An ontological approach to information visualization.

Gilson, Owen Timothy

An Ontological Approach to Information Visualization

Owen Timothy Gilson BSc.(Soton)

Submitted to the University of Wales in fulfilment of
the requirements for the Degree of Doctor of Philosophy

Department of Computer Science
Swansea University, UK

2008

Summary

Visualization is one of the indispensable means for addressing the rapid explosion of data and information. Although a large collection of visualization techniques have been developed over the past three decades, the majority of ordinary users have little knowledge about these techniques. Despite there being many interactive visualization tools available in the public domain or commercially, producing visualizations remains a skilled and time-consuming task. One approach for cost-effective dissemination of visualization techniques is to use captured expert knowledge for helping ordinary users generate visualizations automatically. In this work, we propose to use captured knowledge in *ontologies* to reduce the parameter space, providing a more effective automated solution to the dissemination of visualization techniques to ordinary users. As an example, we consider the visualization of music chart data and football statistics on the web, and aim to generate visualizations automatically from the data. The work has three main contributions:

Visualisation as Mapping. We consider the visualization process as a mapping task and assess this approach from both a tree-based and graph-based perspective. We discuss techniques for automatic mapping and present a general approach for Information Perceptualisation through mapping which we call Information Realisation.

VizThis: Tree-centric Mapping. We have built a tree-based mapping toolkit which provides a pragmatic solution for visualising any XML-based source data using either SVG or X3D (or potentially any other XML-based target format). The toolkit has data cleansing and data analysis features. It also allows automatic mapping through a type-constrained system (AutoMap). If the user wishes to alter mappings, the system gives the users warnings about specific problem areas so that they can be immediately corrected.

SemViz: Graph-centric Mapping. We present an ontology-based pipeline to automatically map tabular data to geometrical data, and to select appropriate visualization tools, styles and parameters. The pipeline is based on three ontologies: a Domain Ontology (DO) captures the knowledge about the subject domain being visualized; a Visual Representation Ontology (VRO) captures the specific representational capabilities of different visualization techniques (e.g., Tree Map); and a Semantic Bridge Ontology (SBO) captures specific expert-knowledge about valuable mappings between domain and representation concepts. In this way, we have an ontology mapping algorithm which can dynamically score and rank potential visualizations. We also present the results of a user study to assess the validity and effectiveness of the SemViz approach.

Selected parts of this thesis have been presented at: I-KNOW '06 - 6th International Conference on Knowledge Management, Graz, Austria; The 3rd International Semantic Web User Interaction Workshop at ISWC '06, Athens, Georgia, USA; and at EuroVis 2008, Eindhoven, Netherlands. At EuroVis 2008 the paper, "From Web Data to Visualization via Ontology Mapping" won the Best Paper Award.

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....22 May 2004........

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ... . (candidate)

Date ....22 May 2004........

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ... . (candidate)

Date ....22 May 2004.........

Acknowledgements

Firstly I would like thank my two supervisors Dr. Phil Grant and Prof. Min Chen. They have provided me with a great deal of guidance and mentoring necessary for me to complete this thesis. Their deep knowledge of both Computer Science and the academic world have helped me greatly in adjusting from a mainly industrial past experience.

I value the support and kindness of the friends who I have made over the years. From my old school friends from Swansea, university friends from Southampton, and past colleagues who I have worked with at various companies.

The computer science post-grad lab in Swansea has provided a great support network and I thank past and present members for their chats and words of wisdom. The staff at Swansea have also been very helpful and supportive both from a technical and more personal perspective.

Part of this research work was completed during a 6 month research visit to Instituto Superior de Engenharia Porto (ISEP) in Portugal. I am indebted to Nuno Silva for the time and assistance he has dedicated to these studies. I also wish to thank the GECAD research group and members for warmly welcoming me to their team.

I am very grateful to the various funding bodies which have supported me financially with grants from Swansea University and also the European Union through the Welsh Assembly Government.

Finally, I must mention the support and guidance my family has provided for me during this journey. Thanks to Mum, Dad, Rhian and also Dianne and co for your kindness. *Mae eich cefnogaeth a'ch cred wedi bod yn ysbrydoliaeth.*

Contents

Chapter 1

Introduction

Contents

1.1 Motivation

The area of information visualization has been researched for over three decades. In the early years there was a focus on creating novel visualization techniques in order to demonstrate sophisticated ways of gaining insight into information. Many valuable techniques have resulted from these efforts. However, this "gold rush" of activity has reduced to a slower pace over the last decade as the number of truly new visualization techniques being developed has diminished. For example, when the TreeMap technique was first presented to the academic community, this was a truly new technique. However, most recent work on developing visualization techniques has only been incremental progress (for example Cushioned TreeMaps), rather than new visualization paradigms. As such, the visualization community has been looking towards an alternative challenge.

This challenge was partly addressed by the need to integrate these various visualization techniques into general purpose visualization toolkits. These toolkits allow users to take a data source and develop an effective visualization in a relatively short amount of time. As these visualization toolkits have become more advanced, the amount of programming required has also reduced so that now a visualization can be produced using a modern GUI system [Ahl96] [Bau06] [VWvH^{+}07] [MHS07].

However, despite the range of visualization techniques and the sophisticated toolkits available to produce them, information visualization is still a domain which is restricted to expert users.

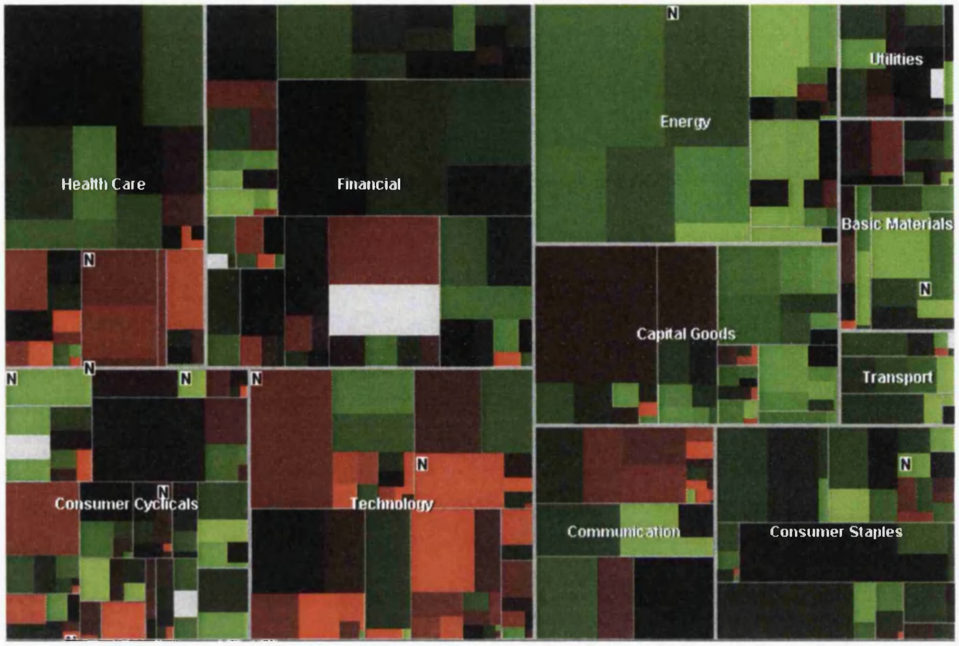


Figure 1.1: Smart Money's MarketMap [SW01] showing the market activity for stocks in all major sectors.

Expert visualization users are defined as either:

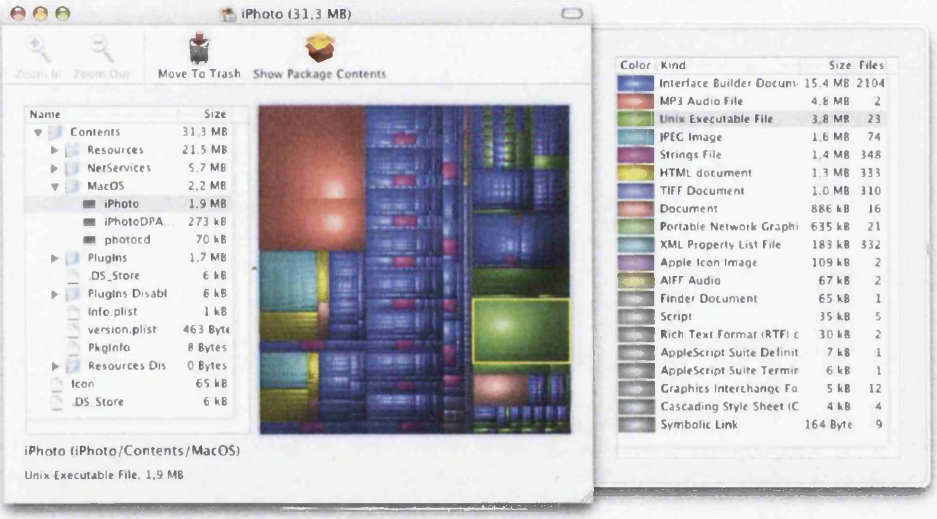
1. People who work in the field of Visualization and as such have a deep knowledge of the techniques, processes and tools. For example, an academic in the field of Volume Visualization.
2. People who work in a field with a heavy reliance on the analysis of datasets. The necessity to analyse their data is of such importance as to warrant the person learning about the techniques, processes and tools of visualization. For example, a oceanographer whose work is focussed on analysing the flows and patterns of the earth's oceans.

Additionally, expert visualization users have specific goals they wish to achieve when creating a visualization - their focus is on analysis and presentation. Also, there is little constraint on the expense of creating the visualization (in terms of time and equipment).

Non-expert visualization users are defined as:

- People whose main focus in life is not on a particular dataset and certainly not in the area of visualization. They have a casual interest in a dataset with no specific goal except to gain more insight into the data over and above what is presented in the original format (often textually) - their focus is on exploration. The availability of time and resources to create a visualization is limited because there is no guaranteed benefit to be gained from the visualization. As such it is difficult to justify expending resources on the task of creating a visualization.

Unfortunately, it is not within the capabilities of non-expert users to create visualizations using the available toolkits. The user interface of the toolkits is specialised and uses visualization specific language and interaction techniques. This has limited the adoption of information visualization. As such, the public as a whole has largely been unable to benefit from these valuable visualization techniques. The resulting status quo is that the limit of most users' experience of visual presentation comes from the relatively simple tools which are provided by office applications such as Microsoft Excel [MHS07]. The expressive capabilities of the bar charts and pie graphs which can be produced by Excel are clearly no match for modern information visualization techniques such as Tree Maps and Parallel Coordinates. There are exceptions to this. For example, Smart Money's MarketMap applet (see figure 1.1) presents users with a TreeMap indicating activity on the stock markets. This applet gained popularity in the financial community, introducing these users to the concepts of TreeMaps. However, the MarketMap applet is domain specific and only supports one visualization technique. Similar domain-specific, single-technique visualization applications exist for other applications (e.g., Disk Inventory X for disk space usage - see figure 1.2). However, none of these applications has pushed information visualization into mass adoption, or significantly outside of the visualization community.



Name	Size
▼ Contents	31,3 MB
▶ Resources	21,5 MB
▶ NetServices	5,7 MB
▼ MacOS	2,2 MB
iPhoto	1,9 MB
iPhotoDPA...	273 kB
photocd	70 kB
▶ Plugins	1,7 MB
.DS_Store	6 kB
▶ Plugins Disabl	6 kB
Info.plist	1 kB
version.plist	463 Byte
PkgInfo	8 Bytes
▶ Resources Dis	0 Bytes
Icon	65 kB
.DS_Store	6 kB

iPhoto (iPhoto/Contents/MacOS)

Unix Executable File, 1,9 MB

Color	Kind	Size	Files
🟦	Interface Builder Docum	15.4 MB	2104
🟥	MP3 Audio File	4.8 MB	2
🟩	Unix Executable File	3.8 MB	23
🟦	JPEG Image	1.6 MB	74
🟪	Strings File	1.4 MB	348
🟨	HTML document	1.3 MB	333
🟦	TIFF Document	1.0 MB	310
🟥	Document	886 kB	16
🟩	Portable Network Graphi	635 kB	21
🟦	XML Property List File	183 kB	332
🟪	Apple Icon Image	109 kB	2
🟨	AIFF Audio	67 kB	2
⬛	Finder Document	65 kB	1
⬛	Script	35 kB	5
⬛	Rich Text Format (RTF) c	30 kB	2
⬛	AppleScript Suite Definit	7 kB	1
⬛	AppleScript Suite Termir	6 kB	1
⬛	Graphics Interchange Fo	5 kB	12
⬛	Cascading Style Sheet (C	4 kB	4
⬛	Symbolic Link	164 Byte	9

Figure 1.2: Disk Inventory X [Der08] showing a cushioned TreeMap visualization of the contents of a hard disk.

As the volume of information increases, with more data being presented on the world wide web, the need for

general purpose information visualization for non-experts

becomes more pressing. This is the primary motivation for the work in this thesis. Below we list other motivations which contribute to the first one.

The Semantic Web There has been much talk of the "chicken and egg" situation with the Semantic Web: users are unwilling to semantically enrich their data as there are very few applications which can exploit it; and organisations are unwilling to expend resources on developing semantically-enabled applications without there being a critical-mass of available semantically enriched information [Shi05] [msK06]. If it is possible to automatically create cognitively useful visualizations from semantically marked-up data, then there is a clear and motivating incentive for users to distribute data which includes semantics.

The explosion in information It is said that the world's total yearly production of print, film, optical, and magnetic content would require roughly 1.5 billion gigabytes of storage [LV03]. Clearly techniques (e.g. visualization) need to be developed which can help users to understand and exploit this information.

The "here and now" attitude Partly due to the amount of information available, and partly due to increased pressures on society, users are unwilling or unable to spend a large amount of time analysing data. Users have very focussed questions and expect quick answers. These needs can in part be met by greater access to visualization techniques.

Visually focussed society Young people are becoming more focussed on vision as their main sense. With so many distractions from television, video games and the many communication channels of the Internet, this heightened sense is becoming more developed and people expect systems which use vision as their main form of communication [Jen95]. Visualization techniques can be exploited as part of this trend.

Distributed and open knowledge As more information and media is distributed via the Internet, society is expecting this information and media to be free and open. This has had the side-effect of making it easier to share knowledge. For example the Wikipedia online encyclopaedia or shared databases such as IMDB. At the moment, this knowledge is mainly held in free text form (as conventional books store knowledge). However, there is the opportunity for knowledge to be captured and encoded in more explicit forms which can be queried and used in a more structured form. Ontologies are one such technology and the use of visualization is a means of making this wealth of information even more accessible.

1.2 Aims and Objectives

With the motivations described above we aim to show how information visualization can be useful for non-expert users. The visualizations should provide a degree of cognitive insight which the user would not gain by looking at the source data in its original (unvisualised) form. In particular, we aim to show how "real-life, everyday data" which has not been explicitly semantically annotated can be usefully visualized.

1. Our first objective is to gain an understanding of the areas of information visualization and ontologies in enough detail such that we can asses and exploit the current state of the art in automatic and knowledge-based visualization systems.

2. Our second objective is to create a visualization pipeline which can take non-semantically marked-up data and automatically create a cognitively useful visualization.

3. Our third objective is to exploit domain knowledge, both from the source data's subject domain and the area of information visualization, in order to create the most cognitively valuable visualization.

4. Finally, we wish to evaluate the visualizations amongst the target user group (non-expert users), to ascertain whether the technique has been successful.

In the objectives above, we discuss the need for *automatic* visualization. The term automatic is broad and needs to be defined explicitly. When a system is described as automatic, it rarely means that it is fully automatic and a certain degree of manual intervention is usually necessary. In the field of visualization we define automatic from the perspective of three actors:

Developer The developer of an automatic visualization system will create the new system entirely manually. Therefore this actor has no access to automated features. However, the developer may choose to reuse components of existing visualization pipelines in order to save expense and simplify the development process. Fundamentally, the developer's role is a non-automatic one.

Maintainer With any live software system there is the need for a maintenance role. This role may be played by the original developer or a different person. The role of the maintainer is to add functionality to the system as and when required (e.g. add the ability to handle new source data domains or new visualization techniques) and to ensure the high level of service (e.g. up time, response time). Again, this role is a non-automatic one in that the maintainer must manually add functionality using a similar set of tools to the original developer. A certain degree of automaticity may be available through specific tools which simplify common tasks (e.g. adding a new visualization technique).

Administrator An administrator's role is similar to that of a maintainer in that the role is present to ensure the ongoing running of the system. However, the responsibilities are less technical in their nature. An example would include adding new users to the system and ensuring provisions for back-up of data are in place. This role has a greater degree of automation than a maintainer and as such, the administrator role can often be done as a part-time task by a knowledgeable end user of the system.

End User This role has the most potential for automation and indeed the whole rationale for a system might be to make this role's task as automated as is possible. In the context of a visualization system, automation means that the user *does not* have to:

- Prepare the source data into a specific format ready for the visualization tool (there may be some constraints on this).
- Decide on a mapping between source data entities and target visualization artefacts.
- Decide on parameters, limits or configuration values.

The user therefore can expect the following responsibilities from an automatic visualization system:

1. User: tell the system where the source data to be visualized resides.

2. System: present the user with one or more visualizations which are cognitively more useful than the source data presented in its original format.

As discussed in this section, we will use the definitions of: *expert* and *non-expert*; and *manual* and *automatic* throughout the remainder of this thesis.

1.3 Thesis Outline

In order to set our research in context, we present a thorough investigation into the fields of Information Visualization, Ontologies and Ontology Mapping. In chapter 2, we discuss how the field of information visualization has evolved and present a theoretical framework for describing visualization techniques and the toolkits which can produce visualizations. In chapter 3, we describe the fields of ontologies and ontology mapping. We discuss the application of these technologies together with an evaluation of different theoretical frameworks. From this foundation, we present the theories and technologies which make up the concepts of visualization as mapping in chapter 4. In chapter 5, we present a constraint-based visualization toolkit which we evaluate within the context of a real-life dataset. In chapter 6, we present a visualization pipeline which exploits domain knowledge and visualization knowledge to automatically produce visualization from web data. In chapter 7, we evaluate the scalability and usability of the tool produced in the previous chapter with a more in-depth example and a user testing exercise. We conclude and present future work in chapter 8. In the rest of this section, we give a more detailed overview of each chapter.

1.3.1 Chapter 2: Information Visualization Systems

In this chapter, we describe the historic development of the field of information visualization. We discuss the early work on defining a theoretical framework of the Human Vision System (HVS) which gives rise to the various techniques of information visualization. Additionally, we describe the various toolkits and frameworks available for general-purpose visualization.

Throughout the description of the practical contributions from the information visualization community, we discuss the theoretical frameworks which underpin the work. In this way, we aim to generalise our approach for any source data format and any visualization technique.

1.3.2 Chapter 3: Ontologies and Ontology Mapping

In this chapter, we define ontologies and describe the technical standards for representation and inferencing. We describe associated tools such as editors and the application of ontologies. We define Ontology Mapping and associated fields together with a discussion of the various approaches and algorithms which can be applied. We also discuss the theory behind the Semantic Web and describe some practical tools which have been developed in this area.

1.3.3 Chapter 4: Concepts of Visualization as Mapping

In this chapter we discuss the various concepts which we need to consider when discussing visualization as mapping. This includes: a study of automatic mapping techniques; a description of a high-level framework for producing audial, visual and textual representations of semantically rich data; and a comparison of Tree and Graph-centric mapping approaches.

1.3.4 Chapter 5: VizThis : A Tree-centric Mapping Toolkit for Information Visualization

In this chapter we discuss the production of a visualization toolkit which takes a Tree-centric approach to mapping, called VizThis. The tool provides a facility for automatic mapping which is based on a type-constrained system. The tool has additional features to assist users in a manual mapping process. These include mapping locks, constraint warnings, data cleansing and value transformation. This chapter finishes with a user testing exercise and an evaluation of the tool.

1.3.5 Chapter 6: SemViz : From Web Data to Visualization via Ontology Mapping

In this chapter we take an Ontology-centric approach to the production of an automatic visualization pipeline. The tool, called SemViz, uses three ontologies to capture useful semantics of the pipeline. The ontologies are: a Domain Ontology (DO) to capture semantics about the subject domain of the data which is to be visualized; a Visual Representation Ontology (VRO) to capture semantics relating to different visualization techniques (e.g., Tree Map, Parallel Coordinates etc); and a Semantic Bridging Ontology (SBO) to capture specific expert knowledge about particular mappings between domain concepts and visual representation concepts. We demonstrate the application of the technique with data from popular music chart data available on the web. The tool outputs visualizations using the ILOG Discovery and Prefuse visualization toolkits.

1.3.6 Chapter 7: SemViz Case Study : Scalability Evaluation and User Testing

In this chapter, we consider a more extensive example of the use of SemViz. We use datasets of football statistics from a variety of sources and perform a comparative evaluation of the visualizations produced. We evaluate the quality of the visualizations against the score and rank which the SemViz tool gives. Evaluation is conducted by three groups: the author; a group of test subjects; and a visualization expert. We also evaluate how well the technique scales, in terms of: number of different source datasets (with different schemas); number of records; and number of source concepts.

1.3.7 Chapter 8: Conclusions

In this chapter, we summarise the findings of this research work. We compare our results against our original objectives and discuss the advantages and disadvantages of an ontological approach to information visualization. We also discuss the opportunities and direction for further work.

1.4 Papers

Below we list the papers which have been produced from the research work detailed in this thesis. The third paper won the Best Paper Award at EuroVis 2008.

Information Realisation: Textual, Graphical and Audial Representations of the Semantic Web Owen Gilson, Nuno Silva, Phil W. Grant, Min Chen and Joo Rocha; *I-KNOW '06 - 6th International Conference on Knowledge Management special track on Knowledge Visualization and Knowledge Discovery*; Graz, Austria; 2006. [GSG$^+$06]

VizThis: Rule-based Semantically Assisted Information Visualization Owen Gilson, Nuno Silva, Phil W. Grant, Min Chen and Joo Rocha; *International Semantic Web Conference 2006 at The 3rd International Semantic Web User Interaction Workshop*; Athens, Georgia, USA; 2006. [GSG$^+$07]

From Web Data to Visualization via Ontology Mapping Owen Gilson, Nuno Silva, Phil W. Grant and Min Chen; *Proc. Eurographics / IEEE VGTC Symposium on Visualization (EuroVis '08)*; Eindhoven, Netherlands; 2008. [GSGC08]

1.5 Contributors

Below we list all persons involved in the research work detailed in this thesis.

1.5.1 Owen Gilson

This person is the main contributor and author of this thesis. He designed and implemented all the software detailed (including the Visualization As Mapping prototype, VizThis and SemViz). He designed, conducted and analysed the data from the user testing study.

1.5.2 Phil W. Grant

This person is a co-supervisor of the work. He provided direction, supervision and background for the research work. He also provided indepth knowledge in the areas of knowledge based systems and visualization.

1.5.3 Min Chen

This person is a co-supervisor of the work. He provided direction, supervision and background for the research work. He also provided indepth knowledge in the area of visualization.

1.5.4 Nuno Silva

This person provided detailed background knowledge on the use of semantic web, ontologies and ontology mapping techniques. He also assisted in the design of the software tools.

1.5.5 João Rocha

This person provided high-level advice on the direction of the initial ontology work.

Chapter 2

Information Visualization Systems

Contents

2.1 Introduction

In this chapter we discuss the research area and application of Information Visualization. We describe the history of the area, its fundamental concepts, and the tools and techniques which have evolved over the last 30 years. We begin by defining the term, "information visualization" and discussing the evolution of the subject. In particular, we focus on how the area has been formalised. Formalisation is important in order to gain a depth of understanding, but also as a precursor to developing automatic visualization systems. Therefore, this area of investigation provides an important basis for the work in this thesis. After discussing formalisation, we list and evaluate a selection of visualization toolkits and discuss their relative merits before discussing the area of automatic visualization in more detail. We then summarise our findings and discuss how this relates to the goals set out in chapter 1.

This chapter is not intended to provide a comprehensive survey of information visualization techniques. Rather, it aims to cover the formalisation of the discipline and to describe how this has resulted in visualization frameworks and toolkits and the development of automated techniques. For detailed descriptions of information visualization techniques, readers are pointed to important books from: Card, Mackinlay and Shneiderman [CMS99]; Spence [Spe00]; and Ware [War04]. Recent surveys on information techniques include: [HSMM00]; [HG02]; [FL03]; and [KHG03].

2.2 Definition of Information Visualization

Perhaps one of the most commonly cited definitions of Information Visualization comes from [CMS99]:

> The use of computer-supported, interactive, visual representations of data to amplify cognition.

We can utilise humans' visual perception in order to provide heightened levels of cognition which would not be as effective if the information were presented in a less sophisticated form such as textually. The reason for this is that a human's visual perception system has an amazing ability to to scan, recognise, and recall images rapidly. Additionally, there is an innate ability to detect patterns and changes in: size; colour; shape; movement; and texture. Information Visualization therefore presents information visually in order to offload the cognitive effort from the human's more conscious, analytical brain area to their visual perception system.

A closely related area to Information Visualization is that of Scientific Visualization. The two are differentiated by their application area, data type and spatialisation:

Scientific Visualization

- Application Area : Science
- Data: Physically-based
- Spatialisation: Provided
- Example: Visualising the flow of a gas.

Information Visualization

- Application Area : Non-Scientific (usually)
- Data: Abstract
- Spatialisation: Assigned
- Example: Visualising the links between pages in a web site.

However, in practice, there is a lot of commonality between the disciplines of Scientific and Information Visualization and many application domains straddle both areas.

2.3 Formalisation of Visualization

2.3.1 Visual Variables

One of the most important pieces of work used by the information visualization community is Jacques Bertin's Sémiologe Graphique [Ber83]. Bertin described visual marks and the ways by which these marks can be modified to communicate information. These are called visual variables or visual attributes.

Marks are defined as:

Points These are dimensionless locations on the plane, represented by signs that must have some size, shape or colour for visualization.

Lines These represent information with a certain length, but no area and therefore no width. Lines are visualised by signs of some thickness.

Areas These have a length and a width and therefore a two-dimensional size.

Surfaces These are areas in a three-dimensional space, but with no thickness.

Volumes These have a length, a width and a depth. They are therefore actually three-dimensional.

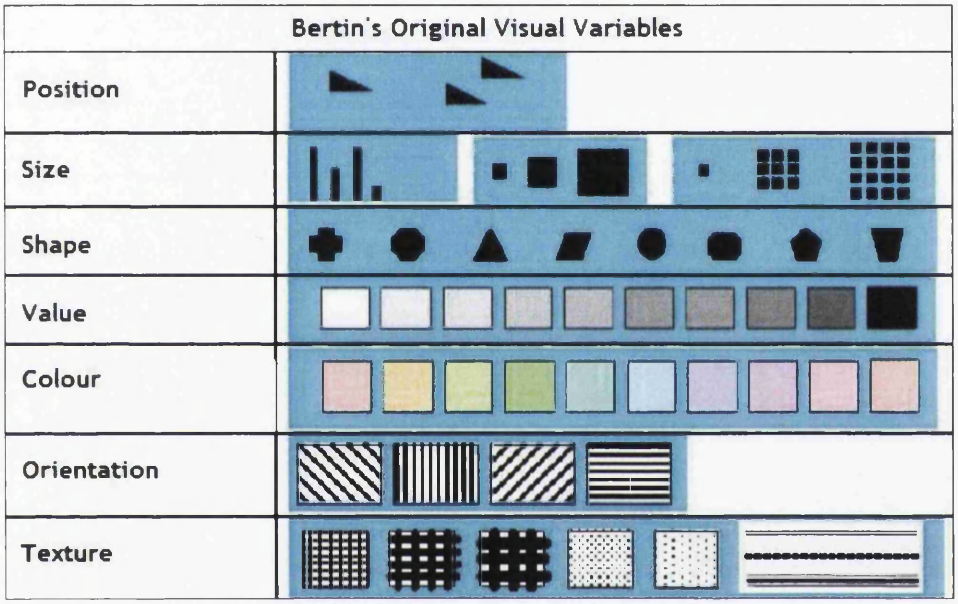


Figure 2.1: Bertin's Seven Original Visual Variables (from [Car03]).

Bertin defined seven visual variables (see figure 2.1):

Position Changes in the X or Y location.

Size Changes in length, area or repetition.

Shape Infinite number of shapes.

Value Changes from light to dark.

Colour Changes in hue at a given value.

Orientation Changes in alignment.

Texture Variation in "grain".

While providing a fundamental basis for information visualization, research into the use of visual variables still continues when considered in new contexts such as multi-surface environments [WSFB07].

Mackinlay [Mac86] later expanded the list of variables and also added an ordering to the variables depending on the task as part of his work on APT (A Presentation Tool). In the lists below, the variables with the highest accuracy are listed first.

Quantitative Task accuracy Position, Length, Angle, Slope, Area, Volume, Density, Colour Saturation.

Ordinal Task accuracy Position, Density, Colour Saturation, Colour Hue, Texture, Connection, Containment, Length, Angle, Slope, Area, Volume.

Nominal Task accuracy Position, Colour Hue, Texture, Connection, Containment, Density, Colour Saturation, Shape, Length, Angle, Slope, Area, Volume.

The most recent work has been in expanding this list to cover motion [Car03]. Additional variables include consideration of changes in: direction; speed; frequency; rhythm; flicker; trails; and style.

2.3.2 Interactive Visualization

With the advent of faster graphics technology, it became possible to present interactive visualizations. Indeed interaction support is just as important as the basic visual representation being presented. Robertson et al. [RMC91] state that if objects are smoothly animated over a period of around one second, the object constancy eliminates the need for re-assimilation of the visualization scene. This reduces the cognitive burden on a user by allowing them to keep their mental model the same despite the graphical scene having changed in some form (position, perspective, zoom level etc.). Modern day visualization applications and toolkits produce interactive visualizations as a core part of their functionality.

2.3.3 Visualization Taxonomies

The initial formalisation work of Bertin and Mackinlay was built on to create taxonomies of visualization. Work on this task continued in parallel between the information and scientific visualization communities, the focus being on the higher level of abstraction of visualization techniques rather than variables.

In Scientific Visualization, the work was pioneered by Wehrend and Lewis [WL90] who classified approximately 400 techniques based on *objects* (scalars, vectors, positions etc.) and *operations* (locate, compare etc.). Similar classifications were presented by Keller and Keller [KK94]. Brodlie created a classification based on E-notation. E-notation captures a *model* of what is being visualised. This was later extended into an O-notation [Bro93] which also captures a *view* of how the visualization takes place.

In Information Visualization, the work was developed by Shneiderman [Shn96] who developed the *Visual Information-Seeking Mantra*:

> Overview first, zoom and filter, then details on demand.

In this work, Shneiderman also creates a *Task by Type Taxonomy* (commonly known as TTT). The seven tasks being:

Overview Gain an overview of the entire collection.

Zoom Zoom in on items of interest.

Filter Filter out uninteresting items.

Details-on-demand Select an item or group and get details when needed.

Relate View relationships among items.

History Keep a history of actions to support undo, replay, and progressive refinement.

Extract Allow extraction of sub-collections and of the query parameters.

Additionally there are seven data types which were identified:

- 1-dimensional (documents, source code, etc.)
- 2-dimensional (map data)
- 3-dimensional (real world objects)
- Temporal (time lines)
- Multidimensional (e.g., FilmFinder)
- Tree (e.g., Treemaps)
- Network (e.g., visualising links in the world wide web.)

Tory and Moller [TM04] looked at the field of visualization as a whole by unifying the work in earlier taxonomies. One of the results of their work was the creation of low-level taxonomies for discrete and continuous models.

2.3.4 Reference Models

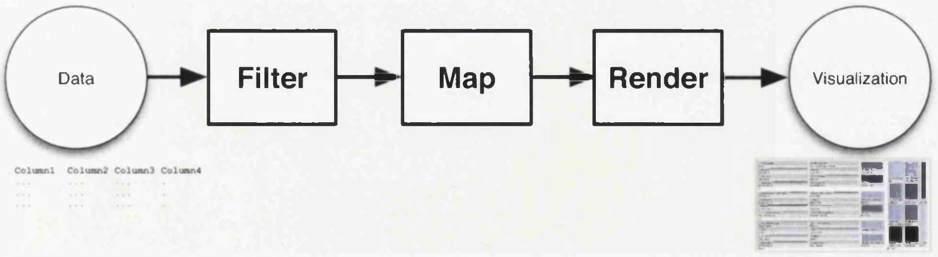


Figure 2.2: The Haber-McNabb Dataflow Reference Model [HM90].

While knowledge of visualization techniques has evolved through the formalisation of visual variables, tasks and datatypes, the *process* of creating a visualization also deserves investigation. Haber and McNabb [HM90] defined a dataflow reference model (see figure 2.2) which defines the process of creating a visualization at a level of abstraction which

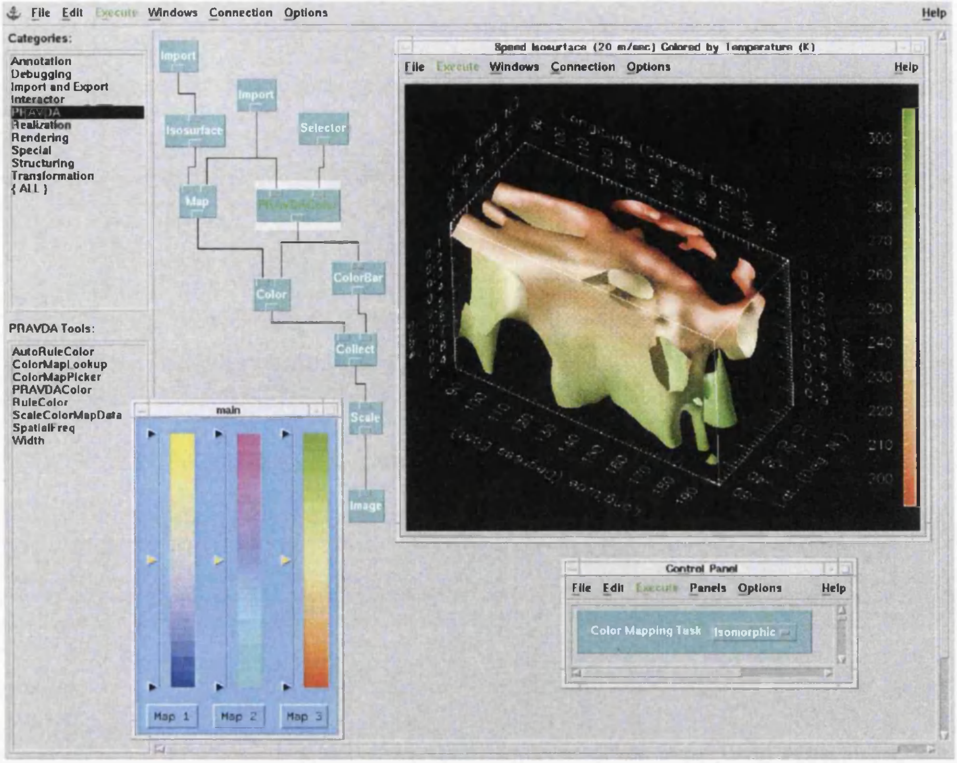


Figure 2.3: IRIS Explorer [Fou95] [Gro04] and the modular visual programming environment.

can be applied to many visualization systems. The visualization pipeline stages are: read in the data; select the data of interest; construct a geometrically correct visualization; and render this geometry as an image. The reference model has been used as a basis for many well known visualization environments such as Khoros [KR94] (now known as VisiQuest) and AVS (Advanced Visual Systems) [FV04]. Systems such as the modular IRIS Explorer [Fou95] [Gro04] (see figure 2.3) allow users to add and structure their own modules using a visual programming approach. IBM also produced a similar system known as Open Visualization Data Explorer [IBM04] (see figure 2.14).

2.3.4.1 Conceptual, Logical and Physical Layers

Wood et al. [WWB97] and Duke et al. [DGC$^+$98] extended the Haber-McNabb reference model by considering collaborative visualization (see figure 2.4). This was further extended to develop a three-layer reference model which allowed progressive resources to be bound into the original Haber-McNabb pipeline [BDSW04]. The three layers are described below:

Conceptual Layer This describes the intent of the visualization (e.g., show me all web pages with link distance of four from the homepage).

Logical Layer In this layer, the modules of the software system are bound in (e.g., use Prefuse, vtk etc.)

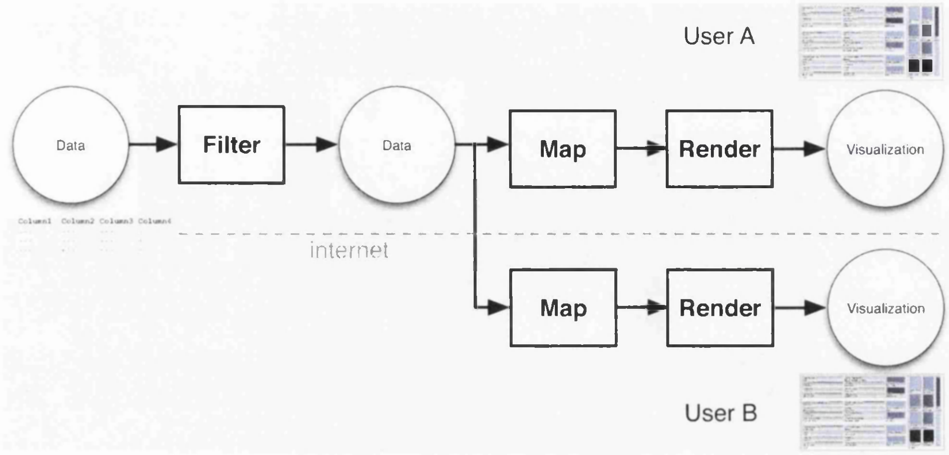


Figure 2.4: The Haber-McNabb Dataflow Reference Model extended to consider collaborative visualization [WWB97] [DGC^{+}98].

Physical Layer In this layer, the particular resources to be used are bound in (e.g., use a given "grid" resource)

These parameters can be formalised at the conceptual level in a language called skML [DS05]. The result of this 3-layer reference model are systems which are both distributed (can run on any available resource on a "grid" network) and collaborative (allows multi-user visualization to be exploited).

2.4 Visualization Toolkits and Frameworks

In this section we describe some of the toolkits and frameworks which have emerged over the last 30 years. There have been many hundreds of different pieces of work produced by commercial and academic groups. However, we focus on the most influential ones in relation to this work and also the research area as a whole. We categorise the visualization tools into one of five categories. Naturally, some visualization toolkits can be positioned in multiple categories. We choose to position the tools into their most dominant category or the one in which it is most commonly known. Within each category, we discuss the tools in chronological order. The first category we start with is that of Domain Specific tools. These are interesting since they provide a high-degree of usability at the expense of being source domain specific and often have a set range of visualization techniques. We then go on to describe the ability of General Purpose visualization tools to address the short-comings of the previous category whilst also discussing their own short-comings with respect to usability. The next category is the least user-friendly in that is discusses Visualization Programming Frameworks. These have the most flexibility but obviously require the greatest development effort. Finally, we discuss the newest category which is Collaborative Visualization Toolkits. Four of the five tools have been developed in the last two years in response to the growing capability of web technologies to display interactive visualizations. The other tool is over ten years old and is still based on web technologies, but its visualizations are largely static.

