I. Imam, Y. Kodratoff, and M. Ali, editors, *Twelfth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE'99*, number 1611 in Lecture Notes in Artificial Intelligence, pages 256–265. Springer Verlag, 1999.