2.4.1 Domain Specific Visualization Tools

In this section we describe some of the more influential domain specific visualization tools. These tools are able to visualise a certain data set or data in a certain format very well. Since the creators of the visualization can restrict the bounds of the source data, the visualization technique and controls can be highly customised to the domain being modelled. Typically, a domain specific tool will emerge after a research area has developed a new visualization technique. This domain specific tool then goes on to become popular, acting as a showcase for the original research work. This is exactly what happened with MarketMap [Sma08] which was one of the first popular use of Tree Maps outside of the research and technical community.

Domain specific visualization tools have served as an effective method of demonstrating the benefits of information visualization to wider audiences. However, they highlight the failure of general purpose visualization toolkits to appeal to this wider audience. This validates our motivation for attempting to "bring general purpose visualization techniques to non-expert users".

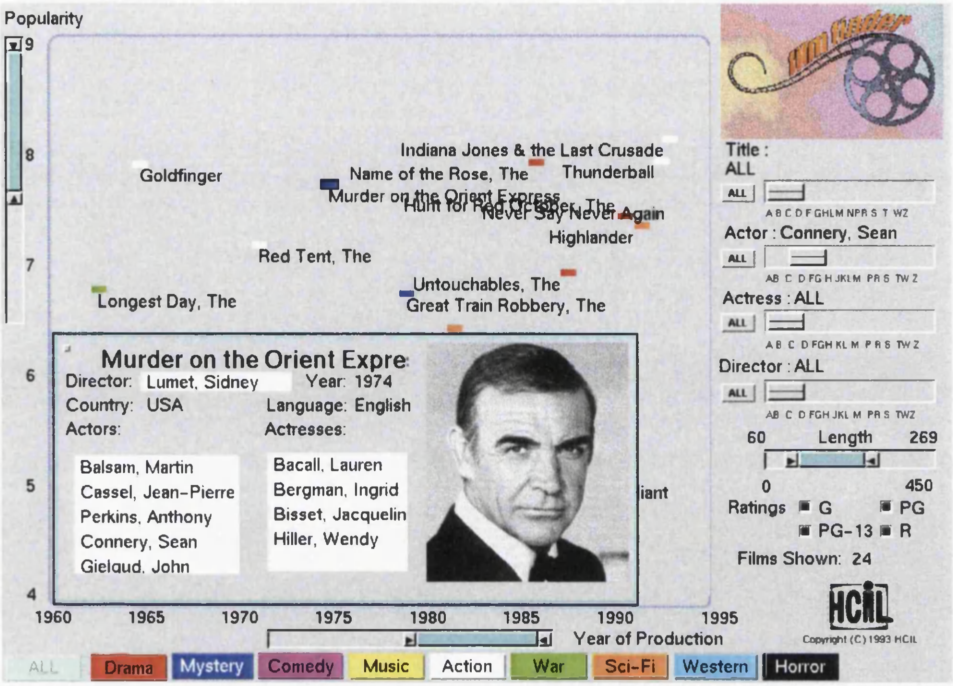
In table 2.1, we give an overview of five influential domain specific visualization tools. Each tool is described in more detail below.

	FilmFinder	MarketMap	GapMinder	CoMIRVA	Sense.us
Year	1996	1999	2006	2007	2007
Platform	X Windows	Java	Flash	Java	Web
Open Availability	N	Y	Y	Y	N
Use Own Data	N	N	N	Y	N
Create Vis's	N	N	N	Y	N
Discuss Vis's	N	N	N	N	Y
View sharing	N	N	N	N	Y
Annotation	N	N	N	N	Y
Automatic Vis's	n/a	n/a	n/a	N	n/a
Vis Techniques	Star Field	TreeMap	Scatter	6 types	6 types
Interactive / Static	I	I	I	I	I
Domain Area	Films	Stock Market	WHO stats	Music	US Census
Direct Data Editing	N	N	N	N	N
Data Re-shaping	N	N	N	Y	N

Table 2.1: An overview of Domain Specific Visualization Tools.

2.4.1.1 Film Finder

FilmFinder [AS94] (see figure 2.5) was an early implementation of dynamic queries. The source data set is fixed as film information. FilmFinder refined the techniques for starfield



Murder on the Orient Expre

Director: Lumet, Sidney Year: 1974

Country: USA Language: English

Actors: Balsam, Martin; Cassel, Jean-Pierre; Perkins, Anthony; Connery, Sean; Gielgud, John

Actresses: Bacall, Lauren; Bergman, Ingrid; Bisset, Jacquelin; Hiller, Wendy

Title : ALL

Actor : Connery, Sean

Actress : ALL

Director : ALL

Length 60 269 (0 450)

Ratings ■ G ■ PG ■ PG-13 ■ R

Films Shown: 24

Figure 2.5: The FilmFinder [AS94] visualization tool showing a dynamic query being performed.

displays (zoomable, colour coded, user-controlled scattergrams), and laid the basis for the commercial product Spotfire [Ahl96].

2.4.1.2 Market Map

Market Map [SW01] (see figure 2.6) was one of the first popular implementations of Tree Maps. Another one was for the visualization of disk space usage [Der08]. MarketMap uses an extension of tree maps, avoiding excessively narrow strips. Heuristics are also used to slice up each rectangle into more evenly proportioned sub-rectangles.

2.4.1.3 Gap Minder

The Gap Minder organisation have a tool called Trend Analyser [RR06] which visualises world economic and health statistics (see figure 2.7). It takes the form of a Flash application preloaded with statistical and historical data about the development of the countries of the world. The software was acquired by Google in March 2007 and parts of it (particularly the Flash-based Motion Chart gadget) have become available for public use as part of the Google Visualizations API.

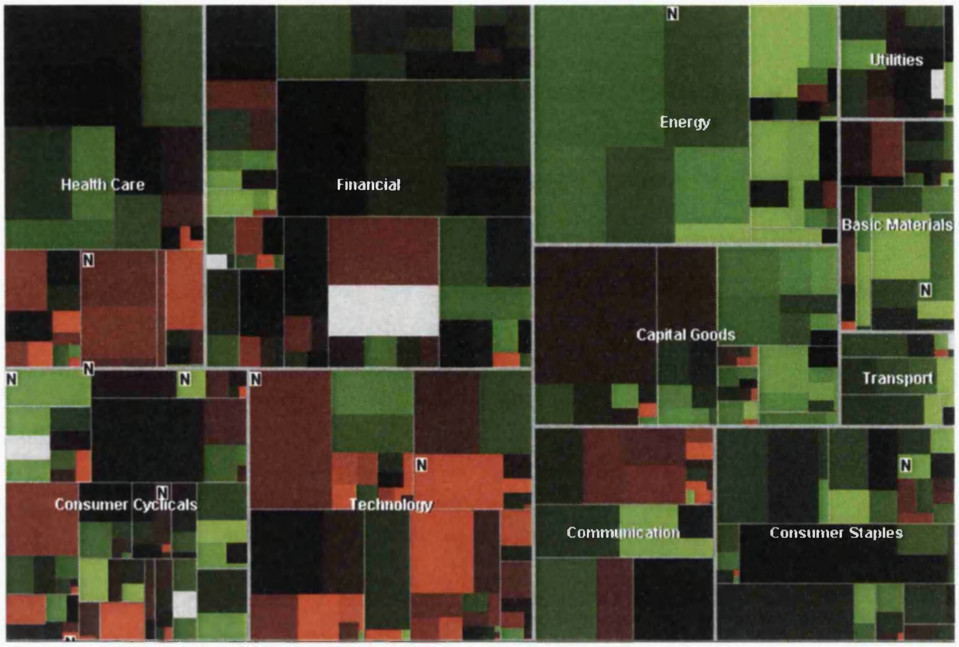


Figure 2.6: Smart Money's MarketMap [SW01] showing the market activity for stocks in all major sectors.

2.4.1.4 CoMIRVA

CoMIRVA [SKSP07] [SKW05] (Collection of Music Information Retrieval and Visualization Applications) is a Java framework for information retrieval and visualization whose main functionalities are music information retrieval, web retrieval, and visualization of the extracted information (see figure 2.8). CoMIRVA supports the following visualization techniques: Self-Organizing Map grid; Smoothed Data Histogram; Circled Bars; Circled Fans; Continuous Similarity Ring; and Sunburst.

2.4.1.5 Sense.us

The Sense.us system [HVW07] (see figure 2.9) supports asynchronous collaboration across a variety of visualization types. Its domain is constrained to US Census data within a closed user group. It has extensive collaboration features such as view sharing, discussion, graphical annotation, and social navigation and includes novel interaction elements.

2.4.2 General Purpose Visualization Toolkits

2.4.2.1 Spotfire

Spotfire [Ahl96] (see figure 2.10) is a direct descendent of the FilmFinder system. The software is now positioned as a Business Intelligence tool after its acquisition by Tibco

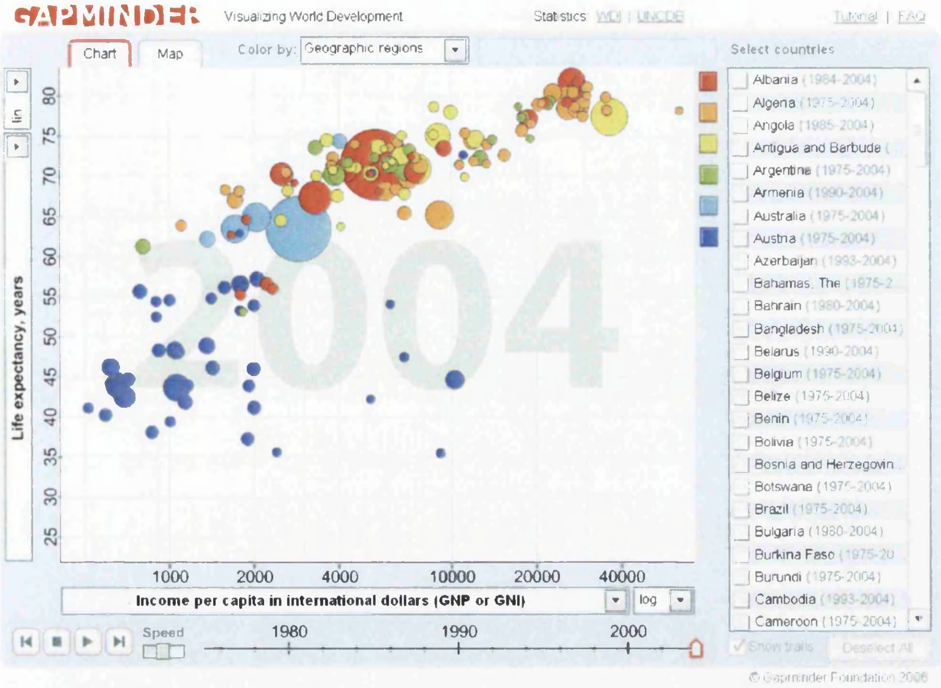
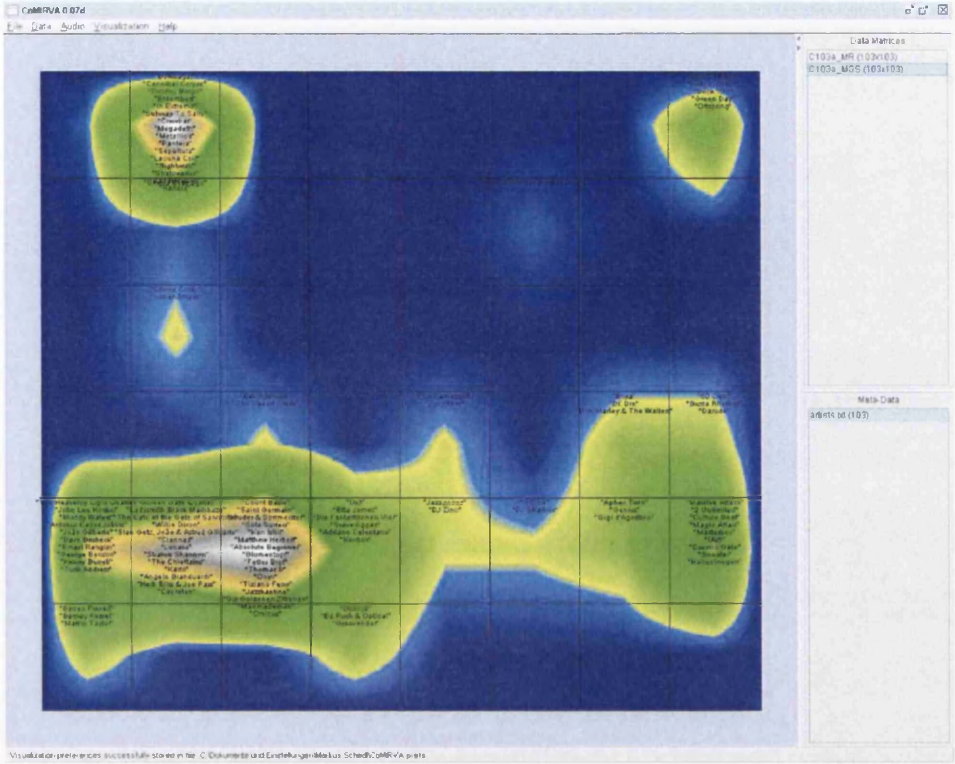


Figure 2.7: GapMinder's TrendAnalyser [RR06].

	SpotFire	ILOG Discovery	Tableau	Fractal:Edge
Year	1996	2004	2007	2008
Platform	Windows	Java	Windows	Windows
Open Availability	Y	Y	Y	Y
Use Own Data	Y	Y	Y	Y
Create Vis's	Y	Y	Y	Y
Discuss Vis's	Y	N	N	N
View sharing	Y	Y	N	N
Annotation	N	N	N	N
Automatic Vis's	Constraint-based	N	Constraint-based	N
Vis Techniques	> 10	6	> 10	1
Interactive / Static	I	I	I	I
Domain Area	General Purpose	General Purpose	General Purpose	General Purpose
Direct Data Editing	Y	Y	Y	N
Data Re-shaping	N	N	Y	N

Table 2.2: An overview of General Purpose Visualization Tools.

in 2007. Spotfire pioneered the accessibility of dynamic queries and starfield displays to business users. Through its direct manipulation model, these visualization techniques were brought to a wider audience.



Visualization preferences successfully stored in file C:\Dokumente und Einstellungen\Markus Schedl\CoMIRVA.prefs

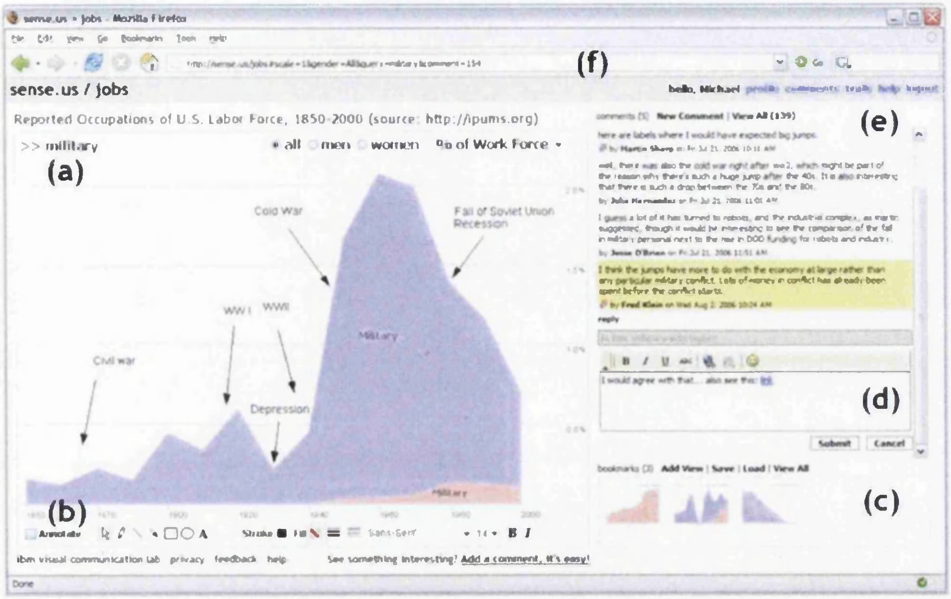
Figure 2.8: CoMIRVA [SKSP07] [SKW05].

2.4.2.2 Tableau

The Tableau system [MHS07] (see figure 2.11) follows Mackinlay's early APT work into automatic visualization. It provides users with suggested visualizations based on the field types of a source dataset which they choose. Its automaticity is founded on a constraint-based type system which is discussed further in section 2.5.2.

2.4.2.3 ILOG Discovery

The ILOG Discovery system [Bau02, BHS03, Bau04, Bau06] (see figure 2.12) is based on a canonical representation of data-linear visualization algorithms. The algorithms are inspired by the co-routine mechanisms of the CLU programming language [LAB^{+}79]. Linear-state-dataflows provide a canonical representation for a large class of visualization algorithms, called data-linear visualizations. The ILOG Discovery software supports a direct manipulation model and a unified Projection Inspector (independent of the visualization technique being used) for adjusting source data mappings and visualization parameters.


sense.us » jobs - Mozilla Firefox
(f)
sense.us / jobs
hello, Michael profile comments trails help logout
Reported Occupations of U.S. Labor Force, 1850-2000 (source: http://ipums.org)
>> military
● all ○ men ○ women % of Work Force
(a)
Cold War
Fall of Soviet Union Recession
WW I
WWII
Civil war
Depression
Military
Military
(b)
Annotate
Stroke Fill Sans-Serif 14 B I
ibm visual communication lab privacy feedback help
See something interesting? Add a comment, it's easy!
Done
comments (5) New Comment | View All (139)
(e)
reply
B I U
(d)
Submit Cancel
bookmarks (3) Add View | Save | Load | View All
(c)

here are labels where I would have expected big jumps.
by Martin Sharp on Fri Jul 21, 2006 10:11 AM

well, there was also the cold war right after ww2, which might be part of the reason why there's such a huge jump after the 40s. It is also interesting that there is such a drop between the 70s and the 80s.
by Julia Hernandez on Fri Jul 21, 2006 11:01 AM

I guess a lot of it has turned to robots, and the industrial complex, as martin suggested, though it would be interesting to see the comparison of the fall in military personal next to the rise in DOD funding for robots and industry.
by Jesse O'Brien on Fri Jul 21, 2006 11:51 AM

I think the jumps have more to do with the economy at large rather than any particular military conflict. Lots of money in conflict has already been spent before the conflict starts.
by Fred Klein on Wed Aug 2, 2006 10:24 AM

I would agree with that... also see this: link

Figure 2.9: The Sense.us [HVW07] web-based visualizer of US Census statistics. (a) A stacked time-series visualization of the U.S. labor force. (b) A set of graphical annotation tools. (c) A bookmark trail of saved views. (d) Text-entry field for adding comments. (e) Threaded comments attached to the current view. (f) Automatically updated URL for the current state of the application.

2.4.2.4 Fractal:Edge

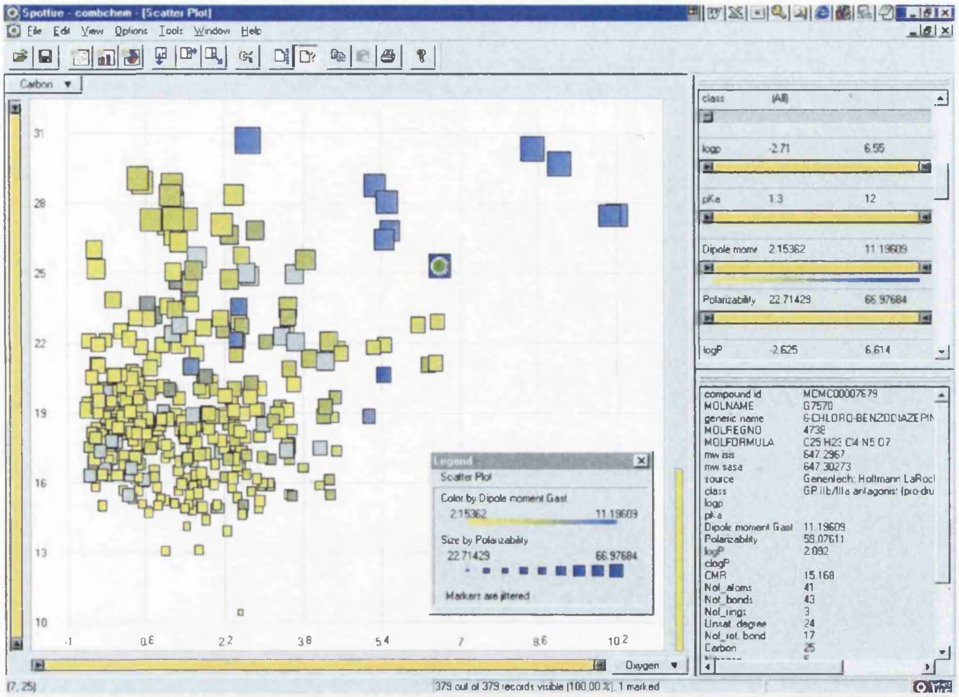
The Fractal:Intelligence tool [Fra08] (see figure 2.13) uses a novel visualization technique to show the top-level overview, intermediate summaries and the underlying detail for many records. While it has an ability to handle many thousands of records, the whole system is centred around the fractal inspired visualization technique. This technique requires a certain amount of training and familiarisation and is therefore not suitable for beginner or novice visualization users.

2.4.3 Visualization Programming Frameworks

In order to assist the process of creating visualization systems, a number of frameworks exist. These frameworks provide APIs (Application Program Interface) to allow programmers to create systems while utilise the frameworks' visualization techniques. This approach is particularly effective when designing domain specific visualization tools.

2.4.3.1 nVizN

nViZn [Wil05] is the successor of the Graphics Production Library (GPL) and is a set of Java class libraries for interactive statistical graphics on the Web. nVizN supports a number of visualization techniques and widgets (sliders, buttons, magnifiers, etc.) to manipulate any


Spotfire - combchem - [Scatter Plot]
File Edit View Options Tools Window Help
Carbon
31
28
25
22
19
16
13
10
-1
0.6
2.2
3.8
5.4
7
8.6
10.2
Oxygen
Legend
Scatter Plot
Color by Dipole moment Gast
2.15362
11.19609
Size by Polarizability
22.71429
66.97684
Markers are jittered
class (All)
logp -2.71 6.55
pKa 1.3 12
Dipole mome 2.15362 11.19609
Polarizability 22.71429 66.97684
logP -2.625 6.614
(7, 25)
379 out of 379 records visible (100.00 %), 1 marked

compound id MCMC00007679
MOLNAME G7570
generic name 6-CHLORO-BENZODIAZEPIN
MOLREGNO 4738
MOLFORMULA C25 H23 Cl4 N5 O7
mw isis 647.2967
mw sasa 647.30273
source Genentech; Hoffmann LaRocl
class GP IIb/IIIa antagonist (pro-dru
logp
pka
Dipole moment Gast 11.19609
Polarizability 59.07611
logP 2.092
clogP
CMR 15.168
Nof_atoms 41
Nof_bonds 43
Nof_rings 3
Unsat degree 24
Nof_rot bond 17
Carbon 25

Figure 2.10: Tibco's SpotFire toolkit [Ahl96].

	nVizN	IVTK	OpenDX	Polaris	Piccolo	Prefuse
Year	2005	2004	2001	2002	2004	2005
Platform	Java	Java	Motif	OpenGL	Java or.NET	Java or Flash
Open Availability	Y	Y	Y	Y	Y	Y
Use Own Data	Y	Y	Y	Y	Y	Y
Create Vis's	Y	Y	Y	Y	Y	Y
Discuss Vis's	N	N	N	N	N	N
View sharing	N	N	N	N	N	N
Annotation	N	N	N	N	N	N
Automatic Vis's	N	N	N	N	N	N
Vis Techniques	> 5	9	open	1	open	> 10
Interactive / Static	I	I	I	I	I	I
Domain Area	GP	GP	GP	GP	GP	GP
Direct Data Editing	N	N	N	N	Y	N
Data Re-shaping	N	N	N	N	N	N

Table 2.3: An overview of Visualization Programming Frameworks.

aspect of a graphic. These features give it the capability for drill-down, brushing, zooming, and other exploration. In this way, nVizN positions itself as a data mining tool.

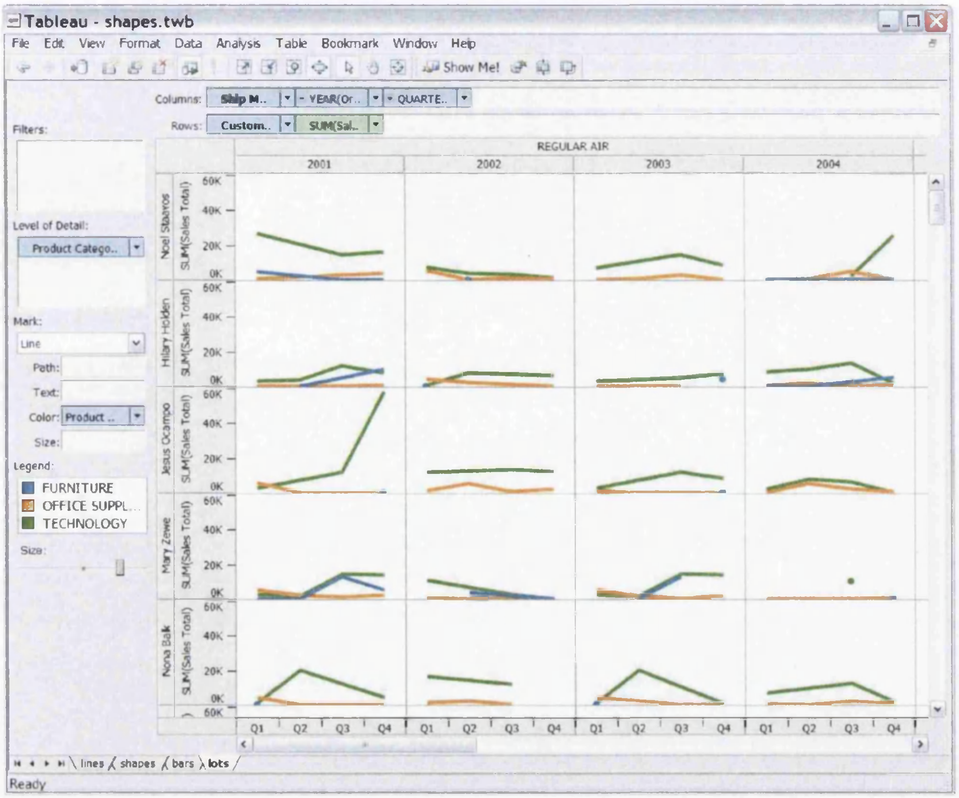


Tableau - shapes.twb

Columns: Ship M.. | YEAR(Or.. | QUARTE..

Rows: Custom.. | SUM(Sal..

Level of Detail: Product Catego..

Mark: Line

Color: Product ..

Legend:
- 🟦 FURNITURE
- 🟧 OFFICE SUPPL..
- 🟩 TECHNOLOGY

Ready

Figure 2.11: The Tableau [MHS07] visualization toolkit.

2.4.3.2 IVTK (The InfoVis Toolkit)

IVTK (The InfoVis Toolkit) [Fek04] is a Java based framework for developing Information Visualization applications and components. It implements nine types of visualization: Scatter Plots; Time Series; Parallel Coordinates and Matrices for tables; Node-Link diagrams; Icicle trees and Treemaps for trees; Adjacency Matrices and Node-Link diagrams for graphs. Additionally, it has a unified set of interactive components which allow interactive filtering. These dynamic queries can be performed with the same control objects regardless of the data structure.

2.4.3.3 OpenDX

OpenDX (Open Data Explorer) [TBF01] [IBM04] is IBM's scientific data visualization software (see figure 2.14). It supports a high-level scripting language and also a visual program editor which can be used to create and modify workflows. OpenDX is a Motif widget toolkit on top of the X Window System.

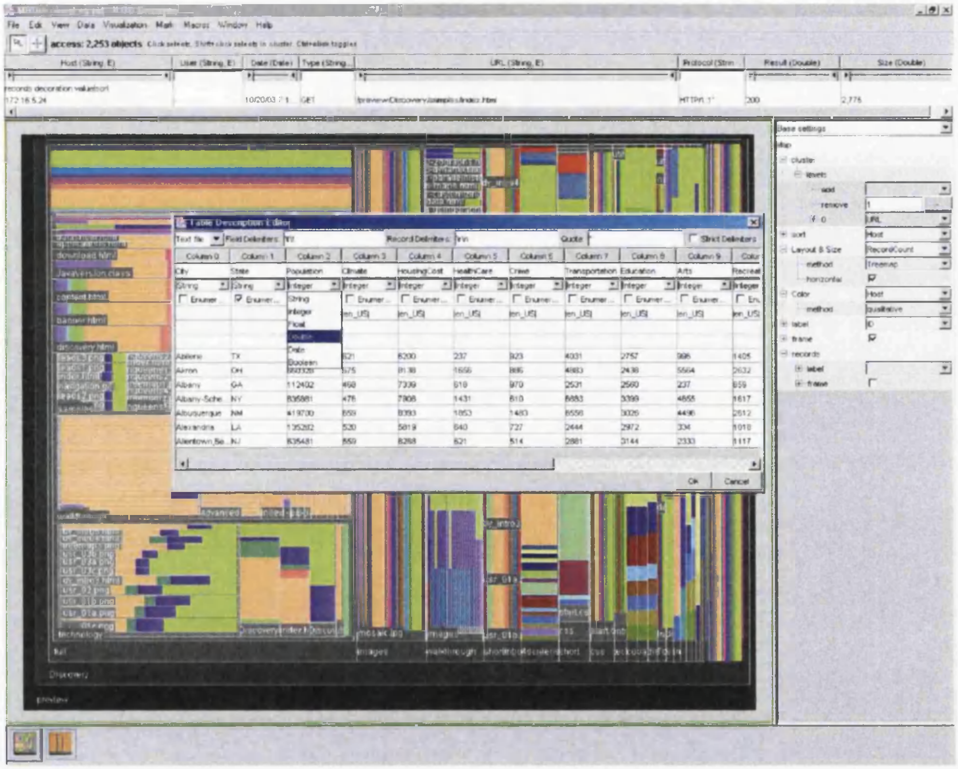


Table Description Editor

Text file | Field Delimiters: \t | Record Delimiters: \r\n | Quote: " | ☐ Strict Delimiters

Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Colur
City	State	Population	Climate	HousingCost	HealthCare	Crime	Transportation	Education	Arts	Recreat
String	String	Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer
☐ Enumer...	☑ Enumer...	String Integer Float Double Date Boolean	☐ Enumer...	☐ Enumer...	☐ Enumer...	☐ Enumer...	☐ Enumer...	☐ Enumer...	☐ Enumer...	☐ En
			en_US	en_US	en_US	en_US	en_US	en_US	en_US	en_US
Abilene	TX		521	6200	237	923	4031	2757	996	1405
Akron	OH	660328	575	8138	1656	886	4883	2438	5564	2632
Albany	GA	112402	468	7339	618	970	2531	2560	237	859
Albany-Sche...	NY	835881	476	7908	1431	610	6883	3399	4655	1617
Albuquerque	NM	419700	659	8393	1853	1483	6558	3026	4496	2612
Alexandria	LA	135282	520	5819	640	727	2444	2972	334	1010
Allentown,Be...	NJ	635481	559	8288	621	514	2881	3144	2333	1117

Figure 2.12: ILOG Discovery [Bau02, BHS03, Bau04, Bau06] showing the visualization of a web site as a TreeMap and also the Import Text and SQL data dialogue box.

2.4.3.4 Polaris

Polaris [SH02] is an interface for exploring large multi-dimensional databases (see figure 2.15). It extends the Pivot Table interface as seen in Microsoft Excel. Polaris includes an interface for constructing visual specifications of table-based graphical displays. It also has the ability to generate a precise set of relational queries from the visual specifications. Its visualization technique is based around data cubes and it also supports panning and zooming features. It was written in C++ and OpenGL but is now superseded by the Tableau software system (see section 2.4.2.2).

2.4.3.5 Piccolo

Piccolo [BGM04] is primarily a framework for creating 2D graphics applications using Java, .NET, or .NET pocket version. Its main features are its support for zooming, animation and multiple representations. As such it has been popular in the creation of custom visualization applications.

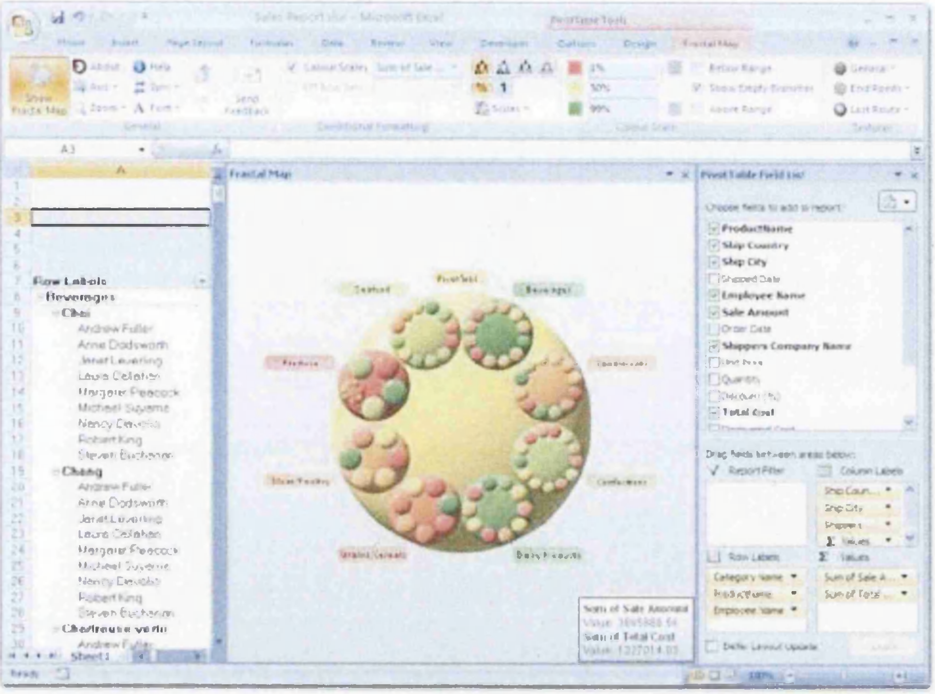


Figure 2.13: Fractal:Edge [Fra08].

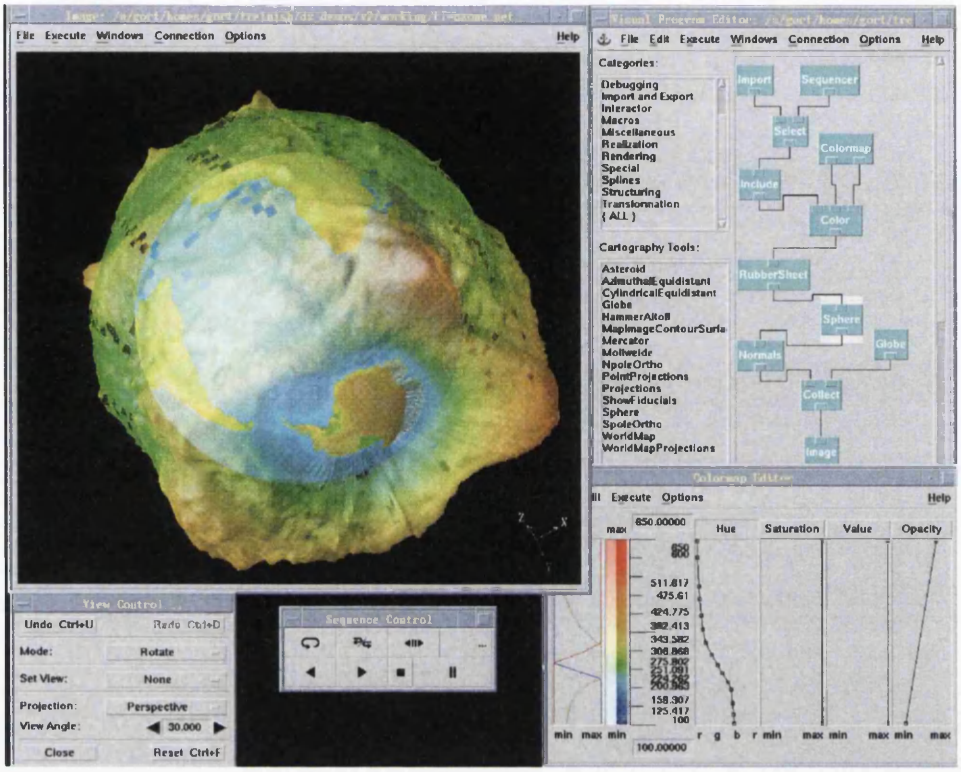
2.4.3.6 Prefuse

Prefuse [HCL05] (see figure 2.16) is an interactive visualization toolkit built in Java. There is also a version which uses Flash and ActionScript called Prefuse Flare. Prefuse has been very popular due to the simple-to-program but sophisticated data modeling, visualization, and interaction facilities it has. It supports data structures for tables, graphs, trees, and a variety of layout and visual encoding techniques. Animation, dynamic queries, integrated search, and database connectivity are also supported. This extensive feature set has seen the toolkit being used in many stand-alone visualization applications and also in the collaborative visualization application ManyEyes (see section 2.4.4.5).

2.4.4 Collaborative Visualization Toolkits

The latest generation of visualization toolkits are focussed on collaborative features such as shared data sources and visualizations with the ability for users to comment on other users' visualization. As such, these visualization toolkits are said to be embracing Web 2.0 features. Please note that in this context we define Web 2.0 based on the definition in [O'R05] and [Hof06], rather than that of the Semantic Web [BLHL01] which has also been used.

Web based collaborative tools such as Many Eyes present a very different user experience challenge from general purpose commercial toolkits such as Tableau or Spotfire. This is because many users will arrive directly at a Many Eyes visualization via a link from an



Mode: Rotate

Set View: None

Projection: Perspective

View Angle: 30.000

Figure 2.14: OpenDX [TBF01] [IBM04].

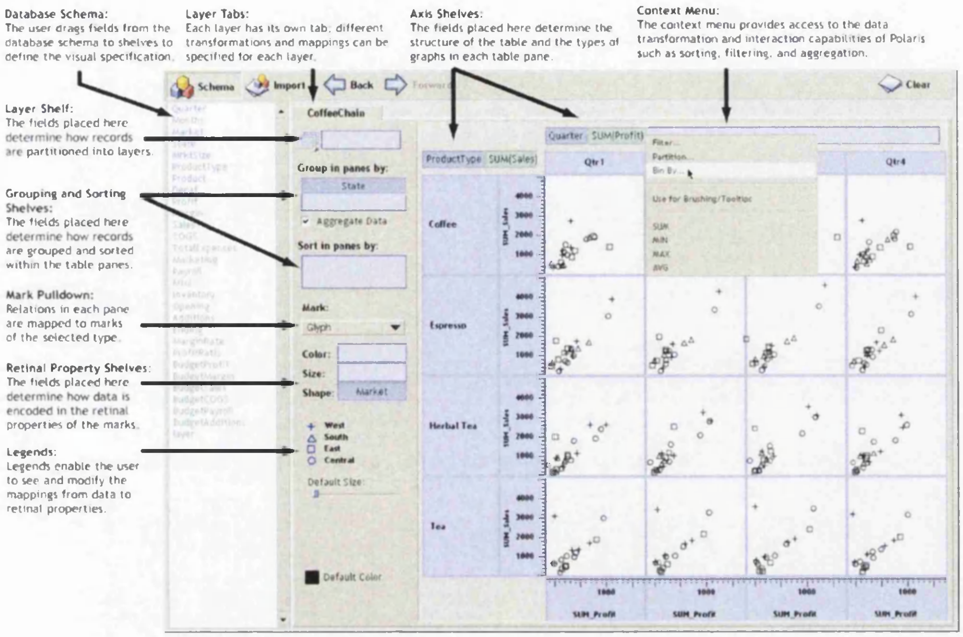
external web site. Therefore the user is often "thrown" into the visualization with little idea of context and most likely no visualization training. If the user does not understand or appreciate the relevance of what they see, they will click away from the visualization. In contrast there are rarely accidental users of commercial systems such as Tableau or Spotfire.

2.4.4.1 DEVise

DEVise [LRB$^+$97] was one of the first applications to allow sharable visualizations. Visual mappings can be customised and shared. There is also an annotation facility. It is however not designed for public accessibility. It runs in a browser but visualizations are relatively static.

2.4.4.2 Dataplace

Dataplace [Dat07b] allows users to visualize basic population statistics in the US, but does not allow users to upload their own data. In this way, it is domain specific.



Database Schema: The user drags fields from the database schema to shelves to define the visual specification.

Layer Tabs: Each layer has its own tab; different transformations and mappings can be specified for each layer.

Axis Shelves: The fields placed here determine the structure of the table and the types of graphs in each table pane.

Context Menu: The context menu provides access to the data transformation and interaction capabilities of Polaris such as sorting, filtering, and aggregation.

Layer Shelf: The fields placed here determine how records are partitioned into layers.

Grouping and Sorting Shelves: The fields placed here determine how records are grouped and sorted within the table panes.

Mark Pulldown: Relations in each pane are mapped to marks of the selected type.

Retinal Property Shelves: The fields placed here determine how data is encoded in the retinal properties of the marks.

Legends: Legends enable the user to see and modify the mappings from data to retinal properties.

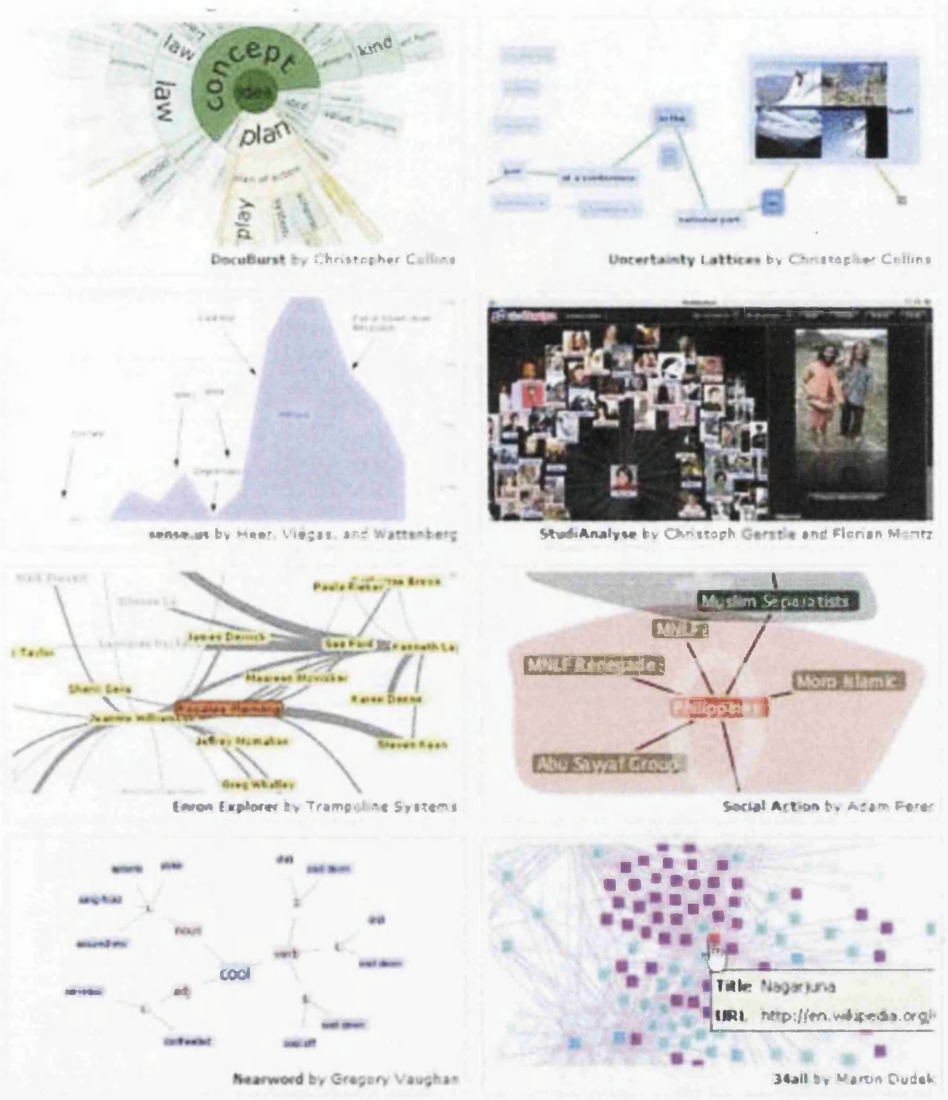
Figure 2.15: Polaris [SH02].

	DEVise	Data places	Data360	Swivel	ManyEyes
Year	1997	2007	2007	2007	2007
Platform	Web	Web	Web	Web	Web
Open Availability	N	Y	Y	Y	Y
Use Own Data	N	Y	Y	Y	Y
Create Vis's	Y	Y	Y	Y	Y
Discuss Vis's	Y	Y	Y	Y	Y
View sharing	Y	Y	Y	Y	Y
Annotation	Y	N	Y	Y	Y
Automatic Vis's	N	N	N	N	Partial
Vis Techniques	4	5	3	4	> 10
Interactive / Static	S	S	S	I	I
Domain Area	GP	US Population Stats	GP	GP	GP
Direct Data Editing	N	N	N	N	N
Data Re-shaping	N	N	N	N	Y

Table 2.4: An overview of Collaborative Visualization Tools.

2.4.4.3 Data360

Data360 [Dat07a] allows users to upload, share and discuss their own datasets with other users. It is a web based application. However, its visualization techniques are limited to fairly static techniques which do not allow users to "drill-down" into the details of the data. Also, users must specify the mappings between source data and visualization manually.



DocuBurst by Christopher Collins

Uncertainty Lattices by Christopher Collins

sense.us by Heer, Viégas, and Wattenberg

StudiAnalyse by Christoph Gerstle and Florian Moritz

Enron Explorer by Trampoline Systems

Social Action by Adam Perer

Nearword by Gregory Vaughan

34all by Martin Dudek

Figure 2.16: Example applications using Prefuse [HCL05].

2.4.4.4 Swivel

Swivel [Swi08] is a similar application to Data360 in its scope and features. However, it tries to encourage users to explore different visualization parameters by automatically creating visualizations. These visualizations are not based on any semantics, but are just created to encourage users to see visualizations from different perspectives. In this way, it has no automatic or semi-automatic visualization features.

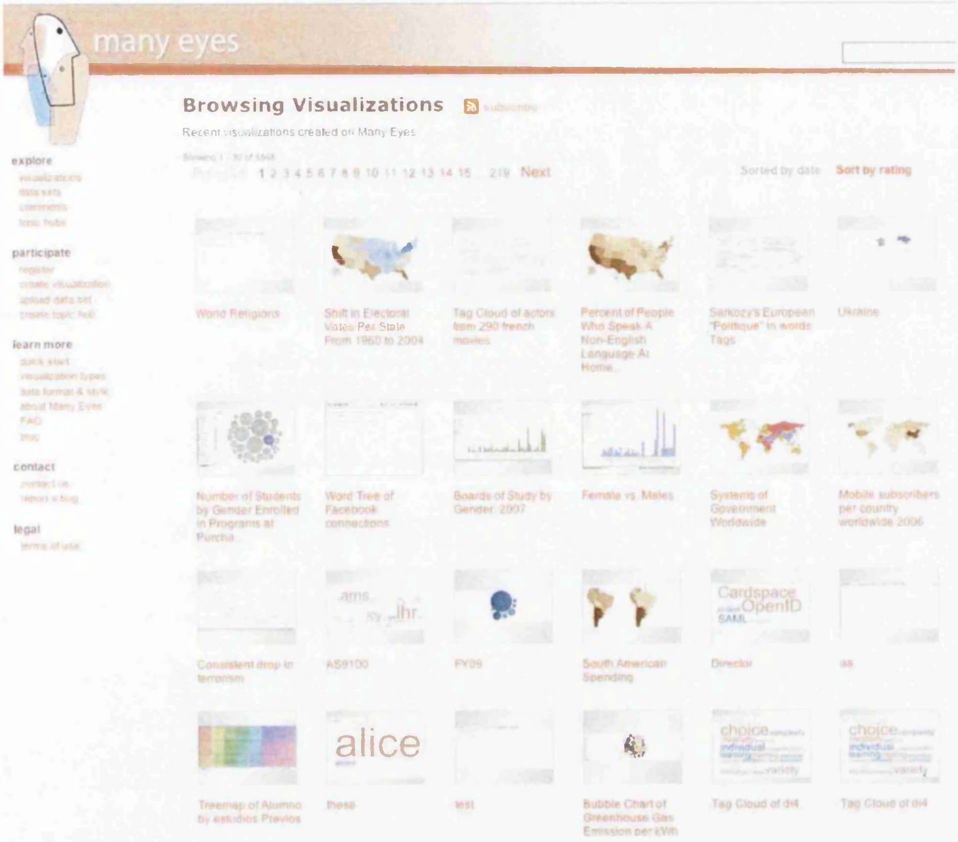

many eyes
Browsing Visualizations
subscribe
Recent visualizations created on Many Eyes
explore
visualizations
data sets
comments
topic hubs
participate
register
create visualization
upload data set
create topic hub
learn more
quick start
visualization types
data format & style
about Many Eyes
FAQ
blog
contact
contact us
report a bug
legal
terms of use
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ... 219 Next
Sorted by date
Sort by rating
World Religions
Shift in Electoral Votes Per State From 1960 to 2004
Tag Cloud of actors from 290 french movies
Percent of People Who Speak A Non-English Language At Home...
Sarkozy's European "Politique" in words Tags
Ukraine
Number of Students by Gender Enrolled in Programs at Purcha...
Word Tree of Facebook connections
Boards of Study by Gender 2007
Female vs. Males
Systems of Government Worldwide
Mobile subscribers per country worldwide 2006
Consistent drop in terrorism
AS9100
FY09
South American Spending
Director
aa
Treemap of Alumno by estudios Previos
these
test
Bubble Chart of Greenhouse Gas Emission per kWh
Tag Cloud of di4
Tag Cloud of di4

Figure 2.17: ManyEyes [VWvH^{+}07].

2.4.4.5 ManyEyes

ManyEyes [VWvH^{+}07] (see figure 2.17) is backed by IBM's research facility and is perhaps the best known and most widely used of the web-based collaborative visualization toolkits (see figure 2.17). The goal of ManyEyes is to support collaboration at a large scale by encouraging a social style of data analysis. This is intended to be a medium to foster discovery and discussion among users. Facilities such as discussion, view sharing and annotation support these goals.

As with Tableau (see section 2.4.2.2), ManyEyes does support semi-automatic visualization. However, again it is only type semantics which are considered and in this way, ManyEyes is also built on a type-constrained system. ManyEyes will allow users to create any visualizations which it deems valid in terms of type mapping between source data entities and target visual artefacts. However, there is no means by which deeper semantics of the data are considered to help create more cognitively useful visualizations.

2.5 Automatic Visualization

Automatic Visualization (or Automatic Presentation) has been a goal of the visualization community for nearly as long as the research area has existed. In order to achieve automatic visualization, there are two distinct areas to consider:

The Interaction Methodology Some visualization toolkits attempt some form of guided or semi-automatic visualization. It is therefore worth considering these interaction methodologies and how they strive for automaticity.

The Automation Algorithm There are different approaches to how a system derives the best way to take a source data entity and visualise it using a target visual artefact.

Further discussion of approaches to automatic visualization is given in [CEH$^+$09].

2.5.1 Interaction Methodology

There are four main interaction methodologies commonly deployed in the visualization process. We describe all four below. The first three were discussed in detail in [PLB$^+$01].

2.5.1.1 Trial and Error methodology

The *trial and error* methodology [YaKSJ07] relies on the interaction between users and the visualization system to derive satisfactory results with minimum assistance from the computer. A large collection of visualization tools (e.g., SpotFire [Ahl96] and ILOG Discovery [BHS03]) support this approach by providing fast rendering and effective exploration of the visual space.

2.5.1.2 Design Galleries methodology

The *design galleries* methodology [MAB$^+$97] is a data-centric approach that relies on limited knowledge of any underlying data model. With some basic knowledge of the application domain and visualization tool (i.e., volume visualization in [MAB$^+$97]), the visualization system automatically selects parameters and generates a set of visualizations, from which users select the most relevant and useful visualizations. This process is repeated until satisfactory visualizations are obtained in a manner resembling the semi-automatic genetic algorithm.

2.5.1.3 Information-assisted methodology

The *information-assisted* methodology relies on some understanding of the underlying model of the data. It extracts more abstract information from the data (e.g., histogram [YMC05], cluster [GDGL07] and topology [WBP07]), and uses it to guide users in their interactive visualization process. The Trial and Error methodology (see section 2.5.1.1) and

Design Galleries methodology (see section 2.5.1.2) involve partial automation, but users' interaction is an essential part of the process.

2.5.1.4 Automatic Visualization methodology

The *automatic visualization* methodology attempts to generate visual representations from data automatically. [Fei85] and [Mac86] first set the agenda for this research direction. [MHS07] presented a set of user interface commands, "Show Me", as part of the user interface of Tableau, providing a number of automated functions in user interaction. In comparison with the other three methodologies, this approach is least studied. We describe this methodology in detail in section 2.5.2.

2.5.2 Automation Algorithm Method

The first significant work in this area was Mackinlay's APT (A Presentation Tool) system [Mac86]. This work codified Jacque Bertin's semiology of graphics [Ber83] as algebraic operators. The APT system was then used to search for a high-quality presentation of information using *expressiveness* and *effectiveness* criteria:

Expressiveness An example being, to display magnitudes without using length of line, use colour or line thickness to represent magnitude.

Effectiveness An example being, to display quantitative information, use geometry rather than colour.

The foundation work of APT was extended by Casner [Cas98] who also considered the level of effort required depending on the visualization task in hand. Additional visualization techniques (including interactivity) were considered by Roth [RKMG94] [GRKM94].

In the scientific visualization community, Senay and Ignatius [SI94] developed a system called Vista. The system decomposed data into partitions which could be separately visualized. Then, a decision-tree is consulted to select visualization choices from expressiveness and effectiveness rules.

Recently, Mackinlay's work has focussed on the user interface aspects of automatic visualization. This work has produced the Tableau visualization toolkit with its ShowMe [MHS07] automatic visualization features (see figure 2.11). Mackinlay's path of work has focussed on how to communicate graphically using computational algebra. This takes the form of a set of rules based on the decades of work in codifying effective means of visualization. Tableau tries to guide the user from data type to chart type by classifying data as: Categorical; or Quantitative; and as: a Measure; or a Dimension. After this, default visualizations are recommended for particular combinations. The user can then employ the "Show Me Alternatives" feature to see alternative visualizations which meet the constraints of the source dataset.

However, there are disadvantages to any constraint-based system:

Rigidity of Constraints The constraints within systems such as Tableau have been developed and refined over many years by some of the best minds within the information visualization community. However, these constraints are rigidly enforced. If there is any problem or questionable assumption which has been made within the foundation of these constraints, it is very difficult or impossible to alter these.

Singularity of Results With a constraint-based system, most commonly, a single result (visualization) is given. If this visualization is indeed what the user requires, then this is fine. However, if there is any part which is not what the user is looking for, then there are no alternatives in which to explore. This singularity is acceptable for knowledgeable visualization users who can use the software to tweak and update parameters. However, for non-expert users presenting only one result is not an effective solution.

Narrowness of Consideration - Type Data Only Constraint-based systems such as Tableau only consider data type information when creating automatic visualizations. That is, whether a data type is numeric or textual, its range (if numeric), its variance, and in some systems, any hierarchy. Any further semantics in the data (e.g., field names) are not considered. Therefore, the scope of semantics of the data considered by the system is limited. This has the effect of limiting the effectiveness of the automatic visualizations produced.

Readability and Edit-ability of Constraints The logic rules stored within a constraint-based system are often in a proprietary format. Also, they are not accessible to users. Therefore, if a user wishes to view the constraints in order to investigate a particular automatic visualization decision, or if a user wishes to edit one of these constraints which is wrong or unsuitable for their application, they have no option. In many circumstances a user will know more about a particular visualization techniques or a source data domain than the system itself. The opportunity to exploit this knowledge is missed.

Although, Tableau with its ShowMe feature contains the latest advances in automatic visualization technology, because of the reasons described above, it remains suited to only trained or expert users. There is also an opportunity to potentially increase the cognitive value of automatically created visualizations by exploiting additional semantics which are not currently considered.

At the same time as Mackinlay's initial APT work, Ahberg was developing visualization tools based on starfield displays with FilmFinder [AS94]. This could only visualise a fixed data source. However, this was developed into work on Dynamic Queries and an implementation called IVEE (Information Visualization and Exploration Environment) [AW95]. This tool was commercialised into the product SpotFire [Ahl96] which has been a successful tool for visual discovery especially within the pharmaceutical industry. SpotFire's features are predominantly focussed around dynamic query filters and starfield displays. It does contain automatic visualization features. However, they are centred around one particular visualization technique and interaction metaphor. Additionally, the automatic visualization functionality is based on data type semantics only. Therefore, SpotFire is constrained by many of the same aspects listed above for Tableau. The company and product SpotFire was recently acquired by Tibco [Tib08].

Mackinlay's work and Ahberg's work in the form of Tableau and Spotfire (respectively), consider *how* to communicate graphically. In contrast, another line of work considers *what* to say by extending the area of computation linguistics into the area of visualization. Feiner's work [Fei85] and Zhou's work [Zho99] used computational linguistics to develop an information assistant by modelling communication with the user. Despite the creation of systems such as APEX, this technique has had little recent work.

A more recent technique for automatic visualization is Visual Data Mining [Kei02] which uses statistical techniques. Again this considers *what* to show rather than *how* to show visualizations. However, the expertise required to create effective visualizations is high. Most visualization users do not possess this level of statistical knowledge. Therefore, it is not suited to non-expert users.

A different focus for automatic visualization has been employed by the scientific visualization community. Rather than concentrating on the mapping of source data entities to target visual artefacts, the focus has been on automation of pipeline stages. In scientific visualization, the number and variety of pipeline stages tends to be greater than in information visualization. Therefore, this provides an opportunity for automation. Fujishiro et al [FTIN97] combined Shneiderman's [Shn96] and Wehrend's [WL90] taxonomies to produce a system called GADGET (Goal-oriented Application Design Guidance for modular visualization Environments). This is a modular visualization environment which employs heuristics to assist the design of visualization pipelines.

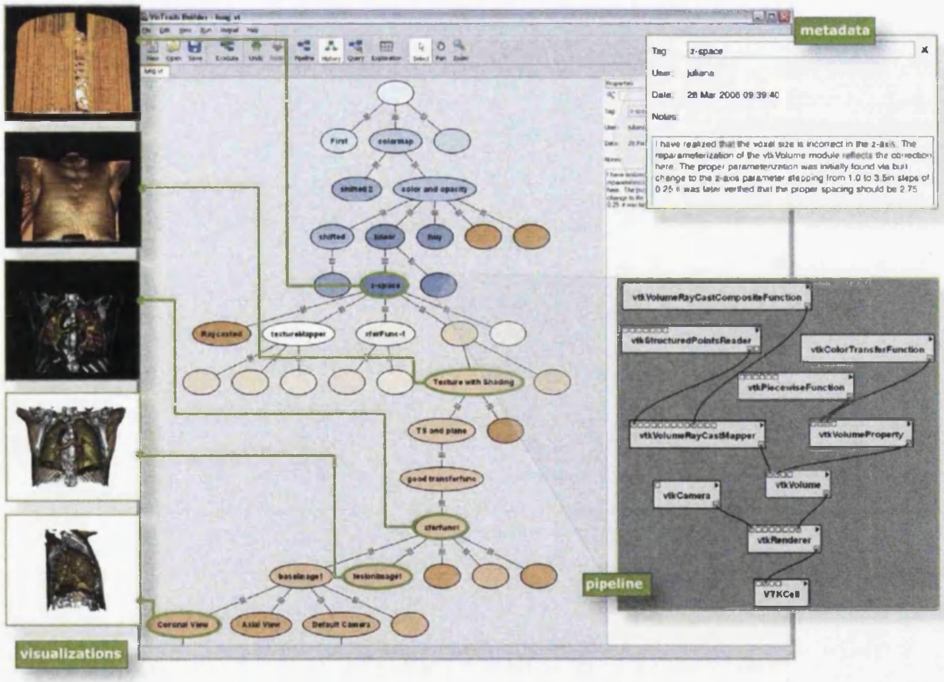

Tag: z-space
User: juliana
Date: 28 Mar 2006 09:39:40
Notes:
I have realized that the voxel size is incorrect in the z-axis. The reparameterization of the vtkVolume module reflects the correction here. The proper parameterization was initially found via brut change to the z-axis parameter stepping from 1.0 to 3.5in steps of 0.25 it was later verified that the proper spacing should be 2.75

Figure 2.18: The VisTrails [ACK$^+$07] [SVK$^+$08] modular visualization environment.

A more recent project based on modular visualization environments for scientific visualization is VisTrails [ACK$^+$07] [SVK$^+$08] (see figure 2.18). This system captures provenance information about pipelines in a database and exploits this to assist with generating new pipelines. In the Query-by-Example feature, the system can be given a fragment of a pipeline and find pipelines in the database which match it. The Visualization-by-Analogy feature allows the changes which occur between two stored pipelines to be applied to a third pipeline. VisComplete [KSC$^+$08] builds on this feature set by allowing users to specify a small fragment of pipeline and querying a large database to find examples of completed pipelines.

The modular visualization environments discussed above have produced some very encouraging results. The most recent work has reduced the level of effort required to produce and experiment with many scientific visualization techniques. Additionally, the ability to reuse previously created effective visualization pipelines is invaluable. In this way, the knowledge and expertise of past users is being captured and reused. However, this knowledge is being captured implicitly via the recording of pipelines which have proved successful. In this way, the reason for any given pipeline being successful is not captured, just the pipeline stages and parameters themselves. Therefore, we can not call these systems truly knowledge-based (although their results may be just as good). This distinction is important when we wish to ascertain why the system has chosen a particular pipeline combination as a recommendation. This knowledge is not stored in the system, but remains in the mind of the user who originally developed that pipeline combination. These factors become increasingly important as the system scales to a larger number of users and a larger database of example pipelines.

2.6 Summary

In this section, we summarise the findings of our investigation into the field of Information Visualization and how it relates to our goal set out in chapter 1 of providing, "general purpose information visualization for non-experts".

Maturity As outlined in section 2.3, the understanding of the perceptions of visualization is clearly well researched, documented and understood. Also, the understanding of other perceptions (e.g., sonification) is steadily developing. Consistent research progress over the last 30 years has led to a large number of well-developed and sophisticated visualization techniques as outlined in section 2.4.

Unification There has been some useful work in unifying many visualization techniques into a single, unified model. The two most common realisations of this are Mackinlay's APT [Mac86] and Baudel's Data-Linear Visualization Algorithms [Bau02].

Generality The range of visualization techniques have moved from domain specific, single visualization technique applications such as FilmFinder [AS94] and Market Map [Sma08] into general purpose, multi-visualization frameworks (e.g., Prefuse [HCL05]) and toolkits (e.g., Tableau [MHS07]).

Automaticity The combination of general purpose frameworks and their multi-technique visualizations has prompted the need for automaticity. Users can present toolkits with a dataset and ask to be given a cognitively useful visualization using the visualization technique of their choosing [MHS07]. However, the level of automaticity provided can only be described as "semi-automatic". Rather than giving a high-quality, potentially definitive answer, features such as Tableau's ShowMe merely help the user in the iterative process of creating a visualization. Part of this problem stems from the fact that ShowMe's reasoning is based on rigid constraint-based systems which only consider the type semantics of the data.

Collaboration With the advent of web-based visualization applications such as ManyEyes [VWvH$^+$07] and Data360 [Dat07a], there has been an opportunity to make the process of a visualization a collaborative task. This has been achieved through features such as discussion forums, annotations and shared views of data and visualizations.

Knowledge Due to the trend towards collaborative visualization, there is an opportunity to exploit the combined knowledge of the community in order to facilitate automatic visualization. One of the main ways in which this knowledge could be exploited is for the creation of automatic visualizations.

Exclusivity Despite the development of modern, integrated visualization environments, whether as native windowing applications or web-based applications, the audience of these tools remains exclusive to the visualization expert or at most the advanced user of the data domain being studied.

For the reasons listed above, the 30 years of development into: understanding the theory of visual perceptions; development of visualization techniques and finally the creation of advanced visualization frameworks and toolkits, is largely confined to members of the visualization community and a select few expert consumers of specific information domains.

Chapter 3

Ontologies and Ontology Mapping

Contents

3.1 Introduction

In this chapter we review the literature associated with Ontologies and Ontology Mapping. The chapter is split into two main sections:

Ontologies The definition of ontologies and related terms; technology standards; editors and helpers; inferencing; and applications.

Ontology Mapping The definition of ontology mapping and related terms; approaches; frameworks and toolkits; and automatic mapping algorithms.

3.2 Ontologies

3.2.1 Definition

The term ontology is a very broad one and is used to cover a wind range of techniques, technologies and applications. Perhaps because of this breadth of applicability, there is often confusion as to the meaning of the term. A common definition is given by Gruber [Gru93]:

> "An ontology is a specification of a conceptualization."

This definition states that an ontology is a description (a formal specification) of concepts and the relationships between them. Although accurate, this definition is a quite abstract. Alternatively, one of the best and most pragmatic definitions of an ontology is provided by Fensel [Fen01]. He describes ontologies by contrasting them with databases. The definition is below:

> "An ontology provides an explicit conceptualization (i.e., meta-information) that describes the semantics of the data. It has a similar function to a database schema. The differences are:
>
> 1. A language for defining ontologies is syntactically and semantically richer than common approaches to databases.
> 2. The information that is described by an ontology consists of semi-structured natural language texts and not tabular information.
> 3. An ontology must be a shared and consensual terminology because it is used for information sharing and exchange.
> 4. An ontology provides a domain theory and not the structure of a data container.
>
> In a nutshell, ontology research is next generation database research where data needs to be shared and not always fit into a simple table. For an elaborated comparison of database schemes and ontologies see [Mee99]."

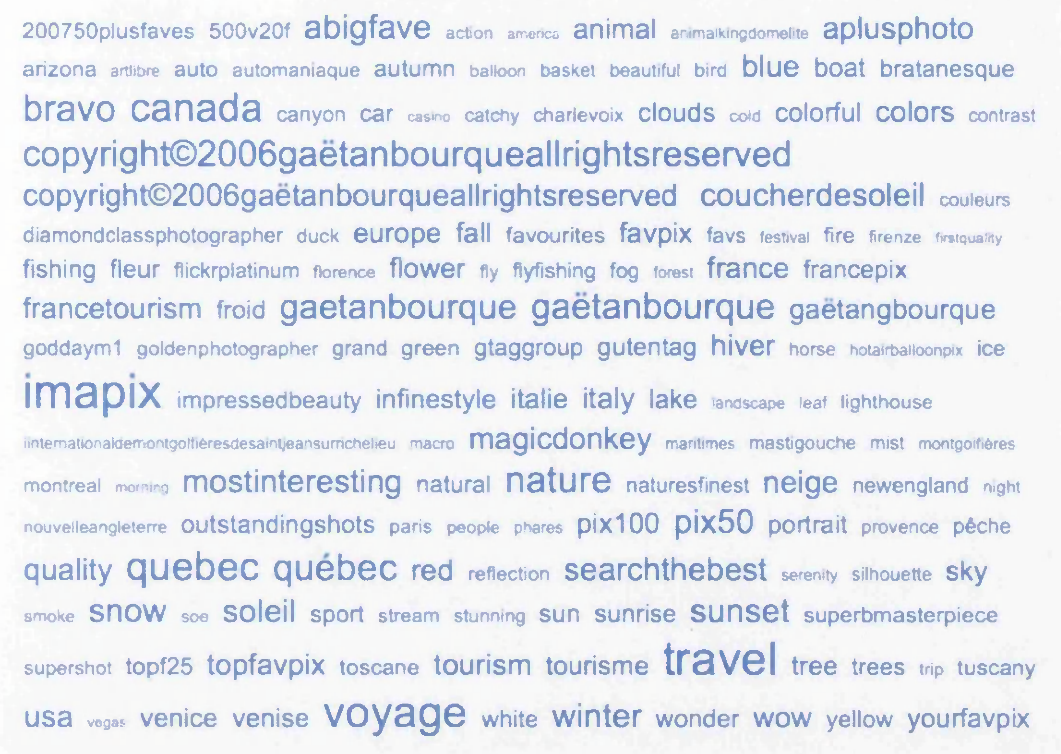
Whichever way one wishes to define an ontology, the main goal is to capture knowledge in a structured manner. The main reasons for doing this are:

- To gain a consensus on terminology.
- To gain a shared understanding of relationships between concepts.
- To facilitate collaboration.
- To allow inferencing on captured knowledge.

There are many other terms which are sometimes confused or used interchangeably with the term ontology:

Taxonomy A taxonomy is a less formal and restricted form of an ontology. It is a hierarchical definition of a restricted set of terminology (and therefore only represents parent-child relationships). Such examples would be the principled classification of species in the animal kingdom [CCMS96], or the Dewey book classification system [Sch99]. An ontology is more sophisticated and formal in that it would define relationships between each concept in the hierarchy. In mathematical terms, a taxonomy is inherently a tree structure whereas an ontology is a graph structure.

Folksonomy A folksonomy is also known as Collaborative Tagging. It is similar to a taxonomy in that its purpose is one of classification. However, the structure is flat because it has no parent-child relationships. The term folksonomy has become popular with the advent of Web 2.0 applications such as the bookmark sharing website, del.icio.us [del08] and the photo-sharing website, flickr [Fli08]. These services allow users to assign tags to items, either their own, or from a list of previous tags. Any relationships which are shown in a folksonomy (see "Tag cloud" below)



200750plusfaves 500v20f abigfave action america animal animalkingdomelite aplusphoto
arizona artlibre auto automaniaque autumn balloon basket beautiful bird blue boat bratanesque
bravo canada canyon car casino catchy charlevoix clouds cold colorful colors contrast
copyright©2006gaëtanbourqueallrightsreserved
copyright©2006gaëtanbourqueallrightsreserved coucherdesoleil couleurs
diamondclassphotographer duck europe fall favourites favpix favs festival fire firenze firstquality
fishing fleur flickrplatinum florence flower fly flyfishing fog forest france francepix
francetourism froid gaetanbourque gaëtanbourque gaëtangbourque
goddaym1 goldenphotographer grand green gtaggroup gutentag hiver horse hotairballoonpix ice
imapix impressedbeauty infinestyle italie italy lake landscape leaf lighthouse
iinternationaldemontgolfièresdesaintjeansurrichelieu macro magicdonkey maritimes mastigouche mist montgolfières
montreal morning mostinteresting natural nature naturesfinest neige newengland night
nouvelleangleterre outstandingshots paris people phares pix100 pix50 portrait provence pêche
quality quebec québec red reflection searchthebest serenity silhouette sky
smoke snow soe soleil sport stream stunning sun sunrise sunset superbmasterpiece
supershot topf25 topfavpix toscane tourism tourisme travel tree trees trip tuscany
usa vegas venice venise voyage white winter wonder wow yellow yourfavpix

Figure 3.1: A tag cloud representing all tags from one user's photos on the photo sharing web application, Flickr [Fli08].

have been derived mathematically (e.g., statistical clustering). Therefore, one could state the a folksonomy is a collaboratively created flat taxonomy.

Tag cloud A tag cloud is a means of displaying the tags associated with an item according to the popularity of those tags. It is often seen on sites such as del.icio.us and flickr (see figure 3.1) and gives an indication as to how the website's community see the item.

3.2.1.1 Ontology Terminology

The following entities are used when discussing ontologies. We use OWL (Web Ontology Language) when naming the kinds of entity. OWL is described in detail in the next subsection.

Concepts (or Classes) These are the main entities in an ontology and are interpreted as a set of individuals in the subject domain. In OWL, they are defined by the `owl:Class` construct.

Instances (or Instances or Object) These are interpreted as an individual of a domain. In OWL, they are defined by the `owl:Thing` construct.

Relations These describe relationships between entities. In OWL, they are defined by the `owl:ObjectProperty` construct and the `owl:DatatypeProperty` construct. It is possible to connect entities by various kinds of relations such as:

- **Specialisation** Specialisation can occur between two concepts (classes), or two properties. It represents inclusion (for example, "car" specialises "vehicle"). In OWL, they are defined by the `rdfs:subClassOf` construct or the `rdfs:subPropertyOf` construct. Specialisation is analogous to inheritance in object-orientated design.

- **Exclusion** Exclusion can occur between two concepts (classes), or two properties. It represents exclusion (for example, "person" excludes "car" because their intersection is empty). In OWL, it is defined by the `owl:disjointWith` construct.

- **Instantiation (or typing)** This is interpreted as membership and can occur between: individuals and classes; property instances and properties; or values and datatypes. For example, "Ford Escort" is an instance of the class "car". In OWL, it is defined by the `rdf:type` construct. Instantiation is analogous to object instances in object-orientated design.

Datatypes These are parts of the domain which have no identity because they specify values rather than individuals. For example, `String` or `Integer` are datatypes.

Data values These are simple values, for example, "royal blue"

3.2.2 Technology Standards

The main technology standard for expressing ontologies is RDF(S)/OWL [SWe04] [De04]. The ontology's schema is defined using RDFS (Resource Description Format Schema) and each ontology instance is defined using RDF (Resource Description Format). The semantics of the relations between concepts in the ontology is defined using OWL (Web Ontology Language). OWL defines common relationships such as `subClassOf`.

An RDF file is made up of statements. A statement (or triple) is made up of three parts: a subject, a predicate and an object. Each object is an identifier or URI (Uniform Resource Identifier). One type of URI is the URL (Uniform Resource Locator) which identifies the location of that resource. However, a URI need not have an online location in order to be a URI. We derive the following example from [Swa02].

An example RDF statement is given below:

```
<http://whsmith.co.uk/>
<urn:xpto/sellsItem>
<http://www.tonyblair.com/books/autobiography/>
```

The first URI is the subject which in this case represents the UK retailer WH Smith. The second URI is the predicate which represents the relationships *sellsItem*. The third URI represents Tony Blair's autobiography. Note that only the first and third URI actually locate a specific resource on the web. The second URI is merely an identifier to the predicate sellsItem which has been defined by the author. There is no constraint on this URI. However, a best practice guide is to not use a third-party URI which the author of the statement does not control. For example, it would not be wise to use the URI:

```
<http://www.tonyblair.com/retail/sellsItem >
```

Unless it is absolutely sure that this URI exists and that the author of the statement knows its true semantics, then we can not guarantee its integrity.

The statement above is actually written in a format called N-Tuples which is a simpler format than RDF, but is more human readable. In RDF the statement would be written:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
    <rdf:Description rdf:about="http:// whsmith.co.uk /">
        <urn:xpto/sellsItem rdf:resource=" http://www.tonyblair.com/books/autobiography/" />
    </rdf:Description>
</rdf:RDF>
```

The RDF above is not as easy to read as the original N-Tuples statement, but it forms the basis of defining statements in RDF.

When we define relationships between resources (or concepts), we can use the semantics of OWL, a knowledge representation language. OWL has three levels of expressiveness:

OWL Lite. This is the least expressive of the three variants and is intended for expressing classification hierarchies (e.g., cardinality). Despite being based on Description Logics, its inferencing capabilities are very limited.

OWL DL. OWL DL is more expressive than OWL Lite and is named after Description Logics (DL). It retains computational completeness, decidability and has a range of practical reasoning algorithms available.

OWL Full. OWL Full was designed to preserve some compatibility with RDF Schema. It has different semantics from OWL Lite and OWL DL. OWL Full can augment the meaning of a predefined RDF or OWL vocabulary. As such, reasoning software for OWL Full is unlikely to be complete.

3.2.3 Formal Definition

We can characterise an ontology formally [ES07] using the following characteristics:

An ontology is a tuple:

$$o \quad = \quad \langle C, I, R, T, V, \leq, \perp, \epsilon, = \rangle$$

such that:
C is the set of concepts (or classes);
I is the set of instances (or individuals);
R is the set of relations;
T is the set of datatypes;
V is the set of values (C, I, R, T, V being pairwise disjoint);
$\leq$ is a relation on (C $\times$ C) $\cup$ (R $\times$ R) $\cup$ (T $\times$ T) called specialisation;

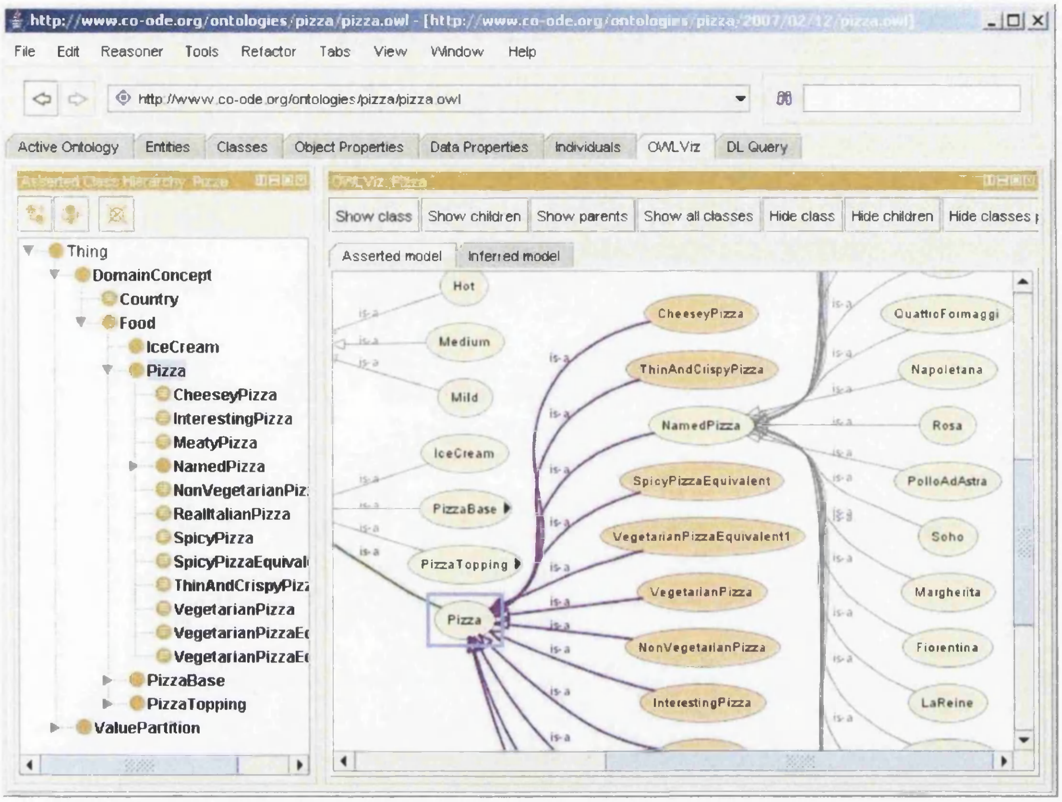


Figure 3.2: The Protege ontology editor [NSD+01].

$\perp$ is a relation on (C $\times$ C) $\cup$ (R $\times$ R) $\cup$ (T $\times$ T) called exclusion;
ϵ is a relation over (I $\times$ C) $\cup$ (V $\times$ T) called instantiation;
= is a relation over I $\times$ R $\times$ (I $\cup$ V) called assignment.

This formal definition of an ontology will be used as the basis for further definitions in this chapter.

3.2.4 Inferencing

One reason for expressing statements in OWL is that we can perform inferencing on the knowledge. Since OWL Lite and OWL DL are made up of Description Logics, we can reason about the knowledge expressed in the concepts and their relationships. Semantic reasoners such as Racer Pro [Rac08], FaCT++ [Hor98] and Pellet [SPG+07] are available for such purposes.

3.2.5 Ontology Editors

As previously mentioned, the readability of RDF(S)/OWL is low for any significant length file, so it is common to employ an editor for such purposes. Protege [NSD+01] (figure 3.2) is an open source, Java-based ontology editor. Ontologies can be exported in a variety of formats such as RDF(S), OWL and XML Schema. Protege allows the creation and editing of classes, properties and their instances. This ability to develop the ontology's schema at the same time as providing instance data helps in the overall process of creating an expressive ontology. Protege also allows the user to specify SWRL (Semantic Web Rule Language) [HPSB$^+$04] rules.

3.2.6 Fuzzy Ontologies

Traditional (or crisp) ontologies are less suitable for domains where concepts and the relationships between them have imprecise definitions. Fuzzy ontologies aim to enrich the relationships between concepts in traditional ontologies with fuzzy weights (i.e., numeric values between 0 and 1). Calegari et al [CS08] defined dynamic fuzzy ontologies in order to model knowledge in creative environments. The ontologies are dynamic in that they adapt over time to the environment in which they are used.

Parry [Par04] describes the use of fuzzy ontologies to improve query results for different users or groups of an ontology from a document retrieval system. By assigning a value to each query term for different users and groups, then the quality of the retrieved documents is predicted to be higher.

In all applications of fuzzy ontologies, we see two common themes:

1. The fuzzy aspect is used to augment traditional ontologies rather than as a replacement for the "crisp" relationships.
2. The fuzzy values are machine-generated and computed. They are not intended for humans to read (either by the original ontology creator, or subsequent users).

3.2.7 Application: Schema Integration

Interoperability between organisations from an information perspective is relatively common, whereas interoperability from a knowledge perspective is less common [SR05]. The difference is that with knowledge integration, it is not just information, but semantics which are exchanged also. For example, in the integration of two schemas about houses between an estate-agent and a local council, we would not only exchange details about houses (i.e. information), we would also exchange the fact that houses have a number of bedrooms which is related to the council-tax band (i.e. knowledge). There are many tools, both commercial and public domain, which can aid the process of information integration (e.g., Altova's MapForce [Alt08a]). Typically the technologies involved are relational databases or even simple flat file structures. The integration difficulty comes from matching the schemas of both organisations. The tools available for this typically have a capable and mature set of facilities to help with this process.

When the schema integration of knowledge-based repositories is concerned, the task for organisations is more difficult. The complex relationship between concepts and the specific semantics of these relationships needs to be taken into account. The ultimate goal is to have automatic integration between knowledge repositories in organisations. However, in practice this is unlikely to be achieved due to the complex nature of these heterogeneous knowledge stores.

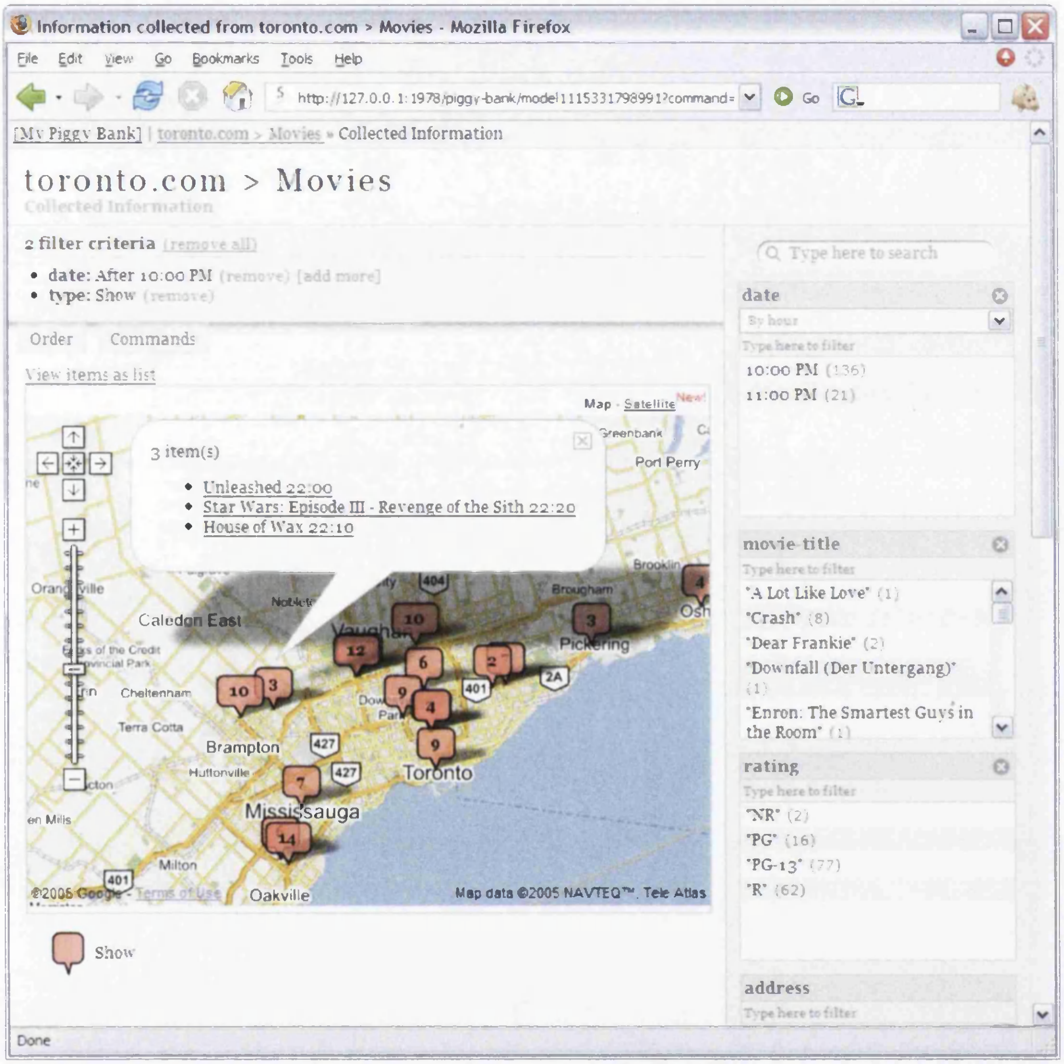
3.2.8 Application: Knowledge Representation

The traditional origins of Knowledge Representation (KR) are from the Artificial Intelligence community. Attempts have been made to transfer these concepts and technologies to the world wide web with technologies such as DAML+OIL and later OWL. Knowledge Representation is concerned with the use of symbols for the representation of a "domain of discourse". This provides a language which can be talked about together with functions which can be applied to allow inferences (formal reasoning) about the objects in the "domain of discourse". A large number of ontologies have been created within different subject domains. An area which is particularly rich with ontologies is the biomedical domain with repositories available publicly such as the OBO Foundry [Fou02], and NCBO BioPortal [NCB02].

3.2.9 Application: The Semantic Web

It has been almost ten years since Tim Berners-Lee proposed the concept of the Semantic Web [BLHL01]. The philosophy is one of evolving the original World Wide Web from a human presentation-orientated paradigm to computer-aided processing paradigm. The principle idea is to annotate web data with machine-processable descriptions, therefore enabling software agents to compute and mediate between users needs and the information sources [Fen01]. The method to model, represent and convey the machine-processable descriptions of the data between information communities is based on ontologies. Information communities can characterise their documents according to the ontologies which are made publicly accessible and sharable, allowing them to describe as best possible the intended semantics of the document content. This would then allow the whole world wide web to be loosely characterised as a database system. This would allow for a degree of interoperability and reasoning not possible with the first-generation, presentation focussed web.

However, this vision has largely failed in its realisation, especially as far as the general web population is concerned. There has been a large amount of academic work produced, but this has not materialised on the mainstream web. There have been a number of commentaries and analyses on the reasons for this. Some have discussed how RDF should not be presented as a graph [msK06]. Others have heavily criticised ontologies [Shi05]. However, one of the main reasons for the lack of take-up of the Semantic Web seems to be peoples' unwillingness to go to the effort of "semantically enriching" their data when there is no clear and immediate benefit [Car07]. Like-wise, developers have not been motivated to create semantically capable applications due to a lack of available semantically rich data. This is a classic "chicken and egg" scenario.


Information collected from toronto.com > Movies - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://127.0.0.1:1978/piggy-bank/model1115331798991?command=
Go
Map - Satellite New!
Greenbank
Port Perry
Brooklin
Orangeville
Caledon East
Nobleton
Vaughan
Pickering
Cheltenham
Terra Cotta
Brampton
Huttonville
Toronto
Mississauga
Milton
Oakville
en Mills
401
427
404
2A
12
10
6
3
9
4
7
14
2
©2005 Google - Terms of Use
Map data ©2005 NAVTEQ™, Tele Atlas
Show
Done

[My Piggy Bank] | toronto.com > Movies » Collected Information

toronto.com > Movies

Collected Information

2 filter criteria (remove all)

- date: After 10:00 PM (remove) [add more]
- type: Show (remove)

Order Commands

View items as list

3 item(s)

- Unleashed 22:00
- Star Wars: Episode III - Revenge of the Sith 22:20
- House of Wax 22:10

Type here to search

date

By hour

Type here to filter

10:00 PM (136)
11:00 PM (21)

movie-title

Type here to filter

"A Lot Like Love" (1)
"Crash" (8)
"Dear Frankie" (2)
"Downfall (Der Untergang)" (1)
"Enron: The Smartest Guys in the Room" (1)

rating

Type here to filter

"NR" (2)
"PG" (16)
"PG-13" (77)
"R" (62)

address

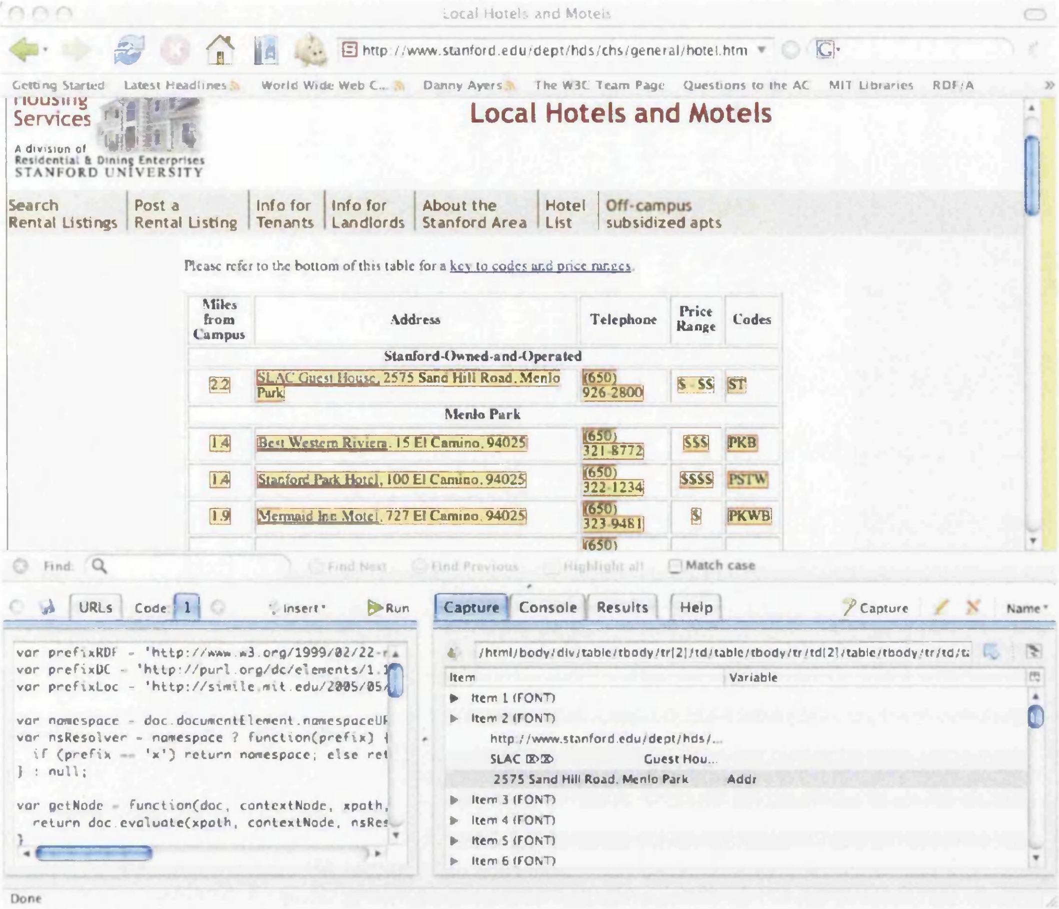
Type here to filter

Figure 3.3: The Piggybank semantic store [HMK07] from the SIMILE project.

3.2.9.1 The SIMILE Project

An interesting and successful project has emerged from the SIMILE (Semantic Interoperability of Metadata and Information in unLike Environments) team, led by David Karger at MIT. The team has developed a suite of tools which is focussed on providing pragmatic solutions to the current generation of web sites. While the tools have not currently gained mainstream adoption, they have served as a very useful demonstration as to the potential of the semantic web amongst a wider community than just those focussing on researching the topic. Below we describe some key tools within the SIMILE stack.

Piggy Bank Piggy Bank [HMK07] is a Firefox web browser extension (figure 3.3) which allows users to extract data from different web sites and mix them together (creating a so called "mashup platform"). The extracted information can be stored locally or shared with other users and can be searched at a later stage.



Local Hotels and Motels

Please refer to the bottom of this table for a key to codes and price ranges.

Miles from Campus	Address	Telephone	Price Range	Codes
	Stanford-Owned-and-Operated			
2.2	SLAC Guest House, 2575 Sand Hill Road, Menlo Park	(650) 926-2800	$ - $$	ST
	Menlo Park			
1.4	Best Western Riviera, 15 El Camino, 94025	(650) 321-8772	$$$	PKB
1.4	Stanford Park Hotel, 100 El Camino, 94025	(650) 322-1234	$$$$	PSTW
1.9	Mermaid Inn Motel, 727 El Camino, 94025	(650) 323-9481	$	PKWB
		(650)		

```
var prefixRDF = 'http://www.w3.org/1999/02/22-r
var prefixDC = 'http://purl.org/dc/elements/1.1
var prefixLoc = 'http://simile.mit.edu/2005/05/

var namespace = doc.documentElement.namespaceUR
var nsResolver = namespace ? function(prefix) {
  if (prefix == 'x') return namespace; else ret
} : null;

var getNode = function(doc, contextNode, xpath,
  return doc.evaluate(xpath, contextNode, nsRes
}
```

/html/body/div/table/tbody/tr[2]/td/table/tbody/tr/td[2]/table/tbody/tr/td/t

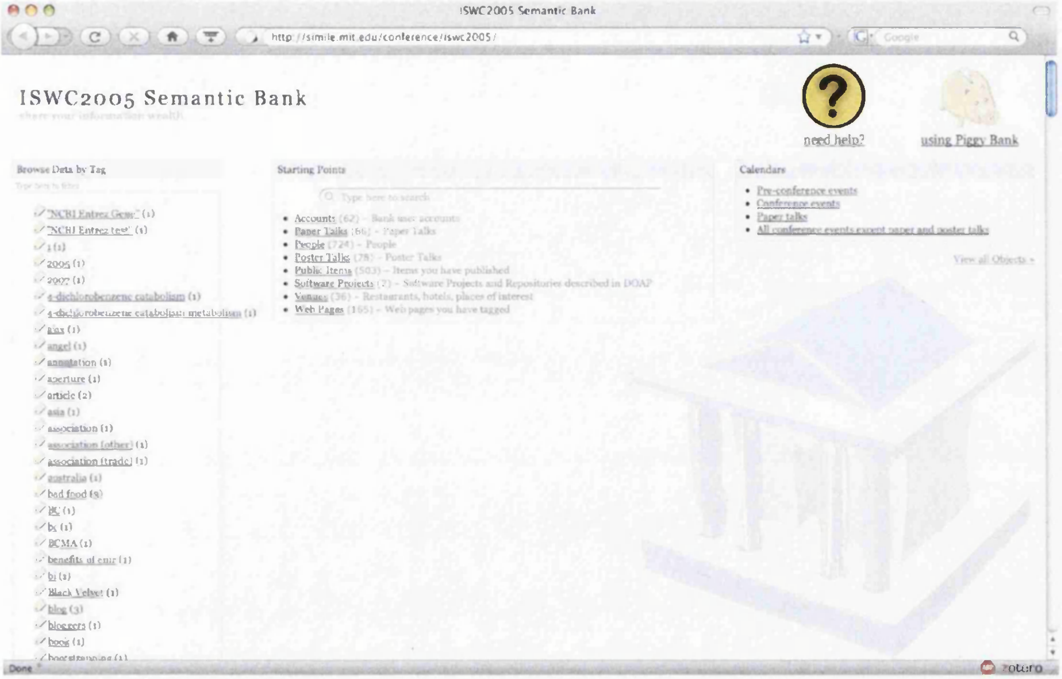
Item	Variable
▶ Item 1 (FONT)	
▶ Item 2 (FONT)	
http://www.stanford.edu/dept/hds/...	
SLAC ⌦⌦	Guest Hou...
2575 Sand Hill Road, Menlo Park	Addr
▶ Item 3 (FONT)	
▶ Item 4 (FONT)	
▶ Item 5 (FONT)	
▶ Item 6 (FONT)	

Figure 3.4: The Solvent screen scraper [HMK07] being used within together with Piggybank within the Firefox web browser.

Solvent Solvent [HMK07] is another Firefox extension which works alongside Piggy Bank to create screen scrapers (figure 3.4). These screen scrapers can extract information from a web page, allowing the data to be captured in a semantically richer format (i.e., RDF in this case).

Semantic Bank Semantic Bank [HMK07] enables users to persist, share and publish data collected by individuals, groups or communities (figure 3.5). It is effectively the server-side component of Piggy Bank.

Exhibit Exhibit [HKM07] allows users to create data-rich web pages (figure 3.6) without doing any server-side programming (e.g., using SQL, ASP, PHP etc). Effectively, Exhibit is a client-side web application which performs similar functions to a three-tier web application but the user only need specify a data file (the "database") and a HTML file (the presentation). The benefit of this approach is that it lowers the barrier-to-entry for users to create data rich web pages - it is not necessary to learn and set-up complex web server and database environments. A side-effect of this is that the data can also be easily presented in a machine-readable, semantically rich format such as RDF or JSON (JavaScript Object Notation) [Cro06] [JZY08].



ISWC2005 Semantic Bank

http://simile.mit.edu/conference/iswc2005/

ISWC2005 Semantic Bank

share your information wealth

need help?

using Piggy Bank

Browse Data by Tag

Type here to filter

- "NCBI Entrez Gene" (1)
- "NCBI Entrez test" (1)
- 1 (1)
- 2005 (1)
- 2007 (1)
- 4-dichlorobenzene catabolism (1)
- 4-dichlorobenzene catabolism metabolism (1)
- ajax (1)
- angel (1)
- annotation (1)
- aperture (1)
- article (2)
- asia (1)
- association (1)
- association (other) (1)
- association (trade) (1)
- australia (1)
- bad food (9)
- BC (1)
- bc (1)
- BCMA (1)
- benefits of emr (1)
- bi (1)
- Black Velvet (1)
- blog (3)
- bloggers (1)
- book (1)
- bootstrapping (1)

Starting Points

Type here to search

- Accounts (62) – Bank user accounts
- Paper Talks (66) – Paper Talks
- People (724) – People
- Poster Talks (78) – Poster Talks
- Public Items (503) – Items you have published
- Software Projects (7) – Software Projects and Repositories described in DOAP
- Venues (36) – Restaurants, hotels, places of interest
- Web Pages (165) – Web pages you have tagged

Calendars

- Pre-conference events
- Conference events
- Paper talks
- All conference events except paper and poster talks

View all Objects »

Done

Figure 3.5: An example of a Semantic Bank [HMK07] (the server-side component of Piggy Bank) from the ISWC 2005 conference.

Potluck Potluck [HMK08] allows casual users to mix (mashup) data using a drag and drop interface for merging fields (figure 3.7). Users can also use the faceted browsing paradigm [OD06] to allow data alignment and syntactic data cleansing.

Timeline Timeline [Huy08] allows the visualization of time-based events (figure 3.8). It is totally client-based and is built upon DHTML and AJAX technologies. Timeline is populated using XML or JSON or data sources.

Timeplot Timeplot [Maz08] allows users to plot time series data with overlays of events (figure 3.9). It supports the same dataformats as Timeline (above).

Together, these tools provide a pragmatic, working demonstration of what the semantic web can do. A useful scenario is to consider a student who wishes to see visually a timeline of the publications of a particular academic. The steps for doing this are outlined below:

1. The student locates the academic's webpage.

2. The student sees that the academic's list of publications is presented as semantically enriched information using an Exhibit.

3. The student can see the information presented tabularly using Exhibit. They can sort and filter the information. However, the student sees that there are many publications and they wish to see the date of publication for each one presented graphically. This facility is not immediately available from the academic's webpage.

4. The student is aware of a Timeline component which can present information

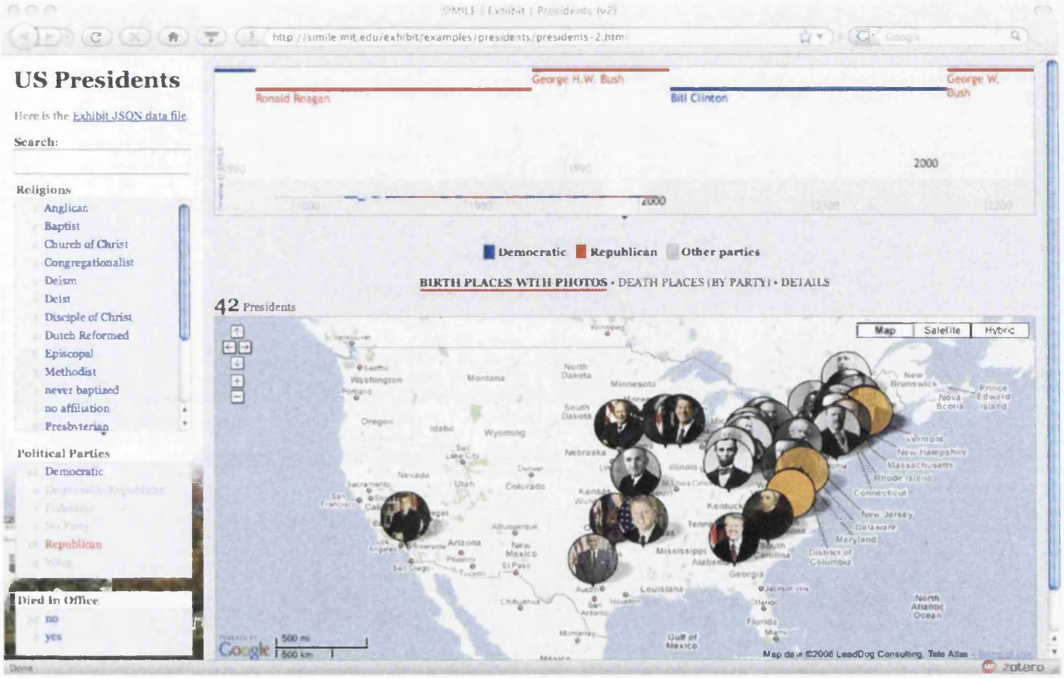
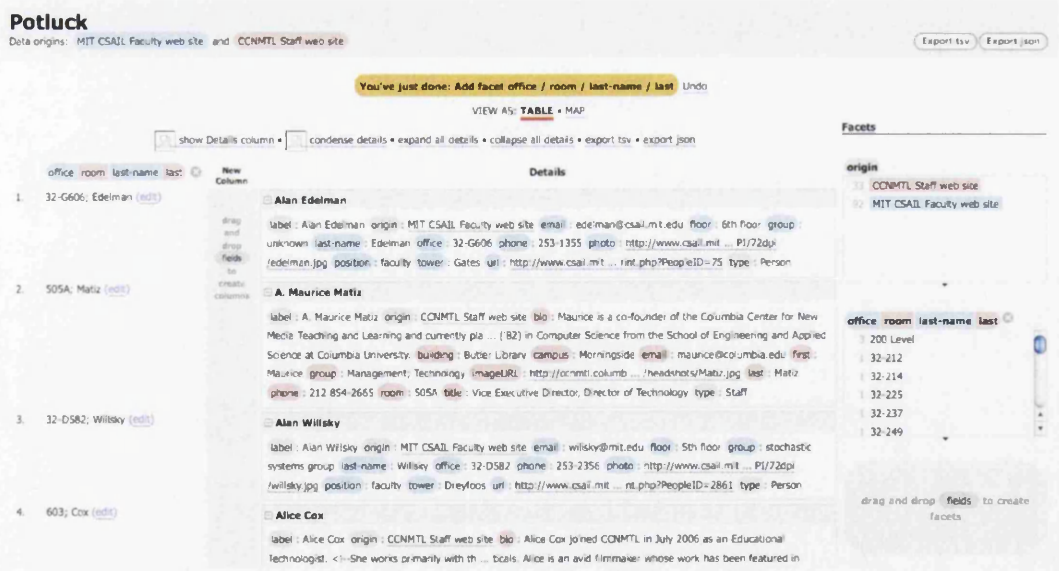


Figure 3.6: The Exhibit framework [HKM07] being used to present information on US Presidents using the Time Line component and Google Maps.


Potluck

Data origins: MIT CSAIL Faculty web site and CCNMTL Staff web site

Export tsv Export json

You've just done: Add facet office / room / last-name / last Undo

VIEW AS: TABLE • MAP

□ show Details column • □ condense details • expand all details • collapse all details • export tsv • export json

office room last-name last	New Column	Details
1. 32-G606; Edelman (edit)	drag and drop fields to create columns	**Alan Edelman** label : Alan Edelman origin : MIT CSAIL Faculty web site email : edelman@csail.mit.edu floor : 6th floor group : unknown last-name : Edelman office : 32-G606 phone : 253-1355 photo : http://www.csail.mit ... PI/72dpi /edelman.jpg position : faculty tower : Gates url : http://www.csail.mit ... rint.php?PeopleID=75 type : Person
2. 505A; Matiz (edit)		**A. Maurice Matiz** label : A. Maurice Matiz origin : CCNMTL Staff web site bio : Maurice is a co-founder of the Columbia Center for New Media Teaching and Learning and currently pla ... ('82) in Computer Science from the School of Engineering and Applied Science at Columbia University. building : Butler Library campus : Morningside email : maurice@columbia.edu first : Maurice group : Management; Technology imageURL : http://ccnmtl.columb ... /headshots/Matiz.jpg last : Matiz phone : 212 854-2665 room : 505A title : Vice Executive Director, Director of Technology type : Staff
3. 32-D582; Willsky (edit)		**Alan Willsky** label : Alan Willsky origin : MIT CSAIL Faculty web site email : willsky@mit.edu floor : 5th floor group : stochastic systems group last-name : Willsky office : 32-D582 phone : 253-2356 photo : http://www.csail.mit ... PI/72dpi /willsky.jpg position : faculty tower : Dreyfoos url : http://www.csail.mit ... nt.php?PeopleID=2861 type : Person
4. 603; Cox (edit)		**Alice Cox** label : Alice Cox origin : CCNMTL Staff web site bio : Alice Cox joined CCNMTL in July 2006 as an Educational Technologist. <!--She works primarily with th ... ocals. Alice is an avid filmmaker whose work has been featured in

Facets

origin
33 CCNMTL Staff web site
92 MIT CSAIL Faculty web site

office room last-name last
3 200 Level
1 32-212
1 32-214
1 32-225
1 32-237
1 32-249

drag and drop fields to create facets

Figure 3.7: The Potluck data mashup tool [HMK08] being used to merge or align two datasource: from the “MIT CSAIL Faculty web site”; and from the “CCNMTL Staff web site”.

chronologically. This component takes as its input: a set of records and a mapping between the fields of that recordset and the fields which the Timeline component uses.

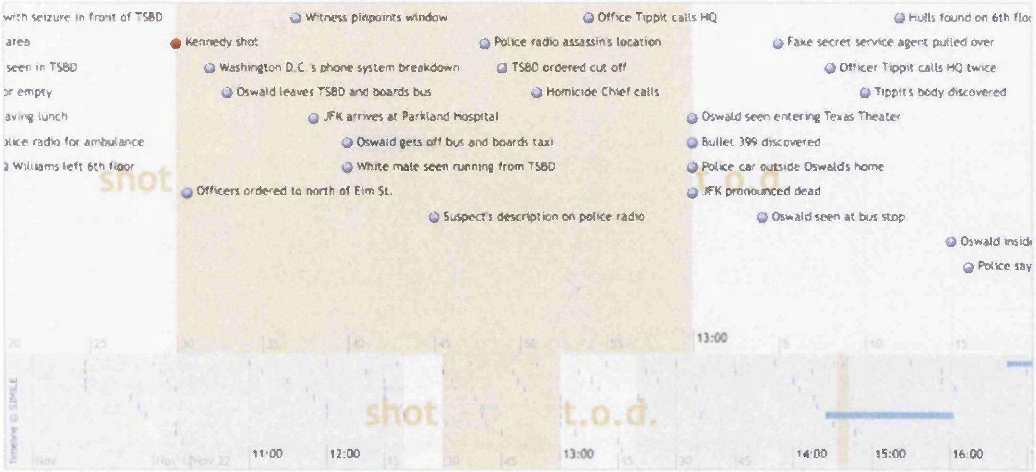
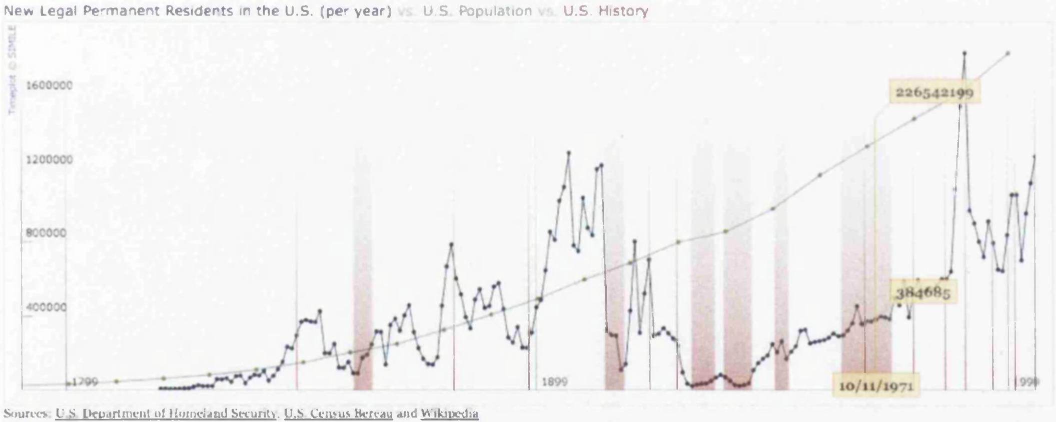

with seizure in front of TSBD
area
seen in TSBD
or empty
aving lunch
olice radio for ambulance
Williams left 6th floor
shot
Kennedy shot
Washington D.C.'s phone system breakdown
Oswald leaves TSBD and boards bus
Officers ordered to north of Elm St.
Witness pinpoints window
JFK arrives at Parkland Hospital
Oswald gets off bus and boards taxi
White male seen running from TSBD
Suspect's description on police radio
Office Tippit calls HQ
Police radio assassin's location
TSBD ordered cut off
Homicide Chief calls
Oswald seen entering Texas Theater
Bullet 399 discovered
Police car outside Oswald's home
JFK pronounced dead
t.o.d.
Oswald seen at bus stop
Hulls found on 6th flo
Fake secret service agent pulled over
Officer Tippit calls HQ twice
Tippit's body discovered
Oswald insid
Police say
13:00
Timeline © SIMILE
shot
t.o.d.
11:00
12:00
13:00
14:00
15:00
16:00

Figure 3.8: A Timeline component [Huy08] being used to visualise the series of events before, during and after the assassination of President Kennedy.



Sources: U.S. Department of Homeland Security, U.S. Census Bereau and Wikipedia

Figure 3.9: A Timeplot component [Maz08] being used to visualise "New Legal Permanent Residents in the U.S. (per year) vs. U.S. Population vs. U.S. History".

This input must be provided as a pre-formatted XML or JSON file.

5. The student decides that it is worth the necessary effort and creates an XML file of the records. The student must also decide which fields from the list of publications should be mapped to which field of the Timeline component.
6. After performing these tasks, the user is shown the publications displayed visually on the Timeline.

We can see that in this scenario, the student is able to gain useful insight into the data through the visual Timeline created. However, there are three problems:

1. The user must decide on the best mapping between the data in the source web page (whether it is semantically enriched via an Exhibit, or extracted via Sifter into

Piggybank) and the visual representation artefacts in the Timeline display.

2. The user must manually create the data mappings chosen between the source web page and Timeline.

3. Timeline and Timeplot are the only available visualization techniques in this "suite"' of visualization techniques. Other information visualization techniques such as TreeMaps or Parallel Coordinates are not available.

Clearly, these 3 issues would present too great a problem for most users. A particularly determined user with the necessary computer science skills may perform the necessary integration work, however, most users would not. This clearly represents a wasted opportunity for users to exploit information visualization techniques.

3.2.10 Application: Ontologies of Visualization

Shahar and Cheng [SC98] developed an ontology and associated methodology for the visualization of temporal abstractions. The KNAVE system (Knowledge-based Navigation of Abstractions for Visualization and Explanation) is specific to the task of interpretation, summarisation, visualization, explanation, and interactive navigation of time-oriented data and interval-based concepts. The system maps domain specific concepts to domain independent concepts. Users are able to query the system for domain specific temporal-abstractions and thus change the focus of the visualization. The approach of using domain experts and the process of mapping between domain-specific and domain-independent concepts is a powerful idea for capturing and reusing knowledge.

An initiative to construct an ontology of visualization is discussed in [BDD04], [DBD04], [Duk04], and [DBDH05]. The scope of the work was to create an ontology for both information and scientific visualization which encompasses: task and use; representation; process; and data. This initiative although successful was not focussed on any particular application. Rather its focus was to promote a shared terminology, classification and understanding of the visualization domain. For this reason, the ontology has not been used as a core part of any system.

Rhodes et al. [RKR06] created VisIOn (Interactive Visualization Ontology). VisIOn is a web-based system which can categorise and store information about Software Visualization systems. A large amount of work was performed to reconcile various categories of terminology from previous taxonomy work. In this sense, VisIOn is a system which utilises an ontology which has been designed for a specific application domain.

Xie et al. [XZS06] created an ontology focussed on scientific visualization which was structured by: Dataset; Filter; Device; and Pipeline. An application centred around the ontology was produced. However, its main focus was on communicating and sharing, quick learning and knowledge reuse.

Shu et al. [SAR08] have designed a visualization ontology based on E-notation [Bro93]. The ontology aims to provide semantics for discovery of visualization services and as such it considers all previous work in developing visualization taxonomies and ontologies. However, this work does not present a system which uses the ontology for the purpose of automatic visualization. Instead, it focusses on service discovery, utilisation and the use of "grid" resources.

3.3 Ontology Mapping

In this section we describe the field of Ontology Mapping. We provide a definition and a motivation before discussing various forms of heterogeneity and giving formal definitions of the ontology mapping process. We then discuss the various approaches for matching ontologies together with their advantages and disadvantages.

3.3.1 Definition

When we consider open and evolving systems which use ontologies, different parties often use different ontologies. These different ontologies are used due to the varying nature of each system and heterogeneity cannot be avoided. The different actors involved in a system have different motivations, interests and habits. They may also use different tools and represent their knowledge at different levels of detail. The act of using an ontology does not therefore reduce the heterogeneity of systems, it raises the problem of heterogeneity to a higher level of abstraction. The process of finding correspondences between different ontologies is called *ontology matching*. Therefore ontology matching can be seen as a solution to the problem of semantic heterogeneity which is faced by open and distributed computer systems. The purpose of ontology matching is to find the correspondence between semantically related entities of different ontologies. These correspondences may represent equivalence as well as more complex relations such as consequence, subsumption or disjointness.

The result of an ontology matching process is called an *alignment*. An alignment can be used for tasks such as ontology merging, query answering, data translation and the browsing of the semantic web. Matching ontologies therefore allows the knowledge represented in matched ontologies to interoperate. There are many approaches to ontology matching which take advantage of different properties of ontologies, for example: structure, data instances, semantics, or labels. These techniques originate from a variety of fields such as statistics and data analysis, machine learning, automated reasoning and linguistics [ES07].

3.3.2 Types of Heterogeneity

Traditionally, the goal of matching ontologies is to reduce the heterogeneity between them. There have been many detailed studies into the types of heterogeneity and associated classifications have been produced: [Euz01] (focussing on interoperability levels); [Kle01] (focussing on mismatches); [HPS04]; [Cor04]; [BEE$^+$04]; and [GG04].

The most common types of heterogeneity are:

Terminological heterogeneity This heterogeneity occurs because ontologies refer to the same entity using different names. This may because a different natural language is being used (e.g., English vs. French), or because of the use of synonyms.

Syntactic heterogeneity This heterogeneity occurs when two ontologies are expressed in different ontology languages (e.g., one ontology is expressed in F-Logic and the other in OWL). Translation between ontologies with this type of heterogeneity is usually very accurate because there are often well defined equivalencies between the ontology languages involved [ES03].

Semantic heterogeneity This is also called conceptual heterogeneity or logical mismatch. It occurs when there are differences in modelling the same domain of interest due to different axioms for defining concepts, or due to the use of totally different concepts. A good illustration of the reasons for conceptual differences is given in [BBG01] and [ES07] by way of describing the ontological modelling of a geographic map.

Difference in coverage This occurs when two ontologies describe different (possibly overlapping) regions of the world at the same level of detail and from a unique perspective.

Difference in granularity This occurs when two ontologies describe the same region of the world from the same perspective but at different levels of detail. Geographic maps with different scales are an example of a difference in granularity.

Difference in perspective This occurs when two ontologies describe the same region of the world at the same level of detail, but from different perspectives. This might occur with geographic maps creates for different purposes, e.g., a political map and a terrain map.

Semiotic heterogeneity This is also called pragmatic heterogeneity. It is concerned with how entities are interpreted by people. Entities are often interpreted by humans in different ways depending on their context. The intended use of entities has a great impact on their interpretation. Therefore matching entities which are not meant to be used in the same context is often error-prone.

Usually several different types of heterogeneity occur together to different degrees. Later in this section, we will discuss approaches which deal with these different types of heterogeneity, both individually and together.

3.3.3 Terminology

Here we describe a common vocabulary for describe ontology mapping:

Ontology Matching This is the process of finding correspondences between entities in different ontologies.

Ontology Mapping This is essentially the same as Ontology Matching. However, "mapping" can be used to describe both the action of matching the ontologies (i.e., as a verb) and as the end result of a matching process (i.e., as a noun). In the remainder of this thesis we use the term mapping both as the action (process) and as the end result. The meaning will be clear from the context.

Ontology Merging This is the process of creating a new ontology from two source ontologies. The new, merged ontology should contain the knowledge represented in the original ontologies. The original ontologies remain unchanged.

Ontology Transformation This is the process of expressing the entities of one ontology with respect to the entities of another ontology.

Ontology Reconciliation This is the process of harmonising the content of two ontologies often as a pre-cursor to merging them. It requires changes to one ontology (and often both as a co-evolution). This subject is discussed in [HPS04], but will not be investigated further here.

Alignment An alignment is the result of the matching process. It is a set of correspondences between two ontologies.

Correspondence This is the relation holding (according to a particular mapping algorithm) between entities in different ontologies. It is sometimes also referred to as a "mapping".

3.3.4 Formal Definitions

In this subsection, we formally define the different stages of the mapping process. We use the original formal definition of an ontology (section 3.2.3) as a basis for this. These definitions are based on those given in [ES07].

3.3.4.1 The Matching Process

The matching process determines the alignment A' between a pair of ontologies o and o'. Other parameters can extend the definition of the matching process:

1. the use of an input alignment A.
2. matching parameters, p. For examples, weights and thresholds.
3. external resources to be used by the matching process. For example, domain thesauri.

The matching process can be seen as a function f which, from a pair of ontologies to match o and o', an input alignment A, a set of parameters p and a set of oracles and resources r, returns an alignment A' between these ontologies:

$$A' = f(o, o', A, p, r)$$

3.3.4.2 Entity Language

It can be desirable to have a different language for identifying the matched entities from the language defining the entities themselves. We use an *entity language* for expressing those entities that will be put in correspondence by matching. The expressions of this language will depend on the ontology on which those expressions are defined.

> Given an ontology language L, an entity language Q_L is a function from $o \subseteq L$ which defines the matchable entities of an ontology o.

The set of relations which hold between entities in an alignment, Θ includes common relations such as equivalence ($=$). It can also include relations from the ontology language itself. For example in OWL, there are `owl:equivalentClass`, `owl:disjointWith`, or `rdfs:subClassOf` relations. These correspond to set-theoretic relations: *equivalence* ($=$); *disjointness* ($\perp$); and *more general* ($\sqsupseteq$).

3.3.4.3 Confidence Structure

The relations in an alignment can be of any type and are not restricted to those relations in the ontology language. These could therefore be fuzzy relations or similarity measures. The relationship between two entities can be assigned a degree of *confidence* which is a measure of trust that the correspondence holds.

> A confidence structure is an ordered set of degrees $\langle \Xi, \leq \rangle$ for which there exists its greatest element $\top$ and its smallest element $\perp$.

The higher the degree with regards to $\leq$, the more likely the relation holds. The most widely used structure uses the real number unit interval [0 1]. Other possible structures are fuzzy degrees, probabilities or other lattices [GATTM05].

3.3.4.4 Correspondence

With the definitions given above, we can define the correspondences which the matching algorithms must find.

> Given two ontologies o and o' with associate entity languages Q_L and $Q_{L'}$, a set of alignment relations Θ and a confidence structure over Ξ, a correspondence is a 5-uple:
>
> $$\langle id, e, e', r, n \rangle$$
>
> such that:
> id is an unique identifier of the given correspondence;
> e $\in Q_L(o)$ and $e' \in Q'_{L'}(o')$;
> r $\in \Theta$;
> n $\in \Xi$.
>
> The correspondence $\langle id, e, e', r, n \rangle$ asserts the relation r holds between the ontology entities e and e' with confidence n.

3.3.4.5 Alignment

An alignment is defined as a set of correspondences.

> Given two ontologies o and o', an alignment is made up of a set of correspondences between the pairs of entities belonging to $Q_L(o)$ and $Q_{L'}(o')$ respectively.

Multiplicity must also be considered. These are alignments with entities involved in more than one correspondence. They are denoted by the use of * (zero-or-more), or + (more-than-zero) in their cardinalities. When the only considered relation is equality and confidence measures are not taken into account, we can make the following observations:

Total Alignment Total alignment occurs when all entities of one ontology must be successfully mapped to the other. This property is useful when it is necessary to thoroughly transcribe knowledge from one ontology to another. In this case, no entity can remain untranslated.

Injective Alignment Injective alignment ensures that entities distinct in one ontology remain distinct in the other ontology, i.e., all entities of the other ontology are part of at most one correspondence. This characteristic of an alignment is useful when the reversibility of an alignment is important.

In cases when correspondence relations are not equivalence, injectivity does not guarantee reversibility of the alignment used as a transformation.

3.3.5 Matching Techniques

There are many basic techniques for matching ontologies. These basic techniques are not normally used individually, but are combined into larger Matching Systems. These Matching Systems are described in detail in section 3.3.6. In this section, we give an overview of matching techniques based on those given in [ES07].

3.3.5.1 Name-based techniques

This matching method compares the name, label, or comments associated with an entity in order to find those which are similar. There are two methods for comparing terms which consider either the character strings only, or which use some linguistic knowledge to interpret those strings. These two methods are described below:

String-based methods These methods take into account the structure of the string as a sequence of letters. For example, these methods would find similarity between *Car* and *MotorCar*, but not *Car* and *Vehicle*. Cohen et al. [CRF03] compares various string-matching methods which view strings as either: an exact sequence of characters, an erroneous sequence of characters, a set of characters and a set of words. Euzenat and Shvaiko [ES07] distinguishes between:

1. Normalisation techniques which are used for reducing strings to be compared to a common format.
2. Substring or sub-sequence techniques that base similarity on the common letters between strings.
3. Edit distances that further evaluate how one string can be an erroneous version of another. A popular example of this technique is the Levenshtein Distance [Lev65]. This calculates the minimum number of insertions, deletions, and substitutions of characters required to transform one string into the other.
4. Statistical measures that establish the importance of a word in a string by weighting the relation between two strings.
5. Path comparison. This technique compares not only the labels of entities but the sequence of labels of objects to which those bearing the label are related [VE97].

Software packages exist for computing string distances such as the Alignment API [Euz04] and SimPack [Sim08].

Language-based methods These methods extract meaningful terms from text using NLP (Natural Language Processing) techniques. Therefore we can gain an insight into the similarity of ontology entities by comparing these terms and their relations. We classify these into two type of technique:

1. Intrinsic Methods. Linguistic normalisation converts each term into a standardised linguistic form which can be recognised. Maynard and Ananiadou [MA01] describe three types of term variation: morphological (variation in the word based on form and function but coming from the same root); syntactic (the grammatical structure of a term varies); and semantic (variation on a term usually as a hypernym or hyponym). Linguistic software chains such as [Bri92] can obtain normal forms of string denoting terms.
2. Extrinsic Methods. These methods use external resources in order to find similarities between terms. These external resources include:

   Lexicons This resource consists of a set of words together with a definition for each word (also called a Dictionary).

   Multi-lingual Lexicon These are dictionaries which have a list of terms together with the same term listed in a different language. Obviously, these resources can be very useful when dealing with ontologies in multiple languages.

   Thesauri A thesaurus is a lexicon which has relational information, including: hypernym, hyponum, synonym, and antonym. An example of a resource which provides these features is WordNet [Mil95].

   Terminologies These domain specific resources contain a thesaurus for terms and often contain phrases rather than single words.

A further comparison of linguistic methods is presented in [BH06].

3.3.5.2 Structure-based techniques

This matching method compares the structure of entities instead of or in addition to their names or identifiers. There are two variations on this technique: *internal structure comparison* which compares name, annotation and properties of an ontology; and *relational structure comparison* which compares an entity to the other entities to which it is related. These two methods are described below:

Internal Structure Comparison These are sometimes referred to as Constraint-based approaches [RB01]. Criteria used for this technique include: the set of each entity's properties; the range of each entity's properties; their cardinality or multiplicity; and the transitivity or symmetry of their properties. Significant work on these techniques includes: Datatype Comparison [VE97]; Domain comparison [Val99]; and Multiplicity and Property comparison [LCM$^+$02].

Relational Structure Comparison An ontology can be considered as a graph whose edges are labelled by relation names. The graph homomorphism problem [GJ79] provides insight into finding correspondences between elements - the maximum common directed subgraph. Typical techniques in this area include: Taxonomic structure [VE97]; the Leacock-Chodorow [LMC98] method which considers the shortest similarity path; and Wu-Palmer [WP94] which takes into account the distance of entities from the root hierarchy. Matching ontologies using these techniques is very powerful because it takes into account the relations between all entities. However, as with all ontology matching techniques, it gains most power when used not alone but in combination with other techniques.

3.3.5.3 Extensional techniques

When matching ontologies, if individual instances are available, this allows a wealth of information on which to base matching decisions. For example, if two classes have the same set of instances, there is a high probability that these two classes can be mapped to each other. We can also infer some useful information when instances do not match. Some attributes should always be the same, for example title of a book or its ISBN number. So if these fields differ, they almost certainly do not represent the same book. There are three categories of extensional methods:

Common extension comparison If two classes share instances we can test their intersection and gain a judgement as to their similarity. Larson et al. [LNE89] and Sheth et al. [SLCN88] formalised this. Problems occur when faults arise - a wrong conclusion on domain relationships may be drawn with small data errors. Using Hamming distance considers the size of the symmetric difference normalised by the size of the union of instances. It therefore tolerates some individuals being misclassified. An additional technique is to compute the concept lattice (based on Formal Concept Analysis [GW99]). There is a duality between a set of instances and their properties such that the more properties which are constrained, the fewer instances which satisfy those constraints. Therefore, a set of instances with properties can be organised into a lattice of concepts covering those instances. From this, a set of correspondences can be extracted.

Instance identification techniques If there is no common set of instances, the next approach is to try to identify which instance from the first set corresponds to which instance in the second set. A natural approach to doing this is to identify key fields in the data schema. When key fields are not available or they are different, we can use instance data to compare property values. One such technique is object identification [LSPR93].

Disjoint extension comparison There are some occasions when it is not possible to infer a dataset common to both ontologies. In this case, it is more appropriate to use approximate techniques. There are two main approaches for doing this:

1. Statistical. We can calculate statistics about property values in the instance data. This includes maximum, minimum, mean, variance, existence of null values, existence of decimals, scale, precision, grouping and number of segments. We can therefore characterise the domain of class properties from the data. These measures, if we are dealing with a statistically representative sample, should be the same or similar for two equivalent classes in different ontologies. Alternatively, data patterns and distribution can be analysed using neural networks. This is the process suggested by [LC94] which results in better fault tolerance.

2. Matching-based comparison. The previously mentioned distance comparison methods calculate their value based on the distance between one pair of members of the sets. However, the average linkage is the value function of the distance between all possible comparisons. Valtchev and Euzenat [Val99] consider that the elements to be compared should be those which correspond to each other. This approach means that we must already have an alignment (mapping) available which we wish to compute.

Extensional information is very useful when performing ontology matching. The information is independent from the conceptual part of the ontology. When a set of instances is available which is characterised in both ontologies, these techniques can be very useful. However, there are times when instance data is not available (e.g., for reasons of confidentiality). In this case, other techniques in this section must be considered.

3.3.5.4 Semantic-based techniques

Ontology matching is essentially an inductive technique. However, deductive techniques can be applied to the problem if a preprocessing phase is used to provide *anchors* (i.e., entities which are declared to be equivalent).

Anchoring Techniques The main method of anchoring is through the use of external ontologies. Common understanding is required between two ontologies before matching can take place. This can be provided via formal intermediate ontologies which define common context or background knowledge [GSY06]. A background ontology which covers the domain of interest comprehensibly helps in disambiguation of terms. Examples of common upper-level ontologies include: Cyc [LG90]; SUMO (Suggested Upper Merged Ontology) [NWN$^+$01]; and DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [GGMO03] [Gan04]. There is a two stepped approach suggested by [AKtKvH06]. First, the ontologies to be matched, o' and o'' must be matched to the background ontology o. Secondly, relations are derived by using the correspondences discovered in the anchoring step. A reasoning service can be used since concepts of the ontologies o' and o'' become a part of the background ontology o via anchors. Combining the anchor relations with the relations between the concepts from the reference ontology allows the derivation of relations between the concepts of o' and o''. Other techniques involve using as many context ontologies as possible (typically provided by the semantic web) and finding the most relevant ones for the problem in hand [SdM06].

Deductive Techniques The first possible technique is to use propositional satisfiability (SAT) techniques. These steps are described in [GS03], [BS03], [GSY04], [Shv04] and are as follows:

1. Build a theory or domain knowledge (axioms). The theory is constructed using matchers (e.g., those based on WordNet), or using external ontologies.
2. Build a matching formula.
3. Check for the validity of the formula.

The second possible Deductive technique is to use Description Logics. In this method, relations can be expressed with respect to subsumption. If two ontologies are first merged, and then each pair of concepts and roles is tested for subsumption, terms with the same interpretation can be matched [BSZS06]. Further techniques which use description logics for ontology matching in the scenario of spatio-temporal databases includes [PS00] and [SVC$^+$05].

3.3.6 Matching Systems

In the previous section we discussed the basic techniques for ontology matching. These techniques are not particularly useful when used in isolation. However, when they are carefully combined into a working matching system, good results can be achieved. In this section we survey matching systems. Where possible we state which basic techniques the systems employ to deliver their matching strategy. The matching systems are categorised into schema-level and instance-level systems. As ontology matching systems are often developments on previous work, we have chosen to describe only the latest and most significant systems. We state where systems are based on earlier work.

3.3.6.1 Schema-based systems

Schema-based systems rely mainly on schema-level input information during the process of ontology matching.

CtxMatch CtxMatch [BMSZ03] [BSZ03] is a sequential system which takes a semantic matching approach (see section 3.3.5.4). It computes logical relations such as equivalence and subsumption between concepts and properties. It uses WordNet to find class matches at the element level.

CtxMatch2 CtxMatch2 [BSZS06] adds additional features to CtxMatch in that it handles properties and description logic reasoners such as FaCT [Hor98] and Pellet [SPG$^+$07].

S-Match S-Match [GSY04] again takes a semantic matching approach and was initially based on CtxMatch. It further evolved by including extensions for element and structure-level matchers, alignment explanations and iterative semantic matching. However, S-Match is limited to tree structures which are encoded in XML and does not consider properties. The libraries in S-Match contain a number of basic element-level matchers from string-based, sense-based and gloss-based matchers. Structure-level matching includes SAT solvers and ad-hoc reasoning methods [GYG05].

HCONE HCONE [KKV06] takes a user-centred approach to ontology matching and provides three operating modes: fully automated; semi-automated and user-based. In the user-based mode, users must provide feedback on all the alignments (calculated using WordNet). In semi-automatic mode, the user is only requested to intervene in a heuristically-calculated limited number of important cases. In this way, HCONE makes important strides in balancing the advantages and disadvantages of manual and automatic mapping.

Moa Moa [KJH$^+$05] is an ontology merging and alignment tool which takes ontologies specified using OWL-DL as its inputs. It is able to compute equivalence and subsumption relations between entities. Moa makes extensive use of WordNet. However, matching itself is based on rules applied to all pairs of entities in the two input ontologies.

ASCO ASCO [BDKG04] takes ontologies in RDF Schema as its inputs. There are three sequential phases:

1. Linguistic matching normalises terms and expressions (see section 3.3.5.1) using techniques such as Jaro-Winkler and Levenshtein distance
2. Structure matching uses the results of the linguistic matching phase to compute similarities between classes and relations.
3. Aggregation of Linguistic and Structural similarity using a weighted sum allows matching candidates which are above a certain threshold score to be selected for the resulting alignment.

ASCO2 ASCO2 [BDK05] has the same feature set as ASCO, but it uses OWL ontologies as its inputs.

BayesOWL BayesOWL [PDYP05] models uncertainty using a probabilistic framework. The approach has three steps:

1. Two Bayesian networks are created from the two input ontologies.

2. A search engine is queried to learn joint probabilities and create candidate matches between the two Bayesian networks.

3. The final alignment is produced by performing Bayesian inferencing.

OMEN The Ontology Mapping ENhancer (OMEN) [MNJ05] is also based on a Bayesian network and produces a semi-automated system. An initial probability distribution between the two input ontologies can be derived from element level linguistic matchers. OMEN then provides a structure-level matching algorithm to derive new mappings or discard existing mappings which it deems false. The approach has four steps:

1. A Bayesian network is created where nodes represent mappings between pairs of classes or properties and edges represent the influences between the nodes of the network (in this way OMEN differs from BayesOWL).

2. A set of meta-rules are used to generate a conditional probability table.

3. Inferences are made to generate *a posteriori* probabilities for each node.

4. The *a posteriori* probabilities which are higher than a certain threshold are selected for the final alignment.

3.3.6.2 Instance-based systems

Instance-based systems rely mainly on the data held in the ontologies during the process of ontology matching.

LSD The LSD (Learning Source Descriptions) system [DDH01] handles tree-structures held in XML schemas (each node is an XML tag name). LSD provides a semi-automatic mapping system where elements of the source schema are aligned to a global schema in data integration. These manually-created mappings between the mediated schema and a selection of source schemas are intended to allow learning which promotes the automatic mapping of subsequent schemas.

GLUE GLUE is the successor of LSD and follows a multi-strategy learning approach using several basic matchers. It calculates the joint distributions of the classes rather than using a particular definition of similarity. The system has three steps:

1. It uses a content learner and a name learner to learn the joint probability distributions of classes of two taxonomies.

2. A similarity matrix between terms of the two taxonomies is created using a user-supplied function.

3. Domain-dependent heuristics are used to filter the best matches from the similarity matrix.

SBI&NB SBI (Similarity-Based Integration) [ITH03] [IHT04] automatically matches classifications using statistical techniques. A naive Bayes classifier was added to produce SBI&NB. Correspondences between two classes of two classifications is done by statistically comparing the membership of documents to these classes. Structural information is taken into account by the naive Bayes classifier, enabling hierarchical classification of documents.

Kang and Naughton The Kang and Naughton technique [KN03] uses a structural instance-based approach to discover correspondences among attributes with opaque column names. Opaque column names are those which have very limited meaning such as "A" or "B" rather than "Name" or "Address". There are two phases to the process:

1. Weighted dependency graphs are constructed from the two input table instances. These are based on mutual information and entropy between the data. Mutual information is computed over all pairs of attributes in a table. A weight on an edge represents mutual information between two adjacent attributes and weight on a node stands for entropy of the attribute.
2. In the second phase, a graph matching algorithm is used to discover matching node pairs. Euclidean distance and other metrics are used to assess the quality of matching.

sPLMap sPLMap (Probabilistic Logic-based Mapping) [NS05] [NS06] supports uncertain schema mappings via both logic and probability theory. There are three main steps in the system:

1. Probability distributions (known as interpretations) are created from a judgement on the quality of all possible correspondences.
2. Basic matchers (see section 3.3.5) are used to measure the plausibility of each correspondence. Linear and logistic functions are then used to aggregate the matcher results.
3. The Bayes theorem is used to create correspondence weights from the computed probabilities.

3.4 Summary

In summarising this chapter, there are six main areas to consider:

- Ontologies
- Ontology Mapping
- Ontology Mapping Algorithms
- The Semantic Web
- Ontologies of Visualization
- Ontology Success Stories
- The Feasibility of Using Ontologies and Ontology Mapping for Information Visualization

3.4.1 Ontologies

Ontology Creation Ontologies are human-designed, yet machine-readable. Even though they are human-designed, they are difficult to understand in isolation from an application unless the person is an ontology modeller. Often ontologies are designed in isolation of an application which means they are good as a shared vocabulary and conceptualisation. However, the process of ontology creation is often more useful than the resulting ontology.

Rules Ontologies consists of hard (or crisp) rules. These are usually specified using Description Languages such as OWL Lite or OWL DL.

Reasoning Reasoning upon ontologies is done via the Description Logic on a reasoner such as Racer Pro. This form of reasoning only gives one result because the rules specified are prescriptive. Note: a class or an instance might be (re-)classified in many ways. If every single re-classification is considered as one result, then many classifications can be produced. However, if re-classification is considered to follow a specific algorithm that evaluates all re-classifications, then this is considered as the final result. Other forms of reasoning such as those provided by MYCIN [BS84] or certainty factors [MM94] [Cha08] give multiple possible answers.

Fuzzy Ontologies A degree of flexibility is provided by fuzzy ontologies. However, they are not easy for humans to understand or alter. In fact, their main purpose is to augment traditional ontologies rather than a replacement for the absolute relations between concepts.

3.4.2 Ontology Mapping

Singularity As with the inferencing on ontologies, ontology mapping is an exercise which only gives one result. This singularity means that there is no scope for comparing the results of different mapping possibilities.

Applications The main application of ontology mapping is for data integration. This is a schema mapping exercise which is typically a one-time exercise in order to gain the best translation of source data to target data.

Exploiting Cognitive Ability The result of an ontology mapping process is a single mapping presented either textually, or using simple graphics. Because of this user interaction approach, it does not exploit aspects of the human brain which could be used to pick out the best options. For example, if the result of the ontology mapping process was a display of the data (or a sub-set of it) which had been translated, then there would be the potential for a human to assess the validity of the mapping, making better use of their cognitive ability.

Application Limitation Ontology Mapping is about mapping between two different schemas which ultimately represent the same thing (data stores). However, there is potential for ontology mapping to be used to translate between ontologies which represent different types of entity (for example a data source and another form of expression such as visual, audial or textual displays).

3.4.3 Ontology Mapping Algorithms

Techniques Ontology Mapping algorithms have been developed which use a variety of different aspects of ontologies to create worthy mappings. These include schema based and instance based techniques. However, schema-based matching systems have been developed further with more examples than instance-based matching systems.

Subject Domain Most ontology matching systems are focussed and have been tested on specific subject domains (cars, books etc.). Additionally, the ontology systems deal with typically only one ontology type (OWL, DTD etc.). There are few general purpose ontology matching systems.

Tree and Graph structures Despite ontologies being inherently graph structures, most systems handle only tree structures.

Multiplicity The main correspondence type between entities is one-to-one alignments. More complicated alignments such as one-to-many and many-to-many are not handled.

Complexity Due to the inherently complex nature of ontologies, ontology mapping algorithms have also become complex. This complexity leads to a system which is difficult for humans to understand or intervene in. In any system which uses ontologies, the ontology mapping systems must use a wide range of basic techniques and combine these in order to achieve effective results. There is no one agreed "best of breed" system for this purpose. As such, the whole process of ontology mapping is a complicated one.

3.4.4 The Semantic Web

Expectations The Semantic Web has largely failed to live up to its expectations. Unfortunately, the mainstream world wide web is largely devoid of explicit semantics. Instead, its focus still remains on one of presentation to humans. Therefore, without a so called "killer application" the take-up of semantically rich data on the web is likely to be a slow one, or indeed one which is never realised.

The Pragmatic Semantic Web There have been some reasonably successful attempts at "boot-strapping" the semantic web. Most notably using tools developed by the SIMILE group. While these tools have proved very useful and interesting in computer science's academic circles, mainstream adoption has yet to emerge. An interesting scenario is the use of Exhibit client-side databases on webpages. These can be semantically tagged and useful visual displays (e.g., Time Line) can be created. However, there is no automatic mapping between the data on the Exhibit and the Time Line. Instead, the mapping must be done manually and linked programmatically.

3.4.5 Ontologies of Visualization

All encompassing There has been a large amount of work to produce a general-purpose ontology of visualization which encompasses: task and use; representation; process; and data.

Focus This task has been performed by a committee and has resulted in some interesting results, including a shared terminology and consensus. However, the resulting ontology has no application aside from integrating the visualization community.

Inferencing As well as providing a shared terminology and consensus, the aim of the existing efforts to create an ontology of visualization has been to allow inferencing also.

3.4.6 Ontology Success Stories

The prognosis given in this chapter related to ontologies has been relatively negative (when compared for example with that of visualization given in chapter 2). However, the discipline is relatively young and its potential has yet to be fully realised. Despite this, there have been some important successes. These are detailed below:

Protege ontology editor The Protege tool [NSD$^+$01] has over 100,000 registered users and is approaching version 4 (see section 3.2.5). It has gained a large number of users both within the academic ontology community and within specific research domains, particular bioinformatics. The tool outputs to a number of common ontology formats which allows interoperation between other systems. However, Protege is often used as a stand-alone tool which allows users to define ontologies purely as an exercise to define and structure terminology. This is particularly important in a field such as bioinformatics which is developing so fast with geographically distributed research groups.

Bioontology Portal The National Center for Biomedical Ontologys BioPortal [NCB02] is a Web-based application for accessing and sharing biomedical ontologies. It has over 130 core ontologies which detail over 700,000 concepts. The system is used by a consortium of biologists, clinicians, informaticians, and ontologists who develop methods allowing scientists to create, disseminate, and manage biomedical information and knowledge in machine-processable form. In this respect, their goal of allowing ontologies to be semantically interoperable and useful for improving biomedical science and clinical care has been achieved.

Both of the above success stories are using ontologies in a very specific context with good tool support and clear objectives. We hope to replicate this success by applying similar goals to the use of ontologies with visualization.

3.4.7 The Feasibility of Using Ontologies and Ontology Mapping for Information Visualization

There has been a great deal of work produced by the academic research community in the fields of ontologies, ontology mapping and the semantic web. Some of this research has made its way out of computer science research into more practical areas. Examples include SemanticWorks [Alt08b] and Protege [NSD+01] which has found popularity in the biomedical research community. However, the use of ontologies as a core technology in large-scale systems has been limited. This is partly due to the complexity of the technology [msK06] but also because of the relative immaturity of the standards [Car07].

Additionally, there seems to be an "all or nothing" approach when using ontologies and related technologies. Rather than being used as an assistive technology to support processes, ontologies typically aim to be the core technology in a system. Obviously, the risk involved in building a full system using new and unproven technology is too great for many software developers. As a result, the technologies are often (unfortunately) shunned in favour of proven but more primitive technologies. In the next chapter, we examine and evaluate the use of ontologies in selective areas of the information visualization process.

Chapter 4

Concepts of Visualization As Mapping

Contents

4.1 Introduction

In chapter 1, we presented the motivation for *general purpose information visualization for non-experts*. In chapters 2 and 3, we surveyed the areas of Information Visualization and Ontologies / Ontology Mapping. Based on the conclusions of chapter 2 and 3, it is clear that there is a need for an *automatic, knowledge-driven visualization pipeline.*

In this chapter we lay the conceptual foundations for chapters 5 and 6 by discussing the following:

Reasoning Model For automatic visualization, there are different methods for deciding the best mappings between source data entities and target visual artefacts. We describe three such methods

Capturing Semantics If we are to provide a knowledge-based approach to the visualization pipeline, we need to define a model for capturing semantics. We describe such a model.

Data Structure When dealing with the mappings between the data source and visualization target, we can consider different data structures. We describe and critique two such approaches.

So far, there are no systems in existence which can address the motivations given in chapter 1. The closest available solution is the (semi-)automatic visualization pipeline provided by Tableau [MHS07]. We define this approach as semi-automatic since the user is still required to use and understand the user-interface of the Tableau tool in order to iterate on the visualizations which have been automatically produced. We illustrate the three levels of user-interaction automaticity in figure 4.1. The goal of this work is to produce a fully automatic visualization pipeline which requires no user-interaction. This is necessary in order to address the motivational goals of *general purpose information visualization for non-experts*.

In this chapter, we describe how mapping principles can be applied to the area of information visualization to form the concept of *Visualization As Mapping*. The general pipeline for this is shown in figure 4.2. Source Data Entities are the data fields in the source input. Target Visual Representation Artefacts are the visual artefacts in the target visualization technique which can be used to represent the source data entities. Both the Source Data Entities and the Target Representation Artefacts have semantics associated with them. The Automatic Mapping process takes the Source Data Entities together with their semantics and tries to generate a high-quality, cognitively useful visualization.

The benefits of building an *automatic, knowledge-driven visualization pipeline* are that we can open the field of information visualization to a wider, non-expert audience. We can also attempt to capitalise on the expert knowledge which is present in the visualization community in order to benefit those who are outside it.

In order to build an *automatic, knowledge-driven visualization pipeline*, we need to address three aspects:

Automatic Mapping Method How do we map the source data entities to the target representation artefacts?

Capturing Semantics How do we effectively capture the semantics of both the source data entities and target representation artefacts in a way that drives the automatic mapping?

Mapping Data Structure When mapping between the source data entities and target representation artefacts, what core data structure should we use?

The possible choices for the three aspects listed above are:

Automatic Mapping Method: ***Type-constrained Inferencing, Hybrid, and Case-based Reasoning.*** We describe three methods by which a system reasons about how best to map between two schemas (whether the schemas are tree-based or graph-based). We discuss the advantages and disadvantages of the approaches and how the work in this thesis relates to each approach.

Capturing Semantics: ***The Information Realisation Model.*** We propose a general mapping model for creating textual, graphical and audial representations of semantically-rich information (which we call Information Realisation). The model is not restricted to Visualization, but can also output to Textual and Audial formats too. Card et al. [CMS99] use the term *Information Perceptualisation* to define Information Realisation. We show how ontology descriptors can be used to capture the expressive

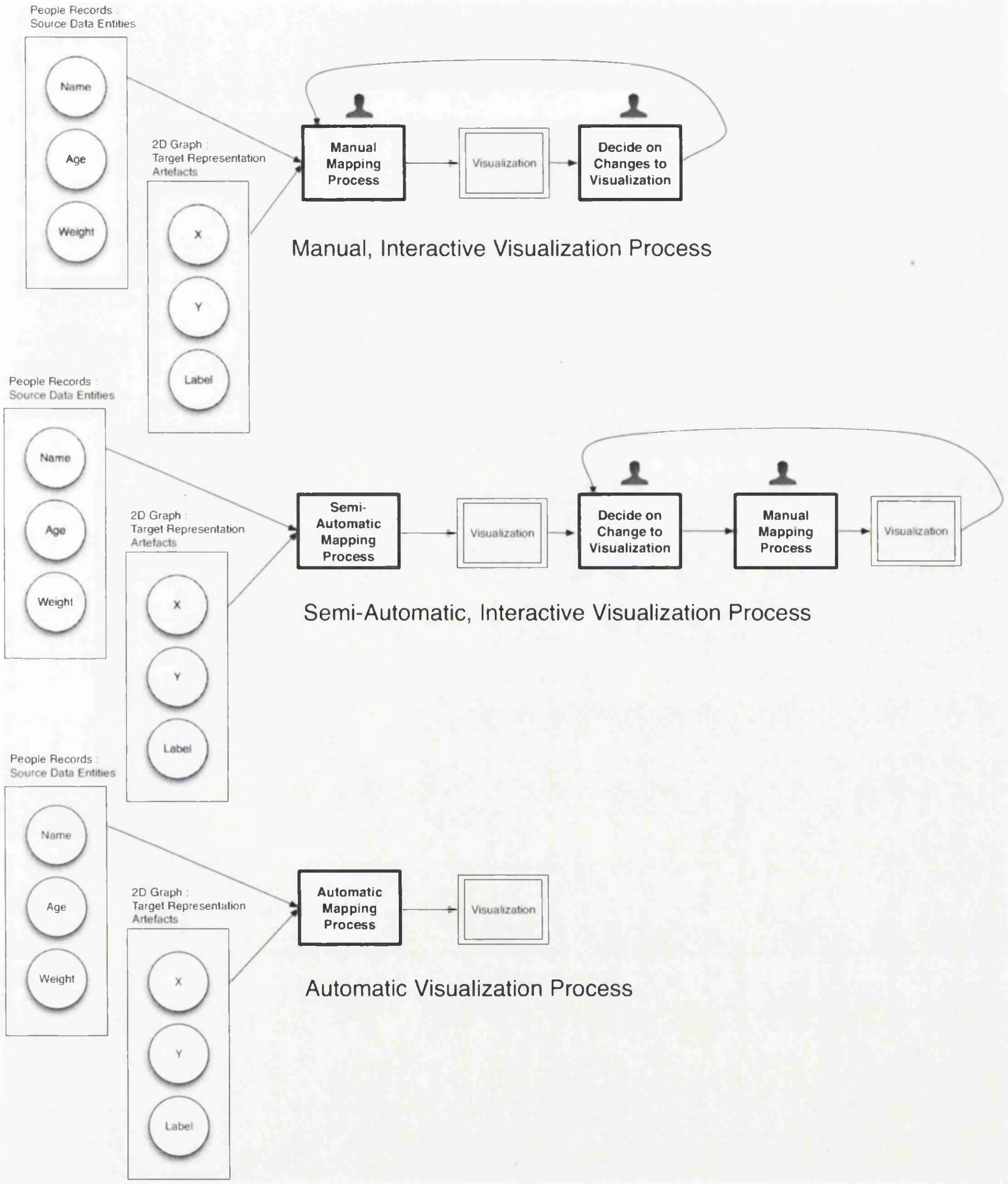

People Records : Source Data Entities
Name
Age
Weight
2D Graph : Target Representation Artefacts
X
Y
Label
Manual Mapping Process
Visualization
Decide on Changes to Visualization
Manual, Interactive Visualization Process
People Records : Source Data Entities
Name
Age
Weight
2D Graph : Target Representation Artefacts
X
Y
Label
Semi-Automatic Mapping Process
Visualization
Decide on Change to Visualization
Manual Mapping Process
Visualization
Semi-Automatic, Interactive Visualization Process
People Records : Source Data Entities
Name
Age
Weight
2D Graph : Target Representation Artefacts
X
Y
Label
Automatic Mapping Process
Visualization
Automatic Visualization Process

Figure 4.1: The three levels of automaticity for information visualization user-interaction.

characteristics of the source format, target format and the perceptualisation environment. Part of this work has been presented in the paper, "Information Realisation: Textual, Graphical and Audial Representations of the Semantic Web" [GSG+06].

Mapping Data Structure: ***Tree-centric versus Graph-centric Mapping.*** We consider two approaches to the information mapping problem: using a Tree-centric approach; and using an Graph-centric approach. In this section, we demonstrate and contrast both approaches using two different mapping toolkits: the commercial product Altova MapForce [Alt08a] for a tree-centric approach; and the public domain research tool

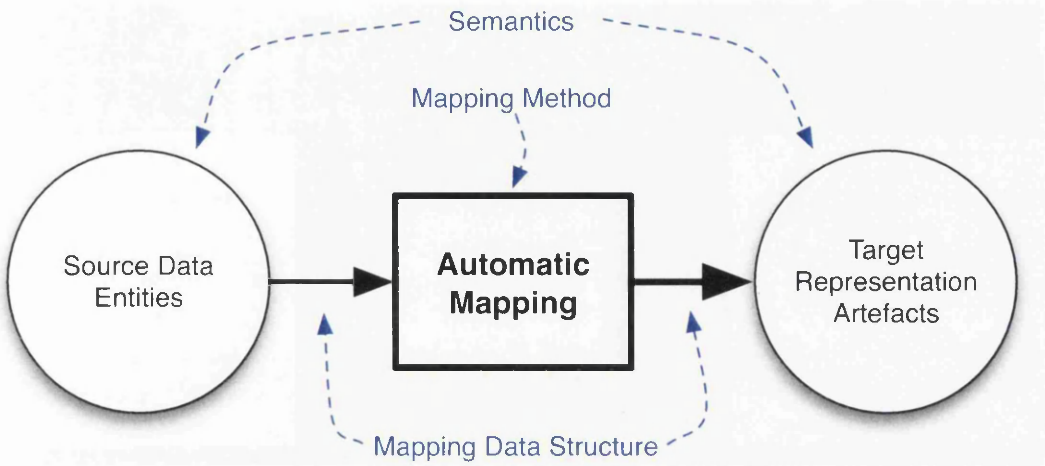


Figure 4.2: The general concepts of Visualization As Mapping

MAFRA (MApping FRAmework) toolkit [MMSV02] for a graph-centric approach.

We summarise the chapter by describing how we will produce two toolkits (VizThis and SemViz) which demonstrate the various aspects of *Visualization As Mapping*.

4.2 Type-constrained Inferencing, Hybrid, and Case-based Reasoning

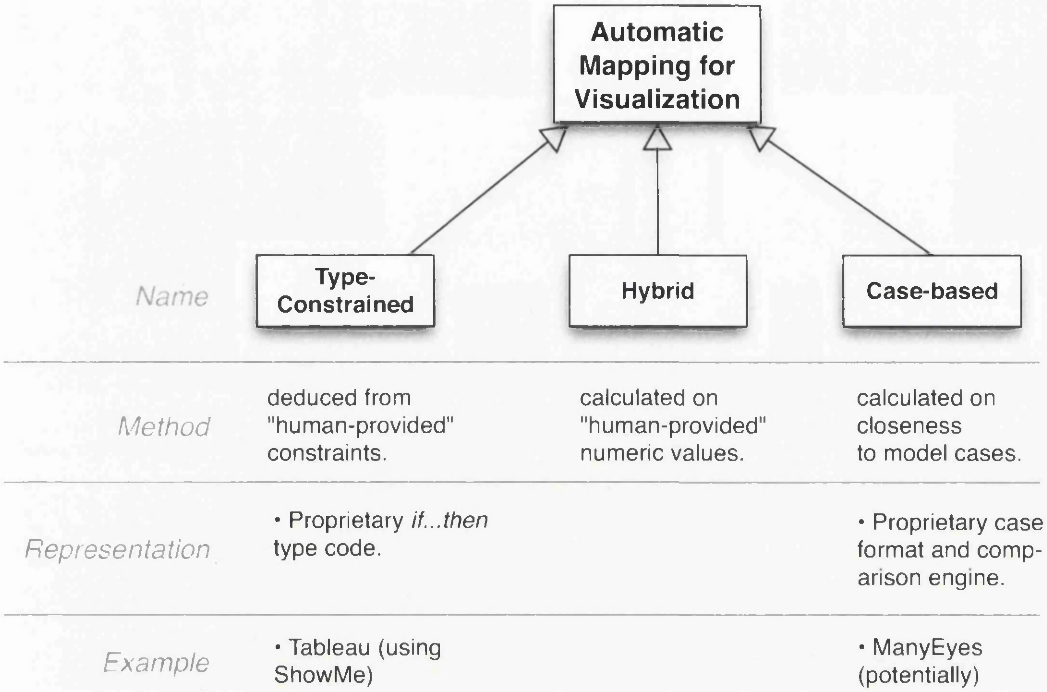
When considering the Visualization As Mapping concept, we have a data source with a defined schema and a target visualization format with a defined schema. It does not matter whether these schemas are tree-based or graph-based. The logic which drives the mapping between these schemas can be categorised into three types. These types are summarised in figure 4.3 and are described below.

4.2.1 Type-constrained Inferencing

A type-constrained inferencing approach consists of a set of general *constraints* which are applied to a set of *facts* about the source and target formats. The *constraint matching algorithm* then tries to produce a mapping which is consistent with the constraints and the facts. There could be many possible mappings which match the constraints given. The system is *closely coupled* in that the system is not designed for the different modules (facts and constraints) to be developed or altered independently of each other.

This is the approach which the commercial visualization package Tableau uses in its feature, ShowMe. Only one result is produced and there is no measure of the correctness of this result, relative to other results or otherwise. The approach is shown in figure 4.4.

The disadvantages of a Type-constrained approach are:



Name	Type-Constrained	Hybrid	Case-based
Method	deduced from "human-provided" constraints.	calculated on "human-provided" numeric values.	calculated on closeness to model cases.
Representation	• Proprietary *if...then* type code.		• Proprietary case format and comparison engine.
Example	• Tableau (using ShowMe)		• ManyEyes (potentially)

Figure 4.3: Type-constrained Inferencing, Hybrid, and Case-based Reasoning

Narrowness of Semantics This technique only considers the type of the source data (i.e., whether it is qualitative or quantitative, the variance of the values etc.) There is no consideration of what the data fields actually represent.

Relative Comparison A type-constrained approach will typically only result in one solution. Even if multiple solutions are output, there is no means of comparing their relative quality.

Closely Coupled The constraints, facts, and constraint matching algorithm are closely coupled. The system does not lend itself to an open or distributed approach to these separate modules. As such it is difficult to add or alter constraints and facts.

4.2.2 Case-based Reasoning

In this approach, there is a database of past *cases* with their solution (the *case base*). The new problem is matched against the whole of the case base and a *fitness score* is calculated for each case (usually using a *similarity measure*). The higher scored cases represent the better solutions. There is usually a post-process to alter the final solution to fit the given problem to be solved. This is the approach which the commercial product, CBR Express takes.

Applying the Case-based Reasoning approach to visualization would result in a system where historically (or previously system generated) cases would reside in the case base. Given a source data input, the system would compare it to previous visualizations in the

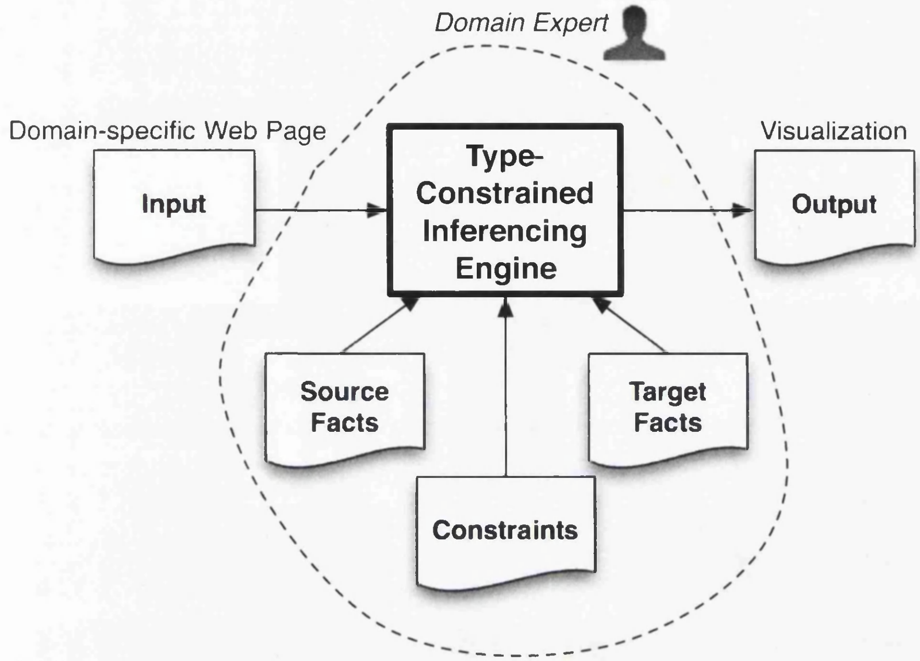


Figure 4.4: Type-constrained Inferencing

case base to find the nearest fit (relative to some measure). The best case would then be appropriately modified according to the source data to produce the final visualization. The new case could then be added to the case base for later use. The approach is shown in figure 4.5.

There are no known visualization toolkits (commercial or public domain) which take this approach. However, the web-based, collaborative visualization toolkit, ManyEyes [VWvH$^+$07] has proved very popular since its launch. As such, it has a "case base" of approximately 20,000 datasets (as of mid-2008). Therefore, this would provide an ideal database and infrastructure on which to add a Case-based Reasoning system. ManyEyes currently uses a process of manual, user-driven visualization. However, Case-based Reasoning would allow an automatic visualization approach to be investigated.

The disadvantages of Case-based Reasoning are:

Scalability The case base can quickly get very large. When the input parameter space is also very large, then the number of cases and parameters to check can quickly become a constraint on performance. However, performance can be increased by clustering cases with common features in order to decrease the size of the search space.

Case bias If the cases in the case base are from a narrowly defined area, then this bias will affect the quality of the results.

Noise Any cases in the case base which are of poor quality can result in badly score and evaluated input cases.

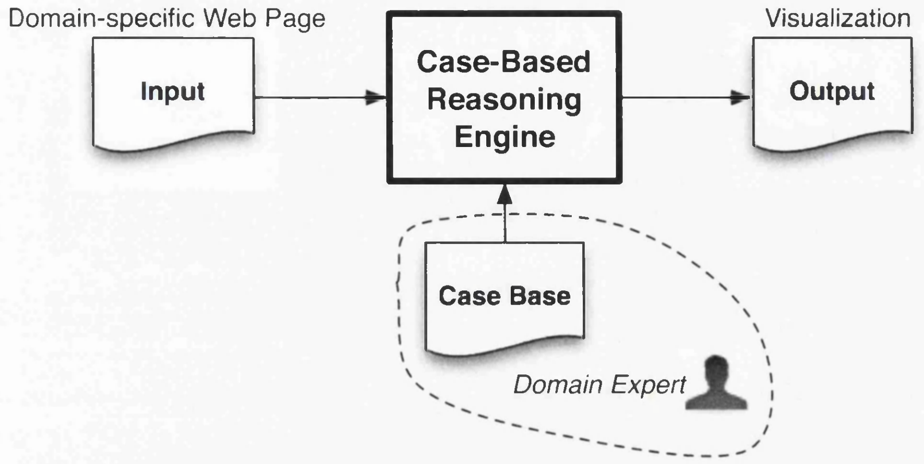


Figure 4.5: Case-based Reasoning

4.2.3 Hybrid Reasoning

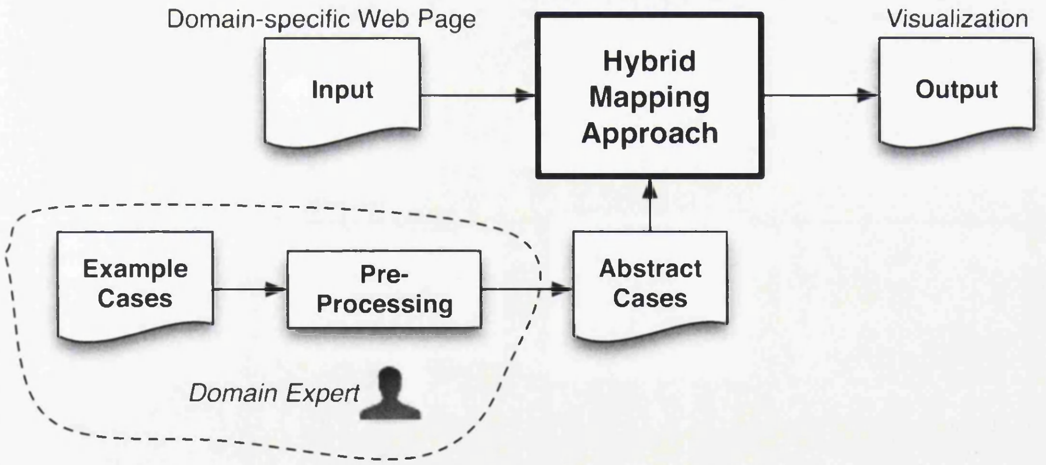


Figure 4.6: Hybrid Mapping Approach

If we take the human-provided aspect of the Constraint-based Inferencing and combine it with the Case approach of Case-based Reasoning, we gain a Hybrid approach which allows us to have Cases which are defined and editable by human. These cases are called Abstract Cases and represent extracted semantics from the Source and the Target Domains. The Abstract Cases are *pre-processed* from a corpus of Example Cases and could be a machine or human-driven process. This approach is shown in figure 4.6.

4.2.4 Comparison Summary : Type-constrained, Case-based and Hybrid Mapping

Type-constrained mapping provides an effective means of providing automatic visualization. The disadvantages associated with the consideration of only a sub-set of semantics can actually manifest themselves in a simpler system design. It is therefore worth investigating the effectiveness of this approach (see section 5).

Case-based reasoning requires a significant infrastructure investment before a judgement of the effectiveness of the automatic mapping can be gained. Together with the danger of case noise and bias, it seems that it is not worth investigating this approach further.

A hybrid approach, where semantics are extracted from a small set of high quality cases is an effective option. The approach of creating Abstract Cases allows a knowledge store of high quality semantics to be created in which to drive automatic mapping. Therefore, this option will also be considered further (see section 6).

4.3 The Information Realisation Model

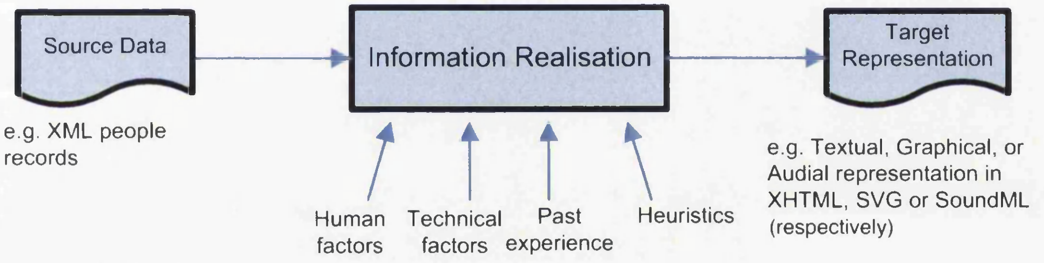
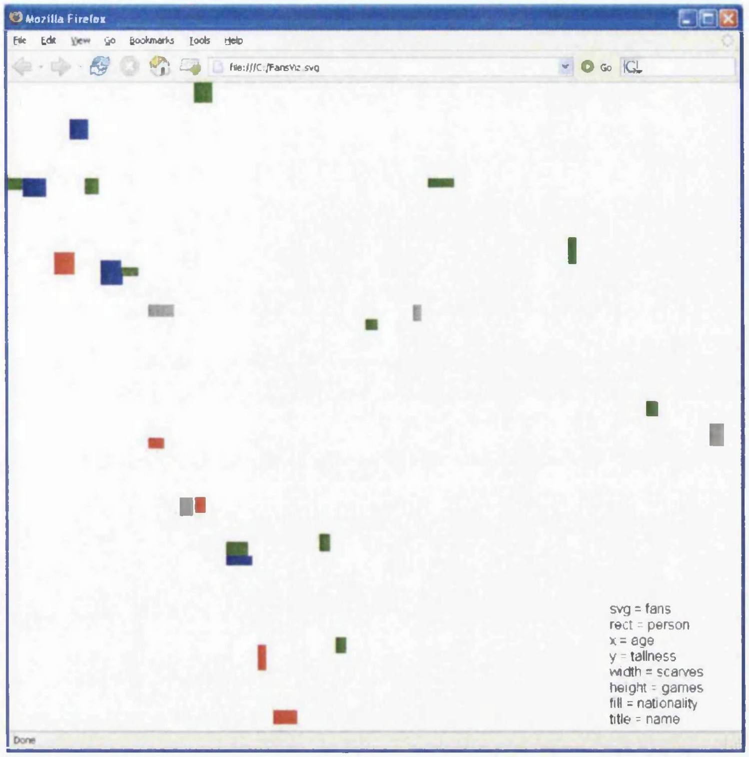


Figure 4.7: A General Model of Information Realisation

Information Realisation is the process of presenting data as Textual, Graphical or Audial information to a human user. It is not limited to these formats, as it could also model touch (haptics [OL05]) or smell (olfactics [NNHY06]) outputs. In this section, we discuss the importance of this concept with respect to the accessibility of Semantic Web data to a diverse target audience. We provide an ontological point of view, defining the expressive characteristics and application domain of representation formats, thus presenting a system which produces representations customised to the user environment and the nature of the source data. Our approach considers the semantics of the data, not just the structure, and aims to present the information in the most semantically appropriate manner for the given target environment. The model we present develops from the Haber-McNabb Dataow Reference Model [HM90] (see section 2.3.4). It also considers the concepts set out in Bertin's Component Analysis [Ber83] (see section 2.3.1). Later in this section we provide examples of a simple data set being realised as popular target representation formats: textual (XHTML, RSS - Really Simple Syndication); graphical (SVG - Scalable Vector Graphics, X3D - eXtensible 3D graphics); and audial (SoundML, VoiceXML).



svg = fans
rect = person
x = age
y = tallness
width = scarves
height = games
fill = nationality
title = name

Figure 4.8: The SVG (Scalable Vector Graphics) target representation

Information visualization is focussed on the visual (i.e., using the eyes) aspect of humans accessing information sources. We believe that a more general approach should be taken which considers other formats which target different sensory organs. For example, blind or partially sighted people should be able to "see" data as an audio representation of data. Additionally, there is the need to have data realised in different formats depending on the environment and circumstance of the user: interactive or passive use; background or foreground activity. A user who is performing a task which requires most of their attention, such as driving a car, would not benefit from a detailed, interactive, graphical visualization of a weather forecast. A passive summary which is presented audially would be more appropriate. A further example is a blind user who wishes to know statistics about a sports team's performance over a season. This could be presented as a summary audio stream. This process considers multiple output types (textual, graphical and audial) as well as multiple audience environments. The environment consists of user factors, technological factors and data factors. This approach is termed Information Realisation and it is an important part of presenting the Semantic Web. The technique is similar to the framework outlined in Jung and Sato [JS05]. We summarise Information Realisation in figure 4.7.

4.3.1 The Information Realisation Process

To illustrate the process, we consider an example concerning a set of sports fans. In this case, the data is represented as an XML file. The source data consists of 26 people (only

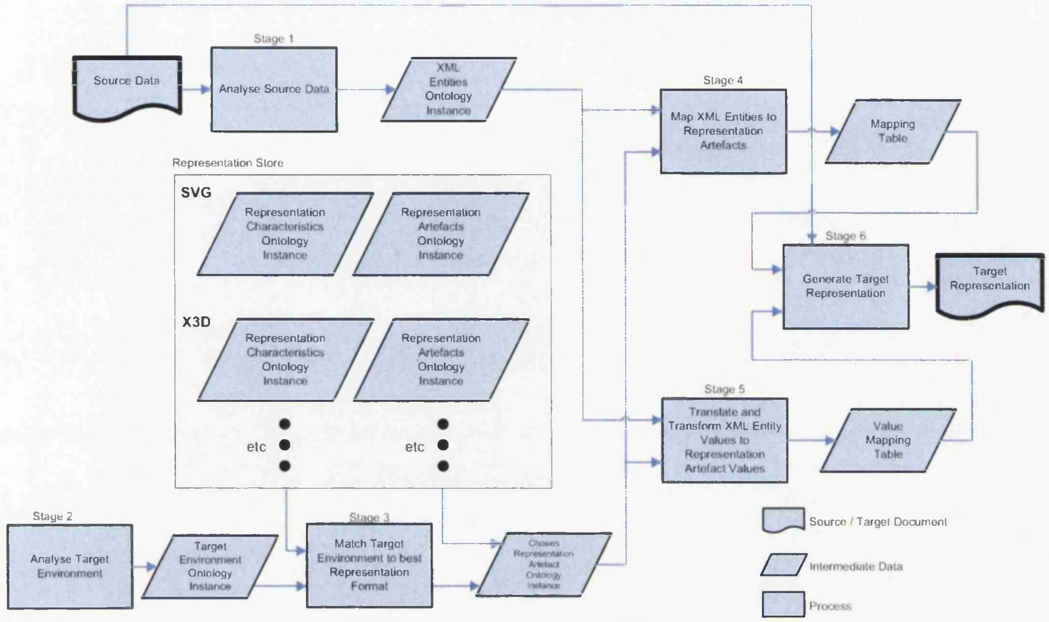

Stage 1
Source Data
Analyse Source Data
XML Entities Ontology Instance
Stage 4
Map XML Entities to Representation Artefacts
Mapping Table
Representation Store
SVG
Representation Characteristics Ontology Instance
Representation Artefacts Ontology Instance
X3D
Representation Characteristics Ontology Instance
Representation Artefacts Ontology Instance
etc
etc
Stage 6
Generate Target Representation
Target Representation
Stage 5
Translate and Transform XML Entity Values to Representation Artefact Values
Value Mapping Table
Stage 2
Analyse Target Environment
Target Environment Ontology Instance
Stage 3
Match Target Environment to best Representation Format
Chosen Representation Artefact Ontology Instance
Source / Target Document
Intermediate Data
Process

Figure 4.9: The Information Realisation Process

3 shown here) with their name, age, tallness (tallness is used rather than height to avoid confusion with the SVG attribute of the same name), nationality, scarves (number of scarves they own) and games (number of games they have been to). The source data is detailed below:

```
<fans>
    <person name="alice" age="28" tallness="1.41" nationality="welsh" scarves="2"
     games="19" />
    <person name="bob" age="37" tallness="1.02" nationality="scottish" scarves="4"
     games="33" />
    <person name="colin" age="16" tallness="1.84" nationality="irish" scarves="6"
     games="8" />
    ...
</fans>
```

The source data is to be *realised* as a visualization. In this case, using the 2D vector graphics format, SVG. The output SVG code is shown below. The SVG is shown rendered in figure 4.8.

```
<svg xmlns="www.w3.orgsvg <http://www.w3.org/2000/svg>">
  <rect title="alice" x="213" y="471" fill="red" width="12" height="18" />
  <rect title="bob" x="373" y="800" fill="blue" width="16" height="25" />
  <rect title="colin" x="0" y="109" fill="green" width="21" height="13" />
  ...
</svg>
```

The process is shown in figure 4.9, and each stage is described below:

- **Stage 1.** *Analyse Source Data* - Firstly, we analyse the source data to ascertain the nature of the data's types and its structure. The results of this analysis are stored in the

Source Entities Ontology Instance. The exact form of this ontology will depend on the source format (e.g., CSV, XML, RDF). In the sports fans example, the source data is represented using XML. Therefore we have an XML Entities Ontology Instance (see Section 4.3.2). In this example, the ontology stores the fact that *name* is a child entity of *person* which is in turn a child entity of *fans*. It also stores the types of the entities - for example that *age* is an entity whose values are numeric.

- **Stage 2.** *Analyse Target Environment* - Next, we analyse aspects of the target environment. The choice of representation format can depend on many factors, including: the User's Abilities (sight, hearing, cognition); the User's Situation (cognitive and physical engagement level); and the Technology Characteristics (interactive, visual and audial capabilities) of the output medium. The information about the nature of the target environment is output in a Target Environment Ontology Instance (see Section 4.3.4).

- **Stage 3.** *Match Target Environment to Target Representation Format* - The system compares the Target Environment Ontology Instance (from stage 2) with all Target Representation Format Ontology Instances in order to find the best match. The purpose of this is to find the best Representation Format for the user's environment. The output of this stage is the Representation Artefacts Ontology Instance (see Section 4.3.3) of the representation format which best meets the requirements of the target environment.

- **Stage 4.** *Map Source Entities to Representation Artefacts* - The inputs are: the Representation Artefacts Ontology Instance from the previous stage and the Source Entities Ontology Instance. We use target environment factors, heuristics and past experience to create mappings between Source Entities and Target Representation Artefacts. For example, we may decide that the source data entity, *age* is best mapped to the target representation artefact, *x*. The output is a mapping table between Source Entities and Representation Artefacts.

- **Stage 5.** *Translate and Transform Source Data Entity Values to Representation Artefact Values* - In many cases the values of the source data cannot be used directly in the Target Representation. Instead, a value mapping or translation process must occur. For example, the *age* source data entity values cannot be used directly for the target representation *x* value. The reason for this is that the x value may typically have a range of 0 to 800, whereas the age value has a range (in this source data) from 16 to 75. Therefore the values must be scaled accordingly. A similar situation occurs when dealing with mapping the *nationality* source data entity to the *color* target representation artefact - the values of *nationality* cannot be used directly, but must be mapped to *color* values first. This is described in more detail in the worked example (Section 4.3.6).

- **Stage 6.** *Generate Target Representation* - We generate the Target Representation by creating Representation Artefacts (in the target representation format) with the values supplied by the Value Mapping Table. The output of this stage is the final Target Representation. There are many considerations in this stage, including: maintaining the structure of the target format; mapping the correct source data entities to the correct target representation artefacts; mapping the source data entity values to the target representation artefact values; and dealing with pre and post-amble code required in the target format.

In the following three subsections, we present the three ontologies used during the Information Realisation process. We call these Ontology Descriptors because they are a schema on which ontology instances can be created which represent specific formats or circumstances.

4.3.2 Source Data Entities Ontology Descriptor

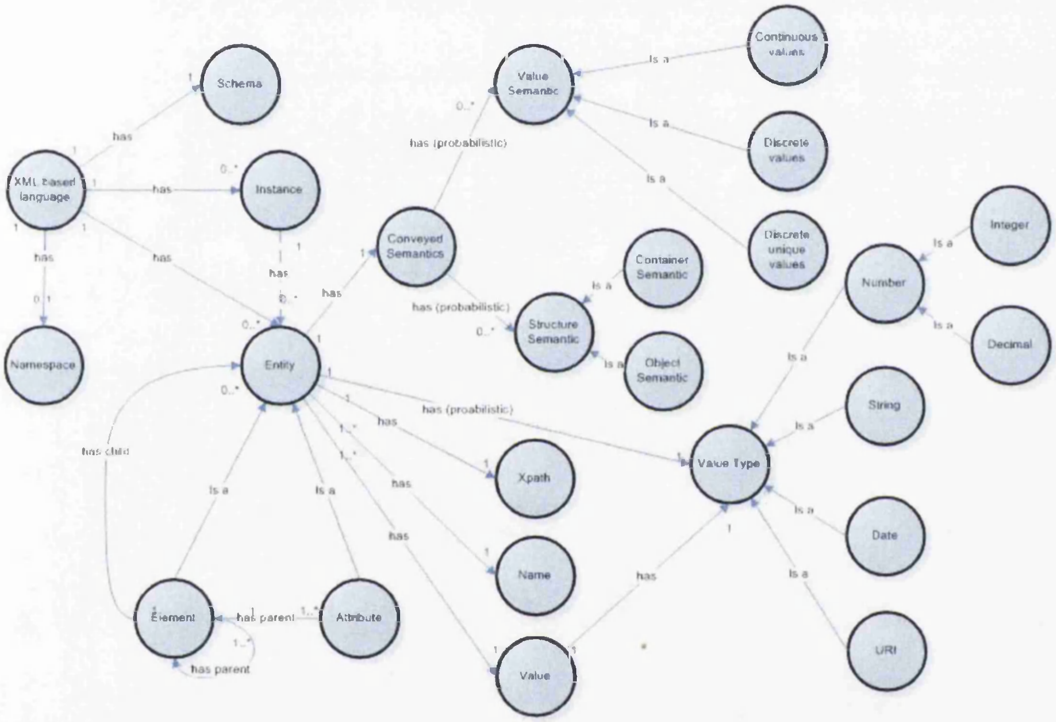


Figure 4.10: The generic XML Entities Ontology Descriptor

A Source Data Entities Ontology Descriptor is designed to capture the nature of the data stored in any particular representation format. This format may be CSV, XML, RDF, or any other general purpose representation format.

The ontology shown in figure 4.10 is designed to capture the nature of data stored in an XML formats. The ontology considers XML elements and attributes as a generalised concept called an Entity. This is similar to the Layered Normal Form which is one of the XML Normal Forms [Tho01]. An Entity has a value, a name, and an XPath. An XPath is a standard means of defining where an XML entity belongs in the schema's hierarchy. An @ symbol indicates that the entity is an attribute. It also has a parent entity, and in some cases it has child entities. In this way, we can concentrate on the values and structure of the data rather than how it is represented.

Value Semantics of the Entity are either: Continuous Values (e.g., age), Discrete Values (e.g., nationality) or Discrete Unique Values (e.g., passport number). Value Semantics may have a probability attached to it because the system may not be able to give a definitive Value Semantic categorisation without user intervention. The Structure Semantic indicates whether the Entity is a Container or an Object:

- Container is an Element which has child elements. It may have attributes, but usually has no value.
- Object is an Element which has no child elements. It may have attributes and/or a value.

4.3.3 Representation Artefacts Ontology Descriptor

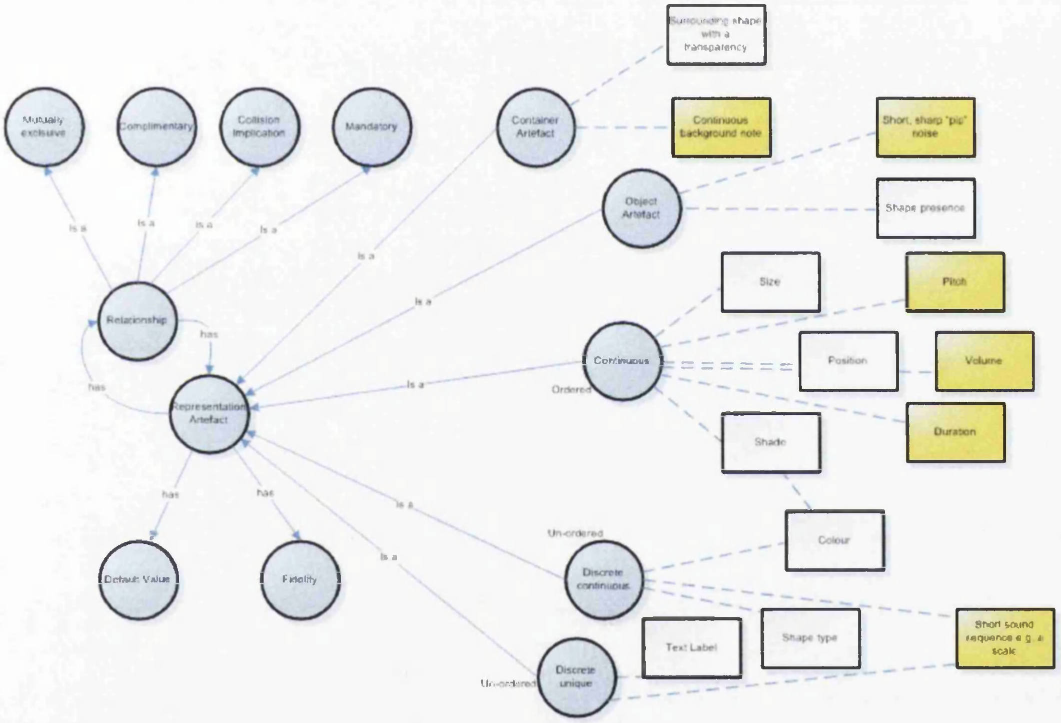


Figure 4.11: The Representation Artefacts Entities Ontology Descriptor. Circles represent concepts, rectangles represent instances (artefacts). Graphical artefacts are represented by white (non-shaded) rectangles. Audial artefacts are represented by yellow (shaded) artefacts.

Each target representation language will have its features categorised as Representation Artefacts. In figure 4.11, we give examples of each Representation Artefact. White (non-shaded) rectangles represents Graphical artefacts, yellow (shaded) rectangles represent Audial artefacts.

4.3.4 Target Environment Ontology Descriptor

This ontology is shown in figure 4.12 and is made up of:

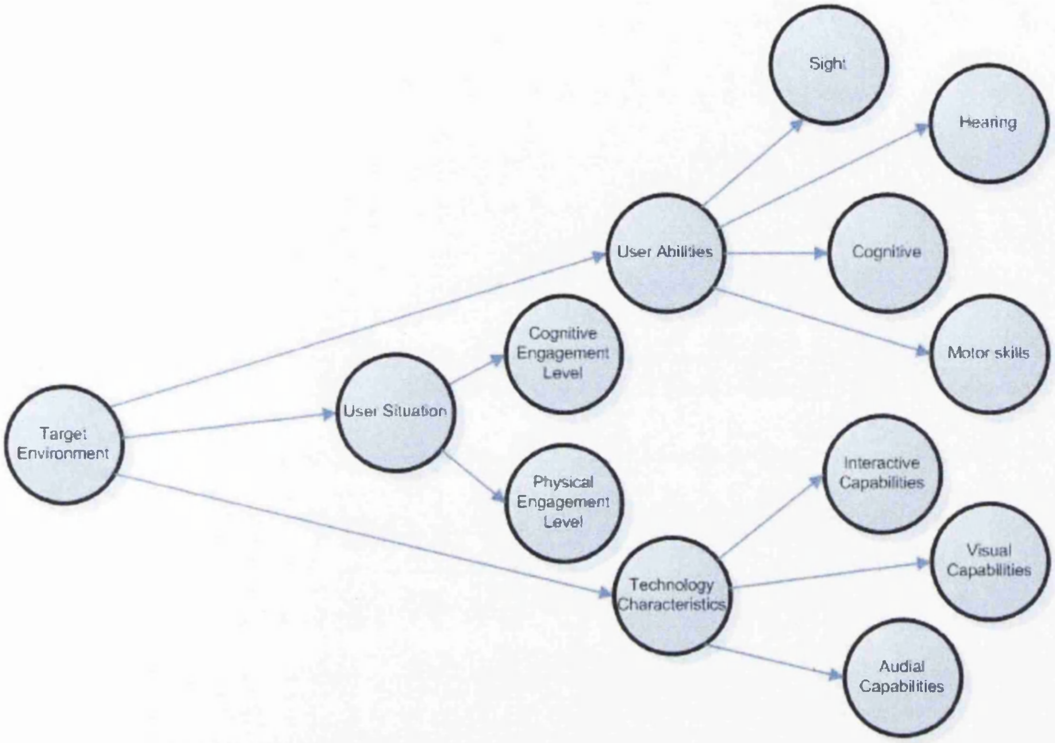


Figure 4.12: The Target Environment Entities Ontology Descriptor

User Abilities The sensory abilities of the user (sight, hearing, cognitive, motor control, etc).

User Situation The situation that the user will be in when presented with this target representation. For example, driving a car (cognitively engaged, physically engaged), or using a computer terminal (cognitively available, physically available).

Technology Characteristics This includes details of the capabilities of the target medium. For example, a computer terminal is interactive and can be provided with a visual representation. However, a car's sound system is non-interactive and must be provided with an audio representation.

4.3.5 Information Realisation Proof of Concept

The Proof of Concept (PoC) application takes a source file and uses Information Realisation techniques to produce a target representation. In the PoC, the target representation is set as a Graphical representation (in this case SVG). Therefore, it does not demonstrate the application of the Target Environment ontology. The following features are demonstrated:

1. Analysis of the source file to categorise Value Semantics and Structure Semantics.
2. Mapping of XML Entities in the source format to Representation Artefacts in the target format.

3. Mapping of source entity values to representation artefacts values.
4. Transformation of source entity values to representation artefact values based on target environment characteristics.
5. Supplying a key of mapped entities and artefacts for the final representation.

4.3.5.1 Class Design

The Proof of Concept class diagram is show in figure 4.13. The system's classes are in 4 groups:

Main controller class This class is the controller class for the PoC. It handles file input and output and error handling.

Entity classes These classes hold the data from the source file format. EntitySet handles the analysis of the source file entities (value and structure). This set of classes contains:

1. EntitySet
2. Entity
3. EntityValueSet
4. EntityValue

Artefact classes These classes hold the data from the target file format. They handle the catgorisation of the target file representation artefacts. This set of classes contains:

1. ArtefactSet
2. Artefact
3. ArtefactValueSet
4. ArtefactValue

Artefact to Entity classes These classes handle the mappings between Entities and Representation Artefacts. It also handles the mapping and translation of the values of these entity and representation artefacts. This set of classes contains:

1. ArtefactToEntitySet
2. ArtefactToEntity
3. ArtefactValueToEntityValueSet
4. ArtefactValueToEntityValue

In the next section, we show the results of using the Proof of Concept to produce an SVG visualization from a sports fans dataset. We also show tabularly the data held in the ontology instances.

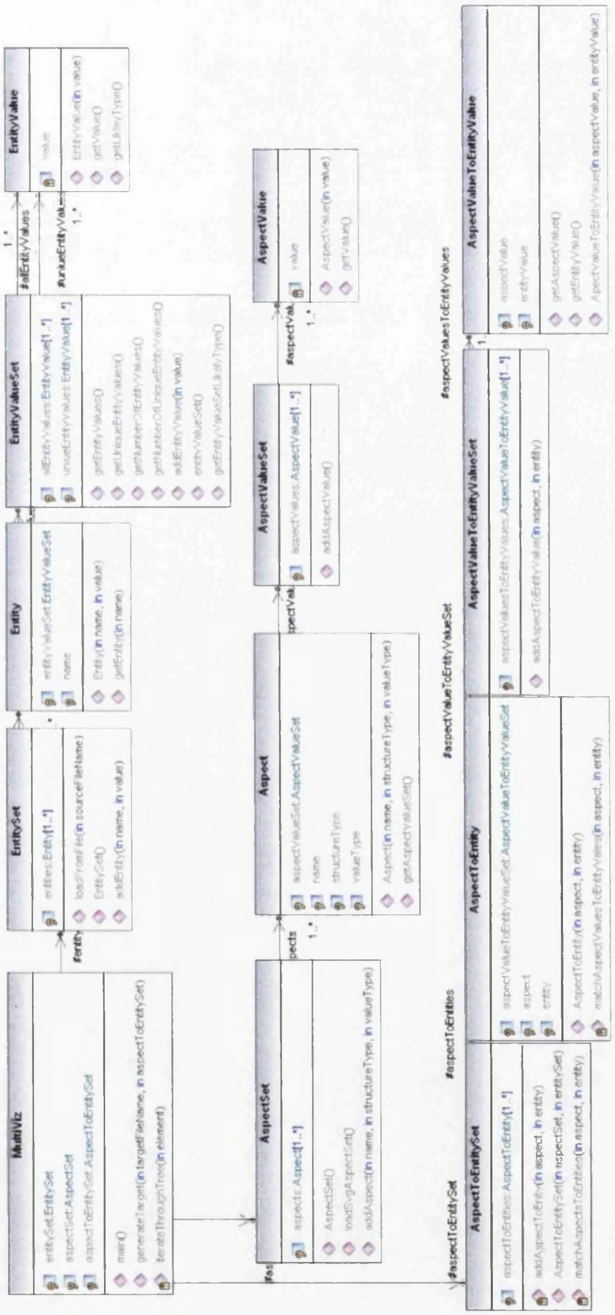

MultiViz
entitySet:EntitySet
aspectSet:AspectSet
aspectToEntitySet:AspectToEntitySet
main()
generateTarget(in targetFileName, in aspectToEntitySet)
iterateThroughTree(in element)
#entity
EntitySet
entities:Entity[1..*]
loadFromFile(in sourceFileName)
EntitySet()
addEntity(in name, in value)
Entity
entityValueSet:EntityValueSet
name
Entity(in name, in value)
getEntity(in name)
EntityValueSet
allEntityValues:EntityValue[1..*]
unuieEntityValues:EntityValue[1..*]
getEntityValues()
getUniqueEntityValues()
getNumberOfEntityValues()
getNumberOfUniqueEntityValues()
addEntityValue(in value)
entityValueSet()
getEntityValueSetLikelyType()
#allEntityValues
1..*
#uniueEntityValue
1..*
EntityValue
value
EntityValue(in value)
getValue()
getLikelyType()
#as
pects
1..*
AspectSet
aspects:Aspect[1..*]
AspectSet()
loadSvgAspectSet()
addAspect(in name, in structureType, in valueType)
Aspect
aspectValueSet:AspectValueSet
name
structureType
valueType
Aspect(in name, in structureType, in valueType)
getAspectValueSet()
pectVal
AspectValueSet
aspectValues:AspectValue[1..*]
addAspectValue()
#aspectVal
1..*
AspectValue
value
AspectValue(in value)
getValue()
#aspectToEntitySet
AspectToEntitySet
aspectToEntities:AspectToEntity[1..*]
addAspectToEntity(in aspect, in entity)
AspectToEntitySet(in aspectSet, in entitySet)
matchAspectsToEntities(in aspect, in entity)
#aspectToEntities
AspectToEntity
aspectValueToEntityValueSet:AspectValueToEntityValueSet
aspect
entity
AspectToEntity(in aspect, in entity)
matchAspectValuesToEntityVales(in aspect, in entity)
#aspectValueToEntityValueSet
AspectValueToEntityValueSet
aspectValuesToEntityValues:AspectValueToEntityValue[1..*]
addAspectToEntityValue(in aspect, in entity)
#aspectValuesToEntityValues
1..
AspectValueToEntityValue
aspectValue
entityValue
getAspectValue()
getEntityValue()
ApectValueToEntityValue(in aspectValue, in entityValue)

Figure 4.13: Class Diagram for the Information Realisation Proof of Concept.

4.3.6 Example - Sports Fans to SVG

We demonstrate the Proof of Concept using a worked example. The example concerns a set of sports fans represented in an XML file. The source data consists of 26 people (only 3 shown here) with their name, age, tallness (tallness is used rather than height to avoid confusion with the SVG attribute of the same name), nationality, scarves (number of scarves they own) and games (number of games they have been to). The source data is detailed below:

```
<fans>
    <person name="alice" age="28" tallness="1.41" nationality="welsh" scarves="2"
     games="19" />
    <person name="bob" age="37" tallness="1.02" nationality="scottish" scarves="4"
     games="33" />
    <person name="colin" age="16" tallness="1.84" nationality="irish" scarves="6"
     games="8" />
    ...
</fans>
```

The process begins by analysing the Source Data (section 4.3.1, stage 1). This results in an instantiated XML Entities ontology. The data is summarised in Table 4.1. Note that the underlined parts of the XPath represents the Entity Name. In the XPath standard, an @ symbol is used to indicate that the entity is an attribute (as opposed to an element).

XPath and Entity Name	Element / Attribute	Value type	Value Semantic	Structure
fans	element			Container (root)
fans/person	element			Object
fans/person/@name	attribute	text	Discrete unique	
fans/person/@age	attribute	numeric	Continuous	
fans/person/@tallness	attribute	numeric	Continuous	
fans/person/@nationality	attribute	text	Discrete	
fans/person/@scarves	attribute	numeric	Continuous	
fans/person/@games	attribute	numeric	Continuous	

Table 4.1: A summary of the information held in the XML Entities Ontology Instance

We then perform the Analyse Target Environment stage (Section 4.3.1, stage 2). In this example, the user has full sight, hearing, cognitive and motor control abilities. The user is at a computer terminal which has full interactive and display capabilities (800 by 600 pixels screen size). The system therefore matches the Target Environment to the SVG Target Representation Format (Section 4.3.1, stage 3). The output is a Representation Artefact Ontology Instance (Section 4.3.3). The data held in the SVG ontology instance is summarised in Table 4.2.

The Proof of Concept then maps the sports fans XML Entities to the SVG Representation Artefacts (Section 4.3.1, stage 4). The mapping process is based on matching Value Type, Value Semantics and Structure Semantics. In this case, the mapping is relatively simple and is detailed in Table 4.3. It can be seen that the Entities: `age`, `tallness`, `scarves` and `games` all have the same Value Type and Value Semantic. Therefore, there is no easy way to differentiate their characteristics and making an appropriate match to the SVG target

XPath and Entity Name	Element / Attribute	Value type	Value Semantic	Structure
svg	element			Container (root)
svg/rect	element			Object
svg/rect/@x	attribute	numeric	Continuous	
svg/rect/@y	attribute	numeric	Continuous	
svg/rect/@width	attribute	numeric	Continuous	
svg/rect/@height	attribute	numeric	Continuous	
svg/rect/@fill	attribute	text	Discrete	
svg/rect/@title	attribute	text	Discrete unique	

Table 4.2: Representation Artefact Ontology Instance for SVG

artefacts. As such, in this example, we just map in the order given. In the next chapter we will see how additional characteristics are captured allowing us to make more appropriate mappings.

Sports fans XML Entity	SVG Representation Artefact
fans	svg
person	rect
age	x
tallness	y
scarves	width
games	height
nationality	fill
name	title

Table 4.3: Mapping table between sports fans XML Entities and SVG Representation Artefacts

The next stage is to Translate and Transform XML Entity Values to Representation Artefact Values (Section 4.3.1, stage 5). This stage involves the processing of value mappings. For example, the Proof of Concept cannot merely assign the fill attribute as the nationality of the person (e.g., Welsh). Instead there must be a mapping to the available values for the fill attribute. This is an example of an attribute which the user may decide to adjust if the assumption made by the Proof of Concept is incorrect. For example, if the Proof of Concept has an available choice of 10 primary fill colours, it may assign the fill value blue to the nationality value of Welsh. The user would probably want to change this to red to more accurately reflect the traditional national colour of Wales. Also, the values of age, tallness, scarves and games would need translating before being set to x, y, width and height respectively. These values would need to be scaled to the dimensions of the screen (in this case 800 by 600 pixels). Again, these are settings which the user may wish to adjust after examining the output. The final stage is to generate the Target Representation (Section 4.3.1, stage 6). This takes the Mapping Table, together with the original Source Data (the sports fans data) and generates the Target Representation in SVG.

The SVG target representation is shown in figure 4.8 and the code shown below:

```
<svg xmlns="www.w3.orgsvg <http://www.w3.org/2000/svg>">
  <rect title="alice" x="213" y="471" fill="red" width="12" height="18" />
  <rect title="bob" x="373" y="800" fill="blue" width="16" height="25" />
  <rect title="colin" x="0" y="109" fill="green" width="21" height="13" />
  ...
</svg>
```

In the above example, we have focussed on the production of a visualization example using SVG. However, as mentioned in section 4.3, Information Realisation (or Perceptualization) covers other means of conveying information besides visual graphics. The next three subsections demonstrate the sports fans dataset perceptualised in six audial, visual and textual formats.

4.3.7 Textual Formats

XHTML

XHTML is the XML compliant version of the standard web hypertext format (HTML). It is supported by all modern web-browsers. The code is shown below and a summary of the information held in the Representation Artefact Ontology Instance for XHTML is shown in table 4.4.

```
<html xmlns=www.w3.orgxhtml <http://www.w3.org/1999/xhtml> xml:lang=en lang=en>
    <head>
        <title>XHTML file</title>
    </head>
    <body>
        <table>
            <tr>
                <td>Name</td>
                <td>Age</td>
            </tr>
            <tr>
                <td>Alice</td>
                <td>28</td>
            </tr>
        </table>
    </body>
</html>
```

XPath and Entity Name	Element / Attribute	Value type	Value Semantic	Structure
html	element			Container (root)
html/@xmlns	ns attribute	uri	Namespace	
html/head	element			Container
html/head/title	element	text	Discrete	
html/body	element			Container
html/body/table	element			Container
html/body/table/tr	element			Container
html/body/table/tr/td	element	text	Discrete unique	Object

Table 4.4: Representation Artefact Ontology Instance for XHTML

RSS

RSS is a lightweight format for distributing news headlines and other web content. It is supported by web and client-based RSS readers and also some web browsers. A summary of the information held in the Representation Artefact Ontology Instance for RSS is shown in table 4.5.

```
<rss>
    <channel>
        <item>
            <title>Football: Chelsea exit 'bad luck'</title>
            <description>Jose Mourinho says the first-leg defeat at Stamford Bridge was
             the reason Chelsea were knocked out by Barcelona.</description>
            <link>news.bbc.co.uk4781672.stm
             <http://news.bbc.co.uk/go/rss/-/sport1/hi/football/teams/c/chelsea/4781672.stm>
            </link>
            <guid isPermaLink="false">news.bbc.co.uk4781672.stm
             <http://news.bbc.co.uk/sport1/hi/football/teams/c/chelsea/4781672.stm>
            </guid>
            <pubDate>Wed, 08 Mar 2006 07:44:00 GMT</pubDate>
            <category>Chelsea</category>
        </item>
    </channel>
</rss>
```

XPath and Entity Name	Element / Attribute	Value type	Value Semantic	Structure
rss	element			Container
rss/@xmlns	ns attribute	uri	Namespace	
rss/channel	element			Container
rss/channel/item	element			Object
rss/channel/item/title	element	text	Discrete unique	
rss/channel/item/description	element	text	Discrete unique	
rss/channel/item/link	element	url	Discrete unique	
rss/channel/item/guid	element	url	Discrete unique	
rss/channel/item/guid/@isPermaLink	attribute	bool	Discrete	
rss/channel/item/pubDate	element	date	Discrete unique	
rss/channel/item/category	element	text	Discrete	

Table 4.5: Representation Artefact Ontology Instance for RSS

4.3.8 Graphical Formats

SVG

SVG is 2-dimensional vector-based graphics format. It is supported natively in Firefox 1.5 and Opera 8. However, a plug-in is required for other web browsers. A summary of the information held in the Representation Artefact Ontology is shown in table 4.2.

```
<svg xmlns="www.w3.orgsvg <http://www.w3.org/2000/svg>">
    <rect y="0" fill="red" width="300" height="20" />
</svg>
```

X3D

X3D is a 3-dimensional vector-based graphics format. It can be viewed in web browsers after installing a plug-in such as Octaga [Oct08]. Octaga also exists as a standalone viewer. A summary of the information held in the Representation Artefact Ontology Instance for X3D is shown in table 4.6.

```
<X3D xmlns:xsd="www.w3.orgXMLSchema-instance">
    <Scene>
        <Transform translation='100 50 0'>
            <Shape>
                <Box size='3 2 0.1'/>
                <Appearance>
                    <Material diffuseColor='1.0 0.0 0.0'/>
                </Appearance>
            </Shape>
        </Transform>
    </Scene>
</X3D>
```

XPath and Entity Name	Element / Attribute	Value type	Value Semantic	Structure
x3d	element			Container (root)
x3d/@xmlns	ns attribute	uri		Namespace
x3d/scene	element			Container
x3d/scene/transform	element			Container
x3d/scene/transform/@translation	attribute	text	Continuous	
x3d/scene/transform/shape	element			Container
x3d/scene/transform/shape/box	element			Object
x3d/scene/transform/shape/box/@size	attribute	text	Continuous	
x3d/scene/transform/shape/appearance	element			Container
x3d/scene/t...m/shape/appearance/material	element			Object
x3d/s...e/t...m/s...e/a...e/m..l/@diffuseColor	attribute	text	Continuous	

Table 4.6: Representation Artefact Ontology Instance for X3D

4.3.9 Audial Formats

SoundML

SoundML is a simple format for representing sounds (similar to the MIDI format) in XML. A summary of the information held in the Representation Artefact Ontology Instance for SoundML is shown in table 4.7.

```
<Song Tempo="60">
  <Notes>
    <Note Duration="Quarter" Pitch="C" Octave="4" />
    <Note Duration="Quarter" Pitch="D" Octave="4" />
    <Note Duration="Quarter" Pitch="E" Octave="4" />
    <Note Duration="Quarter" Pitch="F" Octave="4" />
    <Note Duration="Quarter" Pitch="G" Octave="4" />
    <Note Duration="Quarter" Pitch="A" Octave="4" />
    <Note Duration="Quarter" Pitch="B" Octave="4" />
    <Note Duration="Quarter" Pitch="C" Octave="5" />
  </Notes>
</Song>
```

VoiceXML

VoiceXML specifies interactive voice dialogues between a human and a computer. It can be read by a voice browser such as OpenVXI. A summary of the information held in the Representation Artefact Ontology Instance for VoiceXML is shown in table 4.8.

XPath and Entity Name	Element / Attribute	Value type	Value Semantic	Structure
song	element			Container (root)
song/@tempo	attribute	numeric	Continuous	
song/notes	element			Container
song/notes/note	element			Object
song/notes/note/@duration	attribute	text	Discrete	
song/notes/note/@pitch	attribute	text	Discrete	
song/notes/note/@octave	attribute	numeric	Continuous	

Table 4.7: Representation Artefact Ontology Instance for SoundML

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns="www.w3.orgvxml <http://www.w3.org/2001/vxml>">
    <form>
        <block>
            <prompt>Name</prompt>
            <prompt>Age</prompt>
        </block>
    </form>
</vxml>
```

XPath and Entity Name	Element / Attribute	Value type	Value Semantic	Structure
vxml	element			Container (root)
vxml/@version	attribute	numeric		
vxml/@xmlns	attribute	uri	Namespace	
vxml/form	element			Container
vxml/form/block	element			Container
vxml/form/block/prompt	element	text	Discrete	Object

Table 4.8: Representation Artefact Ontology Instance for VoiceXML

4.3.10 Summary

In section 4.3 we have presented an overview of the Information Realisation process. We have shown how it can be used for presenting data in a variety of visual, audial and textual formats. The process is based on Ontology Descriptors which are used to capture the semantics of the source (e.g., sports fans represented in XML) and target (e.g., SVG, X3D) formats. Based on these Ontology Descriptors, we can attempt to produce a high quality representation of the data (i.e., the data's realisation). Despite using ontologies for capturing the semantics of the source and target formats, in section 4.3 the actual mapping occurs using an XML-based approach. This means that we are mapping XML entities (elements and attributes) directly and as such this is a Tree-centric approach.

In the next section (section 4.4), we compare two different approaches for mapping between source and target formats: Tree-centric (which we have already seen in this section as XML); and Graph-centric mapping (using Ontologies).

4.4 Tree-centric versus Graph-centric Mapping

In this section we demonstrate both mapping approaches using the same task, converting from an SVG scene to an X3D scene. These graphical description languages are commonly used for computer graphics but differ significantly in their technical design, thus presenting some non-trivial challenges for a mapping process. A technical description and comparison of SVG and X3D is given below in section 4.4.1. For Tree-centric Mapping we use Altova's MapForce [Alt08a] and for Graph-centric Mapping we will use MAFRA toolkit [MMSV02]. Altova MapForce uses XML and XSD as its core data formats - XML for source data instances and XSD for source data schemas. MAFRA toolkit uses RDF and RDFS as its core data formats - RDF for source data instances and RDFS for source data schemas.

We have chosen to test both mapping styles using a graphics-language to graphics-language example rather than attempting an information visualization task. Even though an information visualization task would be more appropriate, the software tools which we will use (Altova MapForce and MAFRA toolkit) are general purpose mapping tools and therefore do not support the complete range of features necessary for information visualization. This includes value transformation, value substitution and conditional mapping. However, the generality which both tools provide allows us to quickly and effectively evaluate the merits of Tree-centric versus Graph-centric mapping.

4.4.1 SVG and X3D : Overview

4.4.1.1 SVG

SVG (Scalable Vector Graphics) [SVG08] is a relatively recent technology which was created by the W3C's SVG Working Group. It is an XML specification for describing 2D vector graphics which are primarily static, but can be animated. It has gained a certain degree of popularity due to its native support in Mozilla Firefox. It has yet to gain the widespread adoption of other graphics extensions such as Adobe's (previously Macromedia's) Flash technology. However, SVG is particularly appropriate for testing the concepts of Visualization As Mapping due to its clean design and its ability to convey relatively complex displays. Additionally, SVG is a declarative language, whereas Adobe's Flash is based on a language called ActionScript which is procedural. Since our source XML is declarative by its very nature, it is far simpler to map between declarative languages than it is from declarative to procedural. Below is a simple scene in SVG (figure 4.15) which is rendered in Firefox and produces a rectangle and a circle. The code is shown below (line number on the left).

```
1: <svg xmlns="http://www.w3.org/2000/svg">
2:     <rect fill="red" width="300" height="200"/>
3:     <circle fill="green" r="100"/>
4: </svg>
```

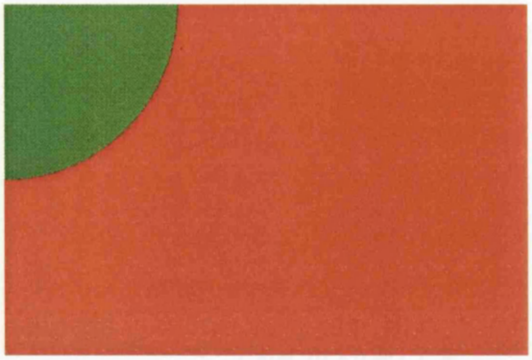


Figure 4.14: Simple shapes (rectangle and circle) depicted in SVG

SVG	**X3D**
2D	3D
Primarily static	Interactive
XML heritage	C-like syntax (via VRML)
"Good" XML design	"Poor" XML design
One value per node	Multiple values per node

Table 4.9: Key differences between SVG and X3D

4.4.1.2 X3D

X3D [Web08a] has a longer history, in particular due to its heritage as VRML. The technology can produce complex, interactive, 3D scenes. However, its design is less clean and human readable than SVG. As such, for all but the simplest of scenes, a graphic design package such as Flux [Flu08] is used. Due to its VRML heritage, the XML schema design (hierarchy and values) of X3D is particularly unfriendly for humans to read. This is partly because it contains both declarative and procedural elements. Below we show the same scene as depicted above (in SVG), but in X3D (figure 4.15), together with the code (line numbers on the left).

```
1: <X3D>
2:    <Scene>
3:       <Shape>
4:          <Appearance>
5:             <Material diffuseColor="0.0 1.0 0.0"/>
6:          </Appearance>
7:          <Sphere radius="100"/>
8:       </Shape>
9:       <Shape>
10:         <Appearance>
11:            <Material diffuseColor="1.0 0.0 0.0"/>
12:         </Appearance>
13:         <Box size="300 200"/>
14:      </Shape>
15:   </Scene>
16: </X3D>
```

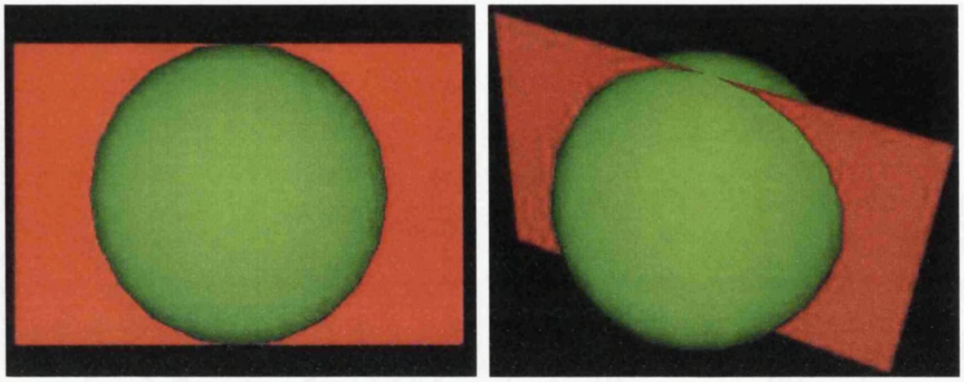


Figure 4.15: Simple shapes (box and sphere) depicted in X3D. Initial view (left), rotated view (right).

4.4.1.3 Mapping Challenges

We summarise the key differences between SVG and X3D in table 4.9. When comparing the SVG code and scene (figure 4.14) to the X3D code and scene (figure 4.15), we can identify a number of mapping challenges. We classify and describe these below. In referencing line numbers in the code we use the following format: `svg.1` is used to reference line 1 in the SVG code; `x3d.14` is used to reference line 14 in the X3D code. We also reference the level of elements in the XML tree hierarchy. For example, `x3d.1` and `x3d.16` are at Level 0 and `x3d.5` is at Level 4.

Attribute to Element (and vice-versa) We often need to map between an attribute in the source (SVG) to an element in the target (X3D). For example, in line `svg.2` we see that the `fill` attribute is set to "red". The equivalent entity in X3D is a code block containing `Appearance` and `Material` elements and a `diffuseColor` attribute (see lines `x3d.10` to `x3d.12`).

Cross Mapping between hierarchy levels In line `svg.3` we have a `circle` object which we need to translate into a `Sphere` object in X3D. The `circle` object has two attributes: a `fill` attribute and an `r` (radius) attribute. In X3D, this is represented by a `diffuseColor` attribute (line `x3d.5`) and a `radius` attribute (line `x3d.7`). It can be seen that the `diffuseColor` attribute is part of the `Material` element which is at level 4 in the hierarchy. Whereas the `radius` attribute is part of the `Sphere` element which is at level 3 in the hierarchy. It can be seen that SVG has a simple hierarchy of levels (only Level 0 and Level 1), whereas X3D has a level hierarchy from Level 0 to Level 4.

Single values per node to Multiple values per node In line `svg.2` we see that the `width` and `height` of the rectangle are specified as two separate attributes. This is translated to a single attribute, `size` in the `Box` element on line `x3d.13`. The X3D XML schema design is poor in this respect since we are representing multiple values in one XML attribute. As mentioned before, this is a throw-back to X3D's VRML heritage which uses a C-like syntax.

Value transformation The SVG specification allows colours to be specified as named colours (e.g., `red` in line `svg.2`), or RGB values in the format `rgb(255,0,0)` where the numbers are integer between 0 and 255. However, when RBG values are specified in X3D, real numbers between 0 and 1.0 are used. Therefore, we need to use values transformations when translating between SVG and X3D.

Synonyms Since SVG is a 2D scene description language and X3D is a 3D scene description language, there will obviously be different terminology used for different object types (e.g., `circle` and `sphere`). We need to hold a thesaurus of synonyms representing equivalent objects in each language. Of course, objects are not true synonyms since one is a 2D object and the other is 3D.

Semantic differences When we have objects or attributes in our thesaurus, there will be semantic differences. For example, with SVG's `rect` and X3D's `Box`. The `Box` object requires a depth attribute which the `rect` object does not possess. In this case, when translating from 2D to 3D, we must make assumptions on what depth values to use for the X3D objects. This can be stored as environment presets which apply to all objects which need a depth value.

Layout differences Note the different position offsets between the SVG (figure 4.14) and X3D (figure 4.15) scenes. If unspecified, SVG positions `rect` objects relative to the top left hand corner. In contrast, X3D positions the `box` object relative to its centre position. This presents additional mapping challenges. These can be addressed as environmental presets which are constants for the translation between any two languages.

4.4.2 Tree-centric Mapping

For XML-centric Mapping we use Altova's MapForce [Alt08a]. This is a commercially available software package from the same company which makes the popular XML editor and design tool, XMLSpy. MapForce is marketed as a graphical data mapping, conversion, and integration tool which can support bi-directional mapping.

Fundamentally, MapForce works on Tree-based principles. The main technique is based on XSLT (eXtensible Stylesheet Language Translation). The translations supported by XSLT work well for trees, but it is limited to these structures (i.e., it does not support graphs). XSLT 2.0 adds further transformation support, but ultimately, further services are still needed to support more complex transformations. Additionally, complex transformations still need procedural support (e.g., some Java code to manage the transformation). Altova MapForce allows users to create mappings between XML elements and attributes using a GUI. The package generates the XSLT which represents the transformation. However, this XSLT contains proprietary tags, and any services not offered as part of the core XSLT specification require that the translation be done within MapForce rather than a generic XSLT translator. The advantages of using a software package such as MapForce to do tree translation is that the transformations are fairly simple and can be done by a non-technical user with minimal training. However, the disadvantages are that the approach only supports tree transformations. Part of the ease of use of MapForce stems from the fact that parents of the elements can be inferred from the structure. This seems like an obvious assumption until graphs are considered. In graphs, a node can have multiple parents, therefore the user has to manually specify the aspects of the relationship between two nodes.

MapForce takes 2 input files initially. Firstly, the schema definition for the source file format (in this case SVG) expressed in XSD:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="svg" type="svg_type"/>
    <xs:element name="rect" type="rect_type"/>
    <xs:element name="circle" type="circle_type"/>

    <xs:complexType name="svg_type">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="circle" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="rect" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="rect_type">
        <xs:attribute name="fill" type="xs:string" use="required" />
        <xs:attribute name="width" type="xs:integer" use="required" />
        <xs:attribute name="height" type="xs:integer" use="required" />
    </xs:complexType>

    <xs:complexType name="circle_type">
        <xs:attribute name="fill" type="xs:string" use="required" />
        <xs:attribute name="r" type="xs:integer" use="required" />
    </xs:complexType>

</xs:schema>
```

Secondly, as an input MapForce takes the schema definition for the target file format (in this case X3D) expressed in XSD:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="X3D" type="x3d_type"/>
    <xs:element name="Scene" type="scene_type"/>
    <xs:element name="Shape" type="shape_type"/>
    <xs:element name="Appearance" type="appearance_type"/>
    <xs:element name="Material" type="material_type"/>

    <xs:element name="Box" type="box_type"/>
    <xs:element name="Sphere" type="sphere_type"/>

    <xs:complexType name="x3d_type">
        <xs:sequence>
            <xs:element ref="Scene" />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="scene_type">
        <xs:sequence>
            <xs:element ref="Shape" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="shape_type">
        <xs:sequence>
            <xs:sequence>
                <xs:element ref="Box" minOccurs="0" maxOccurs="unbounded" />
                <xs:element ref="Sphere" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
            <xs:element ref="Appearance" />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="appearance_type">
        <xs:sequence>
            <xs:element ref="Material" />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="material_type">
        <xs:attribute name="diffuseColor" type="xs:string" use="required" />
    </xs:complexType>

    <xs:complexType name="box_type">
        <xs:attribute name="size" type="xs:string" use="required" />
    </xs:complexType>

    <xs:complexType name="sphere_type">
        <xs:attribute name="radius" type="xs:decimal" use="required" />
    </xs:complexType>

</xs:schema>
```

MapForce then displays the screen shown in figure 4.16. On the left we see a module (rectangle) representing the source file format (SVG_xml_schema). Inside we can see each entity in the source schema shown in hierarchy. There are two types of icon next to each entity. A <> represents an XML element and a = represents an XML attribute. Similarly, for the target file format (X3D_xml_schema) we can see each of the schema's entities positioned hierarchically.

The user is then able to create mapping between entities by drawing lines between the arrows on the edge of the schema modules (SVG_xml_schema and X3D_xml_schema). This is

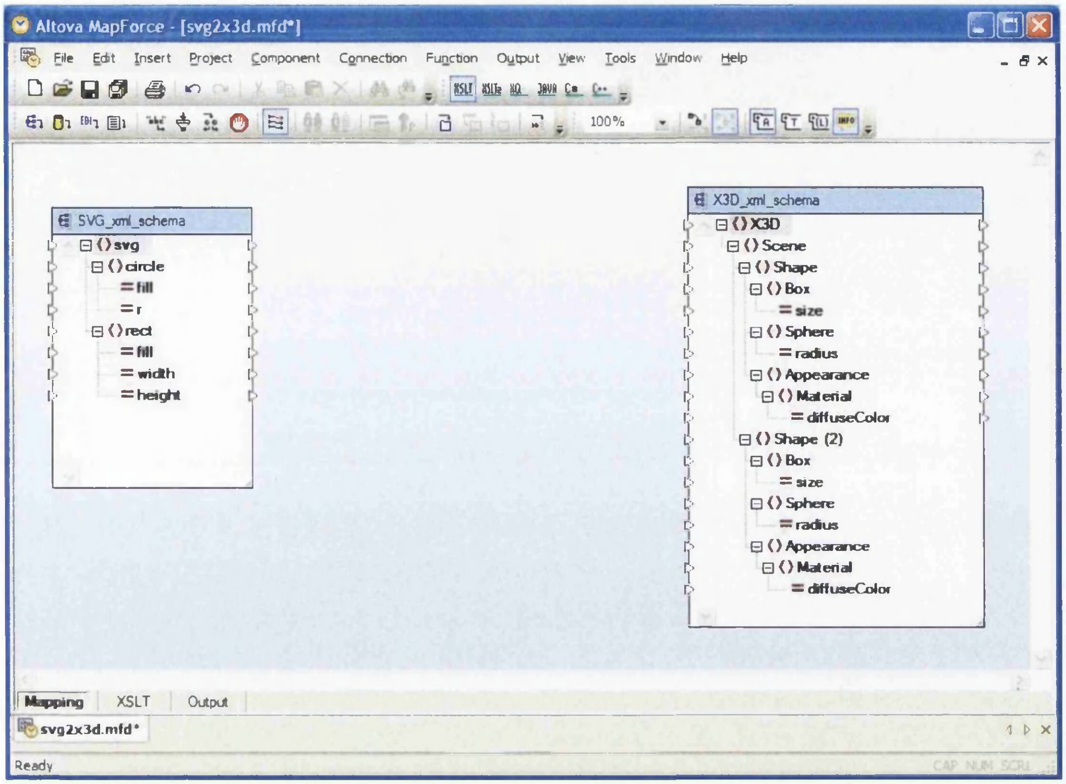


Figure 4.16: The MapForce package showing the SVG and X3D schemas ready to have mappings created.

shown in figure 4.17. Each line from Source to Target represents a mapping. For example, when an `svg` element is encountered in the source file, an `X3D` and a `Scene` element are created. If there are any values associated with a entity, they are also transferred to the target entity. Note that there are two `Shape` entities in the target schema: `Shape` and `Shape (2)`. This is because we must create a different type of `Shape` entity based on whether the source is a `circle` or a `rect`.

The `concat` (concatenate) module (bottom middle of figure 4.17) allows us to deal with the `size` attribute of the X3D `Box` needing multiple values. This is handled by performing string concatenation. Also notice that the `concat` module allows us to set the depth attribute as `10`. This can be seen as the small module in the bottom left-hand corner set at `10`.

Note that each schema module (`SVG_xml_schema` and `X3D_xml_schema`) has both an input and an output arrow for each entity. This allows translation pipelines to be created with multiple file formats involved.

Finally, we give the location of the source file itself to MapForce. In this case it is the SVG expressed in XML (see figure 4.18):

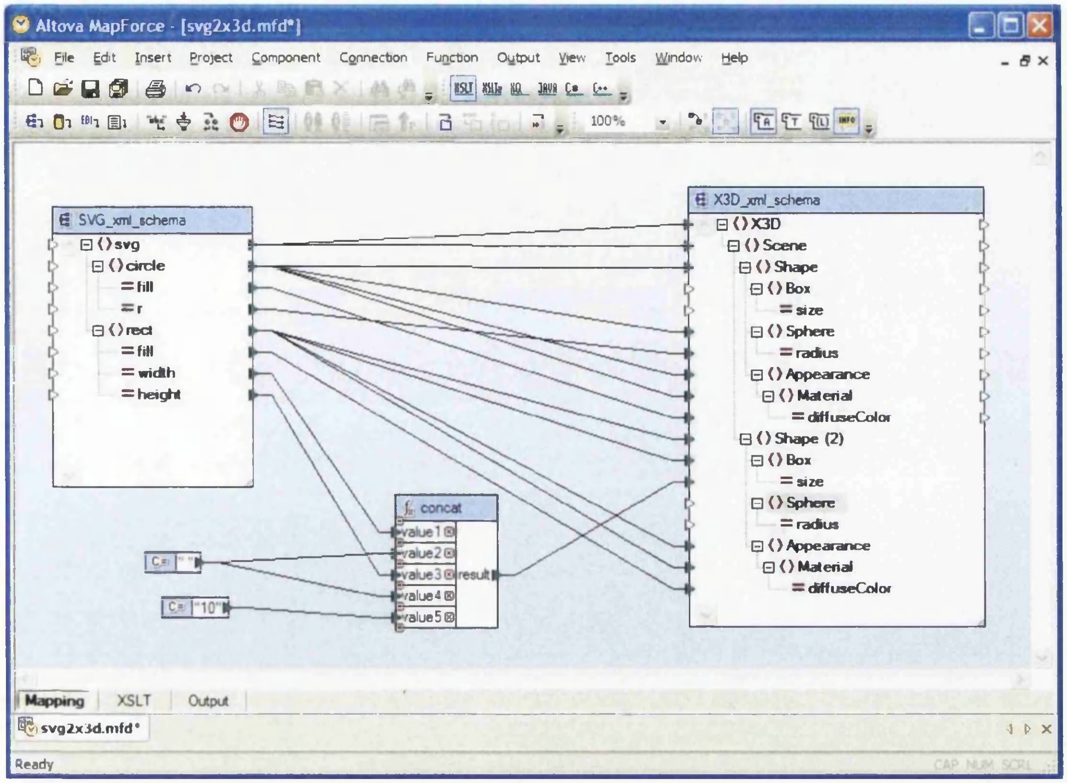


Figure 4.17: The MapForce package showing the SVG and X3D schemas with have mappings created.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg">
    <circle fill="yellow" cx="0" cy="200" r="250"/>
    <circle fill="grey" cx="300" cy="200" r="5"/>
    <circle fill="sandybrown" cx="350" cy="200" r="10"/>
    <circle fill="green" cx="400" cy="200" r="10"/>
    <circle fill="red" cx="450" cy="200" r="5"/>
    <circle fill="brown" cx="600" cy="200" r="80"/>
    <circle fill="khaki" cx="820" cy="200" r="80"/>
    <circle fill="lightblue" cx="980" cy="200" r="40"/>
    <circle fill="deepskyblue" cx="1100" cy="200" r="40"/>
    <circle fill="blue" cx="1200" cy="200" r="5"/>
</svg>
```

We can then click on the "Output" tab to see the results of the translation:

```
<X3D>
   <Scene>
      <Transform translation="600 200">
         <Shape>
            <Appearance>
               <Material diffuseColor="0.65 0.16 0.16"/>
            </Appearance>
            <Sphere radius="80"/>
         </Shape>
      </Transform>
      <Transform translation="1200 200">
         <Shape>
            <Appearance>
```

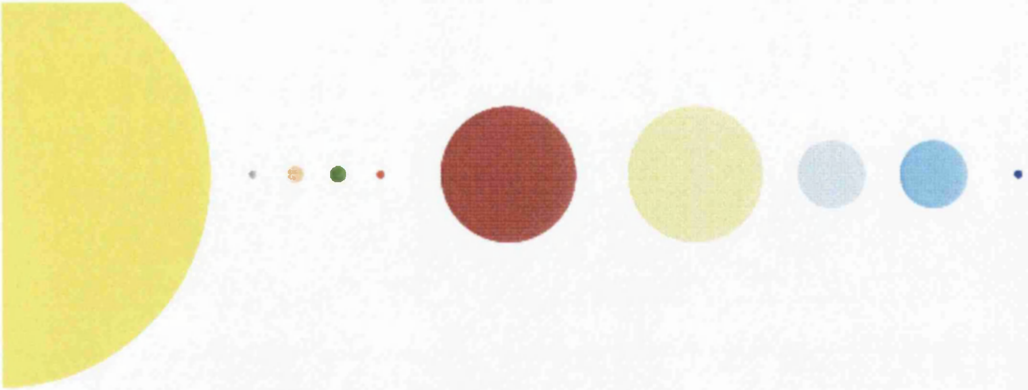


Figure 4.18: Planets depicted in SVG

```
         <Material diffuseColor="0 0 1"/>
      </Appearance>
      <Sphere radius="5"/>
   </Shape>
</Transform>
<Transform translation="400 200">
   <Shape>
      <Appearance>
         <Material diffuseColor="0 1 0"/>
      </Appearance>
      <Sphere radius="10"/>
   </Shape>
</Transform>
<Transform translation="450 200">
   <Shape>
      <Appearance>
         <Material diffuseColor="1 0 0"/>
      </Appearance>
      <Sphere radius="5"/>
   </Shape>
</Transform>
<Transform translation="300 200">
   <Shape>
      <Appearance>
         <Material diffuseColor="0.5 0.5 0.5"/>
      </Appearance>
      <Sphere radius="5"/>
   </Shape>
</Transform>
<Transform translation="0 200">
   <Shape>
      <Appearance>
         <Material diffuseColor="1 1 0"/>
      </Appearance>
      <Sphere radius="250"/>
   </Shape>
</Transform>
<Transform translation="980 200">
   <Shape>
      <Appearance>
         <Material diffuseColor="0.68 0.85 0.90"/>
      </Appearance>
      <Sphere radius="40"/>
   </Shape>
```

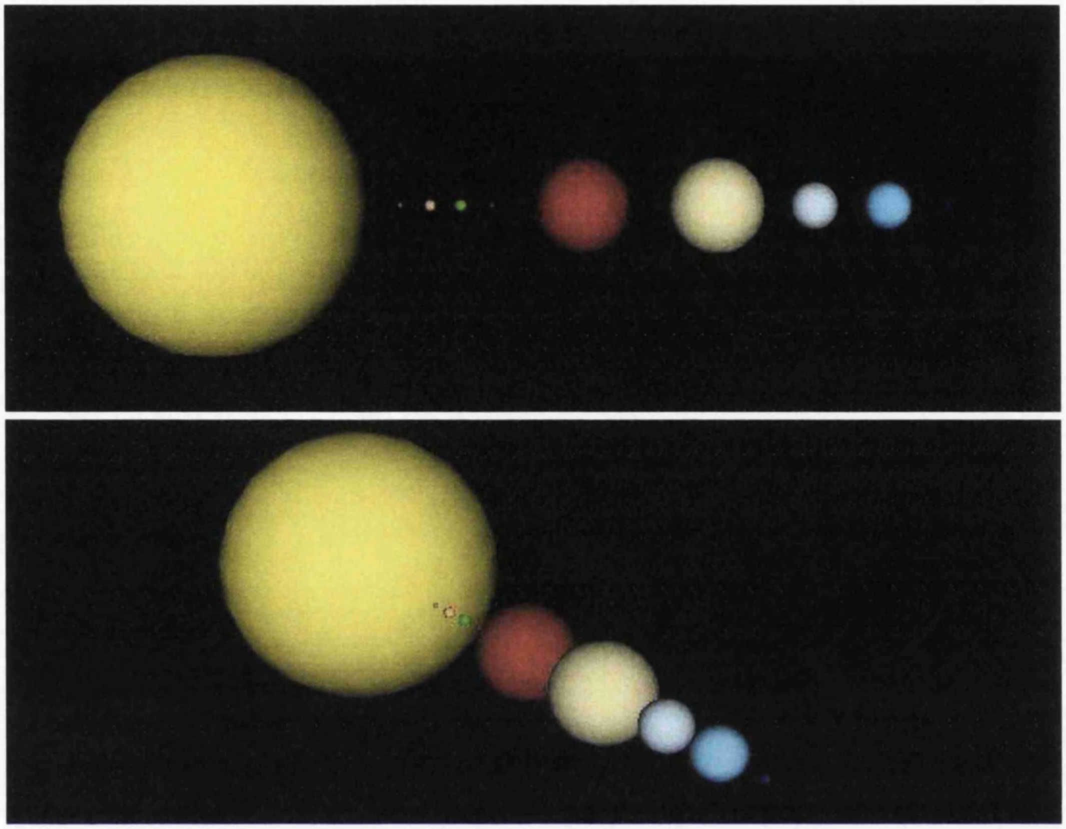


Figure 4.19: Planets depicted in X3D. Initial view (top), rotated view (bottom).

```
      </Transform>
      <Transform translation="820 200">
         <Shape>
            <Appearance>
               <Material diffuseColor="0.94 0.90 0.55"/>
            </Appearance>
            <Sphere radius="80"/>
         </Shape>
      </Transform>
      <Transform translation="1100 200">
         <Shape>
            <Appearance>
               <Material diffuseColor="0 0.75 1"/>
            </Appearance>
            <Sphere radius="40"/>
         </Shape>
      </Transform>
      <Transform translation="350 200">
         <Shape>
            <Appearance>
               <Material diffuseColor="0.96 0.64 0.38"/>
            </Appearance>
            <Sphere radius="10"/>
         </Shape>
      </Transform>
   </Scene>
</X3D>
```

The source picture is shown rendered in Firefox in figure 4.18. The target picture shown rendered in Octaga [Oct08] is shown rendered in figure 4.19.

4.4.3 Graph-centric Mapping

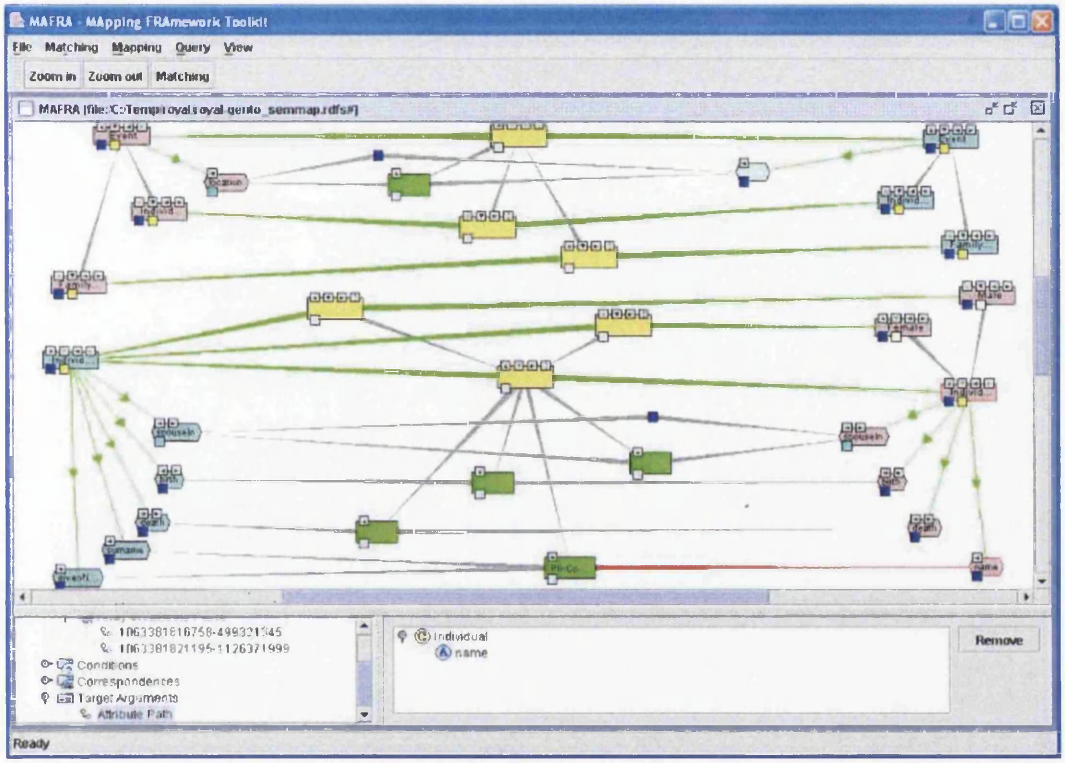


Figure 4.20: The MAFRA Toolkit GUI

For Graph-centric Mapping we will use MAFRA (MApping FRAmework) toolkit [MMSV02]. This is a research tool which has been developed from a framework into a practical toolkit. Other Graph-centric mapping toolkits exist (see section 3.3), particularly within the Ontology Mapping domain. However, MAFRA toolkit was chosen due to its flexible architecture and the author's familiarity with the tool. In figure 4.20 we can see the main MAFRA toolkit window. Source and Target concepts are shown on the far left and far right of the main window as blue or purple rectangles. Properties are shown as hexagons immediately connected to Concepts. Concept Bridges are shown as yellow rectangles towards the middle of the display. Copy Relations are green rectangles connected to Concept Bridges. Copy Relationships allow a relationship in the source to trigger a similar relationship in the target. This is explained in detail below. Finally, Conditions, Extensional Specification, Literals, File Paths and Boolean Arguments are set in the tree view shown at the foot of figure 4.20.

MAFRA toolkit is based on RDF standards. However, it can map between XML-based languages using NormKit (a component of the Harmonise Mapping Framework [FW04]) which has the ability to convert XML data into RDF and vice-versa. The mappings which are created are stored as RDF in an instance of the Semantic Bridge Ontology (SBO). Since Ontology Mapping is primarily in the domain of research tools, no standalone translator exists, so all mapping execution must be performed within MAFRA toolkit.

We have developed a simple translator between SVG and X3D using Ontology Mapping (graph-centric) techniques. The mappings were built and executed using MAFRA toolkit [MMSV02]. Two SVG object types (`circle` and `rect`) can be mapped to the the "equivalents" in X3D (`sphere` and `box`). Note that a 2D to 3D translation is being performed, i.e., this is not a 1:1 semantic mapping. The mapping aspects handled are:

1. Alignment and mapping between object types
2. Mapping between fill colours
3. Mapping between sizes (width/height and radius)
4. Mapping between object positions (x and y)
5. Default values given to unknown parameters (i.e., depth of object and Z position)

The following MAFRA toolkit features are used:

1. Concept Bridges
2. Extensional Specification
3. Property Bridges:
   (a) Copy Attribute
   (b) Copy Relation
   (c) Concatenation
   (d) Attribute Table Translation
   (e) Source Concept Specification

The mapping concepts are demonstrated using the example of transforming a simple 2D SVG house into and 3D X3D house.

4.4.3.1 Translation Process

The complete Ontology Mapping process to translate between SVG and X3D is shown in figure 4.21 and is described below:

1. **Load source ontology schema into MAFRA** The SVG XML Schema (XSD file, see section 4.4.2 for the code) is normalised and converted into an SVG Ontology Schema (RDFS file, see below for the code). The SVG Ontology Schema is then displayed in MAFRA toolkit as the Source ontology.

2. **Load target ontology schema into MAFRA** The X3D XML Schema (XSD file) is normalised and converted into an X3D Ontology Schema (RDFS file). The X3D Ontology Schema is then displayed in MAFRA toolkit as the Target ontology.

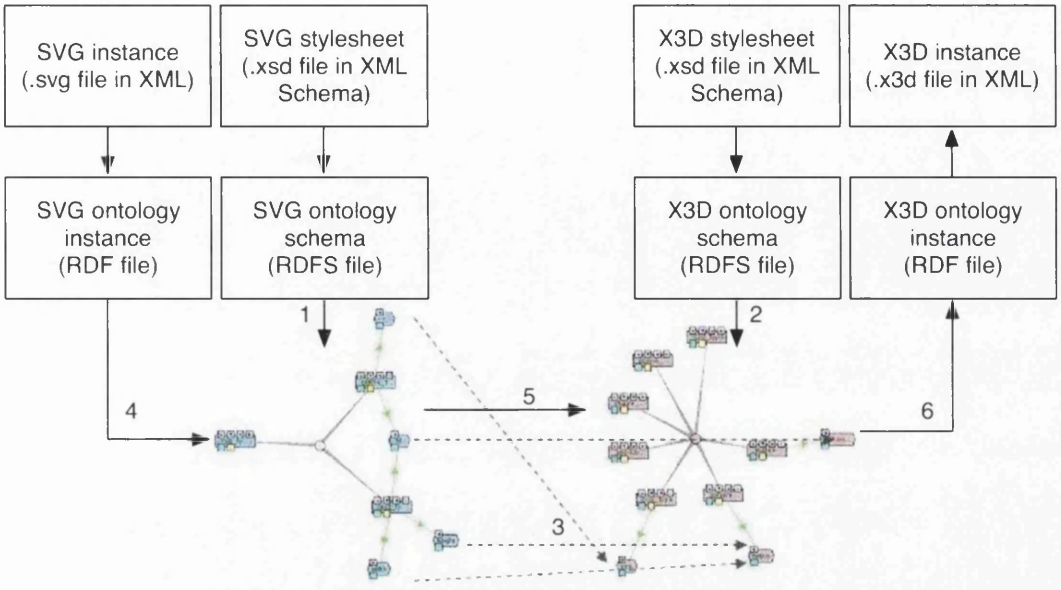

SVG instance (.svg file in XML)
SVG stylesheet (.xsd file in XML Schema)
X3D stylesheet (.xsd file in XML Schema)
X3D instance (.x3d file in XML)
SVG ontology instance (RDF file)
SVG ontology schema (RDFS file)
X3D ontology schema (RDFS file)
X3D ontology instance (RDF file)
1
2
3
4
5
6

Figure 4.21: A file input/output view of the Ontology Mapping Process for translating between an SVG and an X3D scene. The left ontology (blue) is the source, and the right ontology (purple) is the target. Firstly, the schemas are loaded (stages 1 and 2). Then, the user creates mappings between the concepts in the two schemas (stage 3). Next, the source instance is loaded (stage 4). Then, the source instance is translated into a target instance (stage 5). Finally, the target instance is output (stage 6).

3. **Create mappings** Using MAFRA toolkit, the user produces mappings between objects in the Source and the Target ontologies by creating Semantic Bridges. The types of Semantic Bridges and their use are described at a later stage. The user-created mappings are saved as a set of mappings described in RDF format.

4. **Load source ontology instance** The SVG source (SVG file) is normalised and converted into an SVG ontology instance (RDF file). This is then loaded into MAFRA toolkit ready for translation to begin.

5. **Execute mappings** The translation is performed when MAFRA toolkit takes the source ontology instance (SVG represented in RDF) and uses the user-defined mappings to create a target ontology instance (X3D represented in RDF).

6. **Receive target ontology instance** The X3D ontology instance (RDF file) is normalised and converted into a X3D target (X3D file). The user can then view the X3D file in a suitable renderer.

The SVG Ontology Schema (RDFS file) is shown below:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>

<?include-rdf logicalURI="http://kaon.semanticweb.org/2001/11/kaon-root" physicalURI=
 "jar:file:/C:/mafra-toolkit/3rdparty/kaon/kaonapi.jar!/
 edu/unika/aifb/kaon/api/res/kaon-root.xml"?>

<rdf:RDF xml:base="file:/C:/Mafra_Examples/SVG_X3D_Examples/SVG_to_X3D/
  svg_to_x3d_source_model.rdfs"
    xmlns:rdf="&rdf;"
    xmlns:rdfs="&rdfs;">

<rdf:Property rdf:ID="circle">
    <rdfs:domain rdf:resource="#svg_type"/>
    <rdfs:range rdf:resource="#circle_type"/>
</rdf:Property>
<rdfs:Class rdf:ID="circle_type">
    <rdfs:subClassOf rdf:resource="http://kaon.semanticweb.org/2001/11/kaon-lexical#Root"/>
</rdfs:Class>
<rdf:Property rdf:ID="fill">
    <rdfs:domain rdf:resource="#circle_type"/>
    <rdfs:domain rdf:resource="#rect_type"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="height">
    <rdfs:domain rdf:resource="#rect_type"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="r">
    <rdfs:domain rdf:resource="#circle_type"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="rect">
    <rdfs:domain rdf:resource="#svg_type"/>
    <rdfs:range rdf:resource="#rect_type"/>
</rdf:Property>
<rdfs:Class rdf:ID="rect_type">
    <rdfs:subClassOf rdf:resource="http://kaon.semanticweb.org/2001/11/kaon-lexical#Root"/>
</rdfs:Class>
<rdf:Property rdf:ID="svg">
    <rdfs:range rdf:resource="#svg_type"/>
</rdf:Property>
<rdfs:Class rdf:ID="svg_type">
    <rdfs:subClassOf rdf:resource="http://kaon.semanticweb.org/2001/11/kaon-lexical#Root"/>
</rdfs:Class>
<rdf:Property rdf:ID="width">
    <rdfs:domain rdf:resource="#rect_type"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>

</rdf:RDF>
```

4.4.3.2 Creating Mappings

Concept Bridges The root element of SVG (`svg_type`) and the root element of X3D

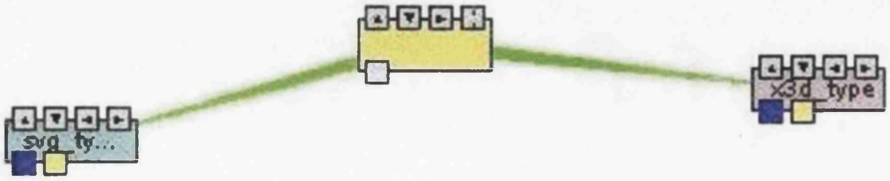


Figure 4.22: Concept Bridges

(x3d_type) have a Concept Bridge. When a svg_type element is encountered in the source file, an x3d_type must be created in the target file.

Extensional Specification When a svg_type element is encountered, a scene_type

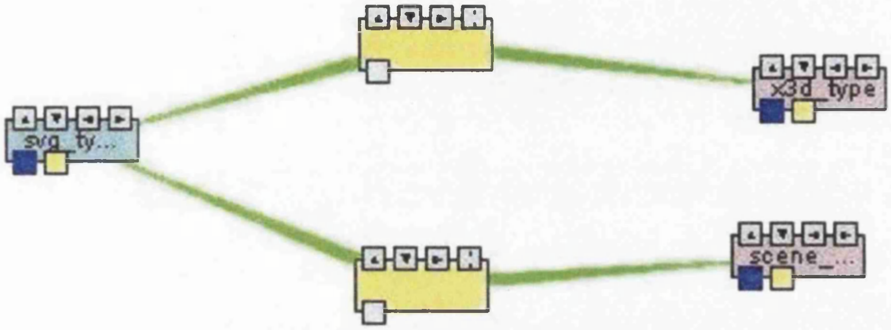


Figure 4.23: Extensional Specification

must also be created in the target file. In this case, there is an ambiguity in the system. MAFRA does not know the conditions which results in the creation of either x3d_type or scene_type.

In this case Extensional Specification is used to clarify the conditions which need to be met before the mapping occurs. For each Concept Bridge between two objects, there must be a unique Extensional Specification. It is acceptable for one of the Extensional Specifications to be not present.

The Extensional Specification between svg_type and x3d_type is not present. The Extensional Specification between svg_type and scene_type is the presence of the cirlce_type child element of svg_type. This is specified textually below. The identifier (i-1139247378719-637570132) seen in figure 4.24 is automatically created by MAFRA toolkit and represents a unique identifier for the RDF node representing the Extensional Specification.

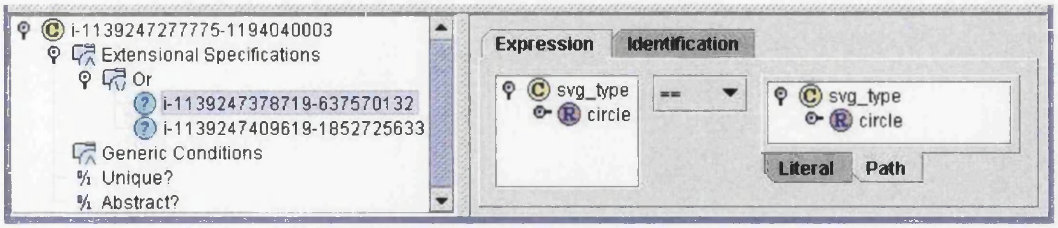

i-1139247277775-1194040003
Extensional Specifications
Or
i-1139247378719-637570132
i-1139247409619-1852725633
Generic Conditions
Unique?
Abstract?
Expression
Identification
svg_type
circle
==
svg_type
circle
Literal
Path

Figure 4.24: Extensional Specification Condition

Copy Relation In the target X3D, we need to specify that x3d_type is a parent element of scene_type. We do this by using a Copy Relation. This states that whenever there is a relationship between two objects in the source ontology, a relationship must be created between two specified objects in the target ontology. In the example above, a relationship

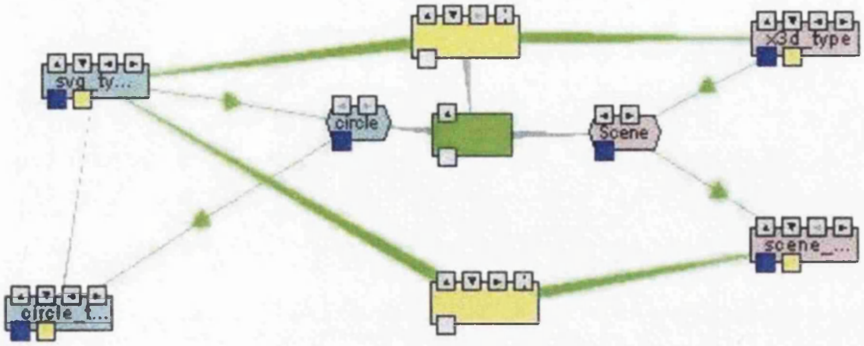


Figure 4.25: Copy Relation (green rectangle). This indicates that a relationship between svg_type and circle_type must result in a relationship between x3d_type and scene_type

between svg_type and circle_type results in a relationship being created between x3d_type and scene_type.

4.4.3.3 Example

The original source SVG file is shown in figure 4.26. Using the MAFRA toolkit and the Graph-centric (i.e., Ontology Mapping) techniques described in this section, the SVG file is translated into the X3D file which is shown rendered in figure 4.27.

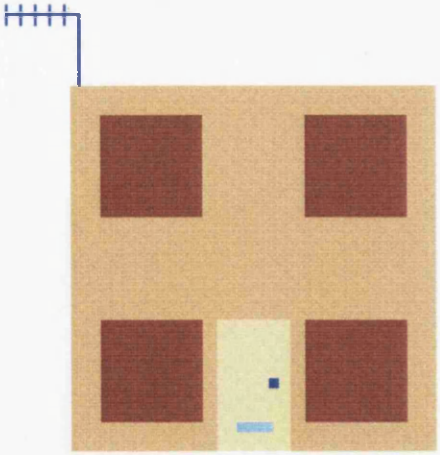


Figure 4.26: A simple house depicted in SVG.

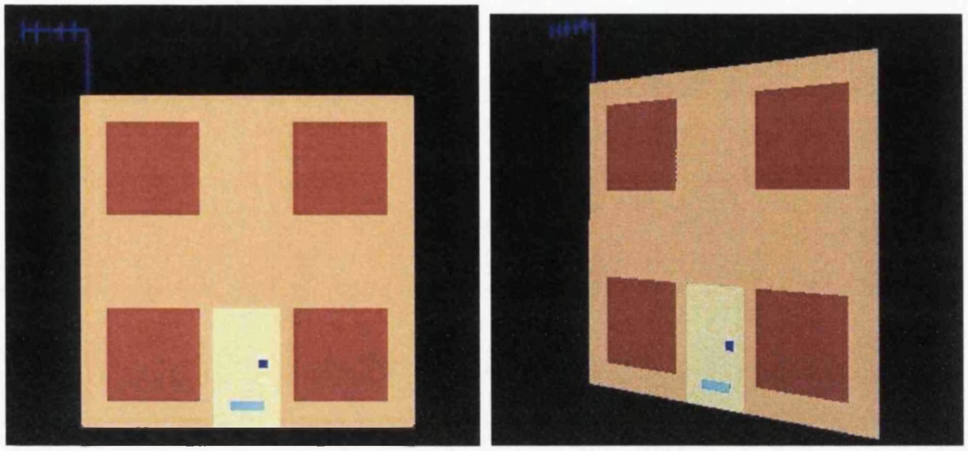


Figure 4.27: A simple house depicted in X3D. Initial view (left), rotated view (right).

	Tree-centric	Graph-centric
Translation software	Altova MapForce	MAFRA toolkit
Expression syntax	XML	RDF
Schema	XSD	RDFS
Data structure	Tree	Graph
Translation specification	XSLT	SBO

Table 4.10: XML and Ontology-centric Mapping Summary

4.4.4 Comparison Summary : Tree and Graph-centric Mapping

Both Tree and Graph-centric Mapping are able to produce SVG to X3D mappings with similar results. Table 4.10 gives an overview of the differences between Tree and Graph-centric Mapping.

The advantages of Tree-centric mapping are:

Ease of use Most modern Tree-centric mapping toolkits are XML-based and are commercially available as mature products. This means that they are robust and mappings are relatively easy to create.

Structural Inference Due to the tree-based nature of XML, we can infer parental links. This reduces the work burden on the user.

The disadvantages of Tree-centric mapping are:

Tree-based Structures Only Many data structures do not exhibit tree-based structures and are more naturally seen as graph-based structures. As such, a Tree-centric solution is not easily able to cope with such structures.

Design ambiguity Since XML can express tree-based structures using a combination of different expressions (element value, child elements, and attributes), there is ambiguity in choosing the most appropriate expression to use. For example, there is no formal definition of the semantics of using a child element for a value versus using an attribute. This provides an additional overhead in the mapping process.

The advantages of Graph-centric mapping are:

Graph-based Structure Support Graph-centric (Ontology) mapping supports more sophisticated and flexible structures than Tree-centric mapping by providing a higher level of abstraction.

Design purity Since ontologies express a graph structure very simply and without semantic ambiguity (c.f. elements and attributes in XML), it is very clear how concepts, properties and relations relate to the formal aspects of nodes, values and edges.

The disadvantages of Graph-centric mapping are:

Complexity Graph-based mapping is more complex than tree-based mapping. There are more aspects which the user needs to consider such as parental links and copy relationships.

Maturity Graph-centric mapping is still in the domain of research tools (for example, no graph-based transformation language has been recommended by the W3C). As such, tools are yet to reach the maturity and ease of use of similar Tree-centric mapping tools.

In summary, Tree-centric mapping is more pragmatic in that it allows the user to deal with the underlying XML structure of source and target formats as is. This is beneficial if the XML is well designed (e.g., SVG). However, for poorly designed XML formats (e.g., X3D), this becomes a burden since the user must understand the idiosyncrasies of the design as well as working out an appropriate mapping.

In contrast, Graph-centric mapping is more theoretical and formal in that the user thinks in terms of the mathematical notions of a graph structure (i.e., nodes, values and edges). There is very little ambiguity in how a graph-based structure can be expressed in an ontology. As such, once the user has understood the principles of ontology mapping, they can focus entirely on the structures involved and create effective mappings.

Based on the above, our comparison of Tree-centric versus Graph-centric mapping is inconclusive in terms of which is more appropriate for the purposes of Information Visualization. Both techniques have their merits, but since we have not evaluated each technique with Information Visualization tasks, further work is needed before we can recommend a technique. We discuss this more fully in the Summary (section 4.5).

4.5 Summary

In the first part of this chapter, we have described the three approaches to Automatic Visualization as: Type-Constrained; Case-based; and a Hybrid Approach. It seems that Type-Constrained and a Hybrid Approach can potentially yield effective results without the significant infrastructure investment which a Case-based approach would require. Therefore we choose take these two approaches (Type-Constrained and a Hybrid Approaches) forward into the implementations we produce in chapter 5 (VizThis) and chapter 6 (SemViz).

	VizThis	**SemViz**
Automatic Mapping Approach	Type-Constrained	Hybrid
Perceptual Representation Model	Information Realisation	Information Realisation
Mapping Data Structure	Tree-Centric	Graph-Centric
Chapter	5	6

Table 4.11: VizThis and SemViz approaches.

In the second part of this chapter, we have described the Information Realisation Model, a general mapping model for creating perceptual representations of semantically-rich information. We will use this model as the basis for the tool implementations we produce in chapter 5 (VizThis) and chapter 6 (SemViz).

In the third part of this chapter, we have shown the use of two different mapping tools to demonstrate Tree-centric and Graph-based mapping. We have compared the merits of each approach by building a graphical language translator (SVG to X3D) and concluded that both approaches have their advantages and disadvantages. Although enlightening, this comparison does not provide a sufficiently definitive answer to the question of whether Tree-centric or Graph-based mapping is most suitable for the Visualization as Mapping concept. In the next two chapters, we therefore set out to investigate this question by building two separate mapping toolkits which are specifically geared towards Information Visualization rather than general translation (e.g., SVG to X3D). The aim of this is to ascertain which approach produces the better result. In chapter 5 we implement a Tree-centric mapping toolkit called VizThis. In chapter 6 we implement an Graph-centric mapping tool called SemViz. VizThis will use a Type-Constrained approach to automatic mapping and SemViz will use a Hybrid Approach to automatic mapping. This is summarised in table 4.11.

Chapter 5

VizThis : A Tree-centric Mapping Toolkit for Information Visualization

Contents

5.1 Introduction

In the previous chapter, we discussed two different paradigms for information mapping: Tree-centric mapping and Graph-centric mapping. In this section, we take the concepts of Tree-centric mapping and apply them to the problem of Information Visualization. We do this through the implementation of a visualization pipeline which uses XML end-to-end. Part of this chapter is based on work presented in the paper, "VizThis : Rule-based Semantically Assisted Information Visualization" [GSG$^+$07].

XML has become a universally popular language for information interchange. It is commonly used as a way of conveying records of information and can scale from the simplest of key-value pairs, to more sophisticated hierarchical records. In addition, XML is used by two popular graphic description languages: SVG and X3D. Since our source and target formats share the same notation, we can build a mapping toolkit which uses XML notation end-to-end. This XML-based pipeline means that we can apply the mapping techniques demonstrated by Altova MapForce in chapter 4, while adding features which are specific to the problem of visualization.

Thus the goal is to show the concepts of Visualization as Mapping (see chapter 4) using a Tree-centric approach with XML as the common data format.

5.2 VizThis: Design Objectives

It is important to have clear requirements for the scope and purpose of the VizThis tool. VizThis is a general purpose visualization toolkit which takes the general principles of XML-centric mapping and applies them to the task of information visualization. The design objectives of VizThis are to:

- Allow XML datasets to be quickly and easily visualized using any XML-based graphic description language.
- Employ the general concepts of visualization as mapping using an XML-centric approach, but build facilities which are specific to information visualization.
- Take a general approach to the source and target formats, thus not restricting the source domain or target visualization style to any particular preconception.
- Rapidly display the effects of data mapping changes on the rendered visualization, thus allowing an iterative approach to visualization creation.
- Take a pragmatic approach to source data (i.e., allow "dirty data" and XML structures which are not designed in the most elegant style).
- Allow the manual editing of source and target code.
- Facilitate automatic mapping between source and target formats, thus allowing an initial visualization to be produced quickly and facilitating an iterative development style.
- Provide an easy to use user-interface which hides (as far as possible) the underlying complexity of the mapping process.

5.3 VizThis: Pipeline Overview

In this section, we give an overview of the VizThis pipeline stages (shown in figure 5.1). The pipeline is based on the general model of Information Perceptualisation (described in section 4.3). However, to simplify the system, we only consider visual aspects and focus on automatic information visualization. Additionally, we do not consider the environmental aspects detailed in section 4.3.3.

Stage 1 - Source Data Analyser In this pipeline stage, VizThis takes the source data and analyses its structure and values in order to produce an Analysis Table. In this way, VizThis does not use or expect the source data to have an explicitly defined schema.

Stage 2 - Data Cleansing and Normalisation This stage is optional, but it allows a user to specify data filters in order to cleanse or normalise the source data. This is done by entering regular expressions. The output of this stage is a new version of the source data (called the Cleansed and Normalised Source). This new version of the source data is fed back into Stage 1 (the Source Data Analyser) so that data analysis may again be performed on the data (since the value types of the source data may be judged as being different after data cleansing and normalisation).

Stage 3 - Entity to Artefact Matcher In this stage we take the source data's Analysis Table (from stage 1) and a Representation Artefacts Characteristics table (e.g., SVG or X3D) from the Representation Store and produce an Entity to Artefact Mapping Table. The algorithm takes facts about the source (the Source Analysis Table) and target (the Representation Artefacts Characteristics) and constraints (discussed in section 5.4.3) and attempts to produce a mapping which satisfies all constraints.

Stage 4 - Translate and Transform Entity Values to Artefact Values During this stage, VizThis translates or transforms values of source data entities into the equivalent values for the target representation artefacts. A *translation* occurs when a source value is translated into a different target value (e.g., when the *country* source entity is mapped to the target artefact *color*, its value is translated from *Welsh* into *red*). A *transformation* occurs when a source value is transformed into a different target value. For example, if considering the fans dataset from section 4.3.1, when the *age* source entity is mapped to the target artefact *x*, its value is translated (scaled) from the values of *16* (the minimum age) and 61 (the maximum age) into the values of *0* (the minimum x coordinate) and *800* (the maximum x coordinate).

Stage 5 - Generate Target Representation In the final stage, VizThis takes the Entity to Artefact Mapping Table from stage 3 and the Value Mapping Table from stage 4 and produces the target file (either as SVG or X3D). VizThis must produce a target file which has the correct entity to artefact mappings and also which adheres to the schema of the target file format. The file is then rendered and shown to the user.

5.4 VizThis: Pipeline Details

In this section, we describe the VizThis pipeline stages in detail (shown in figure 5.1). To illustrate the process, a dataset of sports fans will be used. An extract of the source file is shown below:

```
<fans>
    <person name="alice" age="28" tallness="1.41m" nationality="welsh" scarves="2"
     games="19"/>
    <person name="bob" age="37" tallness="1.02m" nationality="scottish" scarves="4"
     games="33"/>
    ...
    <person name="ziggy" age="42" tallness="1.67m" nationality="english" scarves="1"
     games="20"/>
</fans>
```

VizThis was developed as a Microsoft Windows application, using Visual Studio C#. It uses an embedded Internet Explorer 6.0 browser window together with Adobe's SVG viewer [Ado08] and the Octaga X3D player [Oct08].

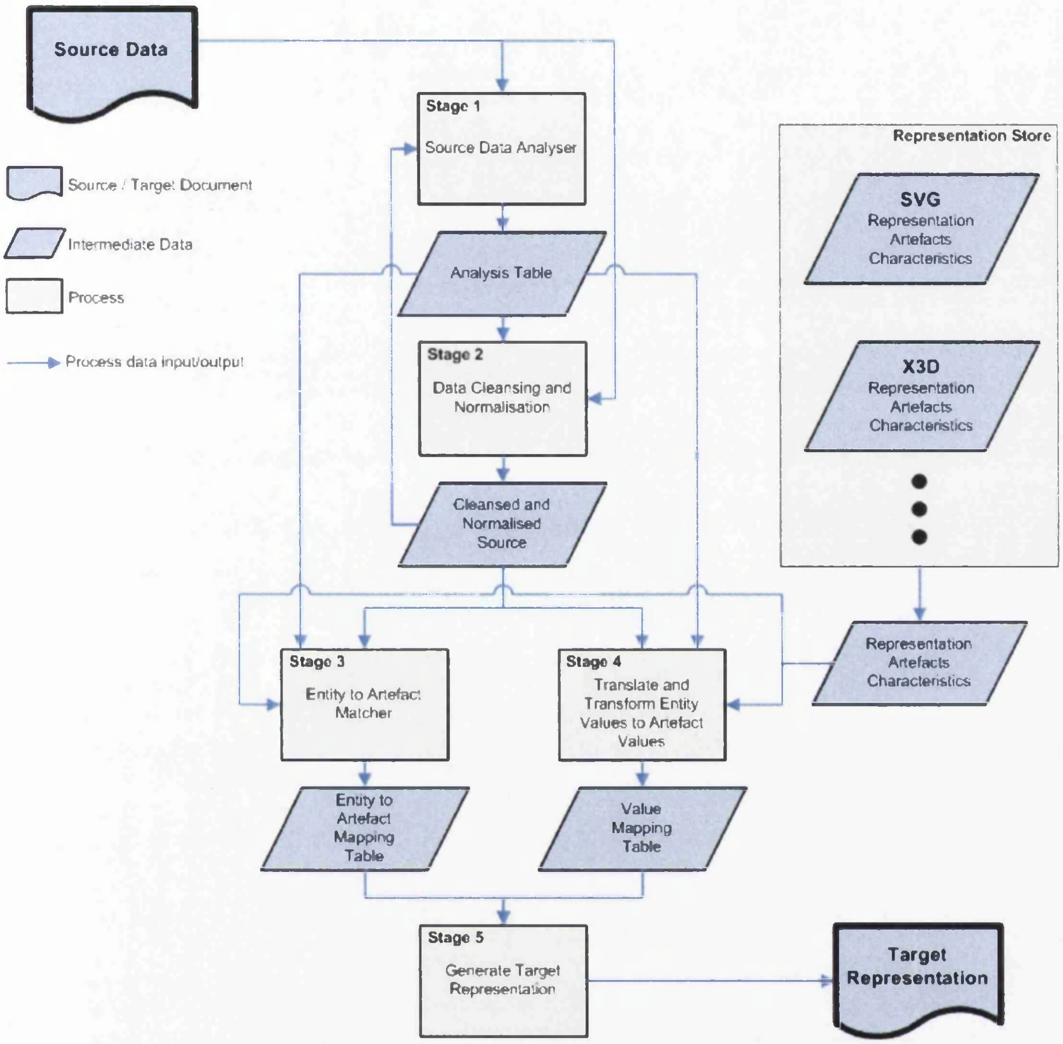

Source Data
Source / Target Document
Intermediate Data
Process
Process data input/output
Stage 1
Source Data Analyser
Analysis Table
Stage 2
Data Cleansing and Normalisation
Cleansed and Normalised Source
Representation Store
SVG
Representation Artefacts Characteristics
X3D
Representation Artefacts Characteristics
Representation Artefacts Characteristics
Stage 3
Entity to Artefact Matcher
Stage 4
Translate and Transform Entity Values to Artefact Values
Entity to Artefact Mapping Table
Value Mapping Table
Stage 5
Generate Target Representation
Target Representation

Figure 5.1: The VizThis Pipeline Stages

5.4.1 Source Data Analyser (Stage 1)

During this stage we analyse the source data in order to create an analysis table. We must analyse the nature of the source data to attempt to extract data type information. There are a number of possible ways to do this, such as using a schema processor or using existing research tools [MKFR03]. We choose an alternative approach which we think is well suited to semantically rich XML data sources.

We consider XML elements and attributes as a generalised concept called an Entity. This is the same model as we described in section 4.3.2. An Entity has a value, a name, and an XPath. The entity also has a parent entity (provided it's not the root entity), and in some cases it has child entities. In this way, we can concentrate on the values and structure of the data rather than its XML specific representation method. This abstraction of entities is modelled in figure 4.10.

For each entity, we calculate the following:

Value Type: This is the type of the data associated with the entity. It can be explicit if provided by a schema, or inferred from analysis of the data. Possible values are number, text, URL or XML namespace.

Value Category: This is a categorisation of the values in terms of the value type and any ordering. A value category is one of 3 values:

1. **Quantitative** Numeric values (integer or real) which by their very nature are ordered. For example, *age* (0 to 120) expressed as an integer.
2. **Ordered** Non-numeric values which have discrete values with an implied ordering. For example, *t-shirt-size* (S, M, L, XL, XXL). Custom types which represent ordered values can be provided to the VizThis tool.
3. **Nominal** Non-numeric values which have discrete values but with no implied ordering. For example, *name-of-person* (Bob, Mary, John).

Structure Semantic: This characterises any structural semantics conveyed by the entity and therefore only applies to entities which have child entities. It implies that the entity is representing a container. Additionally, there is a special type of container which represents the root of the XML tree. Therefore, this characterisation can have 3 values: None, Root or Container.

Uniqueness: The number of unique values when considering all entities of the same name. It is shown in table 5.1 as an absolute value and as a proportion of all records.

Minimum: This is the minimum value stored in the records (only applicable to Entity with Value type as Number).

Maximum: This is the maximum value stored in the records (only applicable to Entity with Value type as Number).

Note that even if some of these characteristics are available from the source schema, others must be inferred from the data itself. The source data entity analysis for the Fans dataset is shown in table 5.1.

XPath	**Value type**	**Value category / Structure semantic**	**Uniqueness** (abs. and proportion)	**Min**	**Max**
fans	-	Container (root)	-	-	-
fans/person	-	Container (object)	-	-	-
fans/person/@name	Text	Nominal	26 values (1.0)	-	-
fans/person/@age	Number	Quantitative	21 values (0.81)	16	61
fans/person/@tallness	Text	Nominal	23 values (0.88)	-	-
fans/person/@nationality	Text	Nominal	4 values (0.15)	-	-
fans/person/@scarves	Number	Quantitative	10 values (0.38)	1	10
fans/person/@games	Number	Quantitative	23 values (0.88)	2	42

Table 5.1: Source data entity analysis for the Sports Fans dataset (no data cleansing or normalisation)

5.4.2 Data Cleansing and Normalisation (Stage 2)

During this stage, we cleanse the data which in turn can help with the semantic type analysis which occurs during the previous stage. The Source Data Analyser is often able to inform this stage where Data Cleansing and Normalisation may be required. In fact, after this stage is complete, we feed this data back to the Source Data Analyser. This is so that the newly cleansed and normalised data can be re-analysed.

In this example dataset, we can see that the *tallness* entity has its units, metres (shown as "m") appended to the value. In this form, VizThis is not able to recognise the value as numeric and as such the visualization which is produced automatically will not represent the value using a visual representation which is good for numeric values. We can thus perform some data cleansing by applying a regular expression to the *tallness* source data entity and all its values. This process is manual in that it requires user intervention. The regular expression, `[0-9.]+` will ensure that alphabetical characters are removed from the *tallness* source data entity, thus removing the "m" and leaving a numeric value. The resulting source data is re-analysed and the new source data entity analysis is shown in table 5.2.

XPath	**Value type**	**Value category / Structure semantic**	**Uniqueness** (abs. and proportion)	**Min**	**Max**
fans	-	Container (root)	-	-	-
fans/person	-	Container (object)	-	-	-
fans/person/@name	Text	Nominal	26 values (1.0)	-	-
fans/person/@age	Number	Quantitative	21 values (0.81)	16	61
fans/person/@tallness	Number	Quantitative	23 values (0.88)	1.02	1.97
fans/person/@nationality	Text	Nominal	4 values (0.15)	-	-
fans/person/@scarves	Number	Quantitative	10 values (0.38)	1	10
fans/person/@games	Number	Quantitative	23 values (0.88)	2	42

Table 5.2: Source data entity analysis for the Sports Fans dataset (with data cleansing and normalisation). The *tallness* entity is now Numeric and Quantitative.

5.4.3 Entity to Artefact Matcher (Stage 3)

In this stage, the data entities are associated with representation artefacts. This states that a specific entity's data will be represented through a specific artefact. In order to explain this stage we must provide some more information about the components used. The system (or user) must first decide in which *representation file format* the visualization will be shown. Each representation file format consists of *representation artefacts*. These are different visual features which are used to represent aspects of the source data (the entities). Just like the source data entities, each representation artefact has a *Value Category*: Quantitative; Ordered; or Nominal. It also has a *Structure Semantic*: Container (root); Container (object); or None. Additionally, a representation artefact also has a *Variance capability* which indicates the artefacts ability to represent many different values. An artefact with a high variance capability (such as `svg/rect/@title`) can convey to the user many different values with a high degree of accuracy. However, an artefact with a low variance capability (such as `svg/rect/@width`) can only convey a small number of different values to the user. This is because if we have the range of the `svg/rect/@width` artefact going from 20 pixels to 40 pixels, then it is unlikely that a user will be able to easily differentiate between more than four values. For the SVG file representation format, the categorisations are shown in table 5.3.

In the context of representation artefacts, the three categories define the semantics which the representation artefact is capable of displaying. In this way, we have an effective method of matching *source data entities* to *target representation artefacts*.

XPath	**Value type**	**Value category / Structure semantic**	**Variance capability**		
svg	-	Container (root)	-	-	-
svg/rect	-	Container (object)	-	-	-
svg/rect/@x	Number	Quantitative	Medium	-	-
svg/rect/@y	Number	Quantitative	Medium	-	-
svg/rect/@width	Number	Quantitative	Low	-	-
svg/rect/@height	Number	Quantitative	Low	-	-
svg/rect/@fill	Text	Ordered	Low	-	-
svg/rect/@title	Text	Nominal	High	-	-

Table 5.3: Target representation artefact definitions for SVG

For every representation format we have a table which defines the capabilities each representation artefact has to handle each *Value Category*. This table is typically created by developer users of the system and would be based on the Information Visualization formalisation discussed in section 2.3. It is created when a developer adds a new representation format to the system. The first two columns of table 5.4 show the value category for the representation artefacts in SVG.

Representation Artefact		**Data Entity**		
		Value Category		
Name	Value Category	Quantitative	Ordered	Nominal
x	Quantitative	***	**	*
y	Quantitative	***	**	*
width	Quantitative	***	**	*
height	Quantitative	***	**	*
fill	Quantitative *	***	**	*
fill	Ordered ***	**	***	*
fill	Nominal **	*	**	***
title	Quantitative *	*	**	***
title	Ordered **	*	**	***
title	Nominal ***	*	**	***

Table 5.4: SVG artefact categorisation and mapping function prioritisation (*** is most favoured, * is least favoured)

Most representation artefacts will accept only one category. For example the x coordinate artefact in SVG only accepts values of category *Quantitative*. However, certain artefacts such SVG `fill` colour and `title` can accept multiple categories.

5.4.3.1 Representation Artefact Functions

For each of the representation artefacts we have a function available for each possible value category of entity (i.e., Quantitative, Ordered and Nominal). In this way, a representation artefact is able to have an entity with any value category mapped to it. However, certain value categories (of entity) are more favoured by certain artefacts (with certain value categories). This is described below.

5.4.3.2 Entity to Artefact Mapping Priorities

At this point, all entities are available to map to all artefacts. For example the x coordinate artefact in SVG has associated functions: Quantitative to x, Ordered to x and Nominal to x. Some combinations are more favourable than others. The system chooses which function to use by consulting the table (see the right-most 3 columns of table 5.4). Combinations marked *** are the most favoured option, while combinations marked * are least favoured option. For example, for the artefact x, the most favoured source data entity will be of value category Quantitative, and the least favoured source data entity will be of value category Nominal.

Most artefacts (e.g x) can only handle one value category. However, some artefacts (e.g., `fill` and `title`) can handle multiple value categories. In this case, the representation artefact has a value category which is most favoured. This is the value category with which the artefact has most semantic affinity. For example, the most semantically rich way to use `fill` is via Ordered values. And the most appropriate value category for `title` is Nominal.

For the example fans data to be visualized in SVG, table 5.5 shows the mappings.

Source Data Entity		**Target Representation Artefact**	
Name	*Value Category*	*Name*	*Value Category*
age	Quantitative	x	Quantitative
tallness	Quantitative	y	Quantitative
nationality	Nominal	fill	Nominal
scarves	Quantitative	width	Quantitative
games	Quantitative	height	Quantitative
name	Nominal	title	Nominal

Table 5.5: Entity to Artefact mappings - Fans source data mapped to SVG target representation format

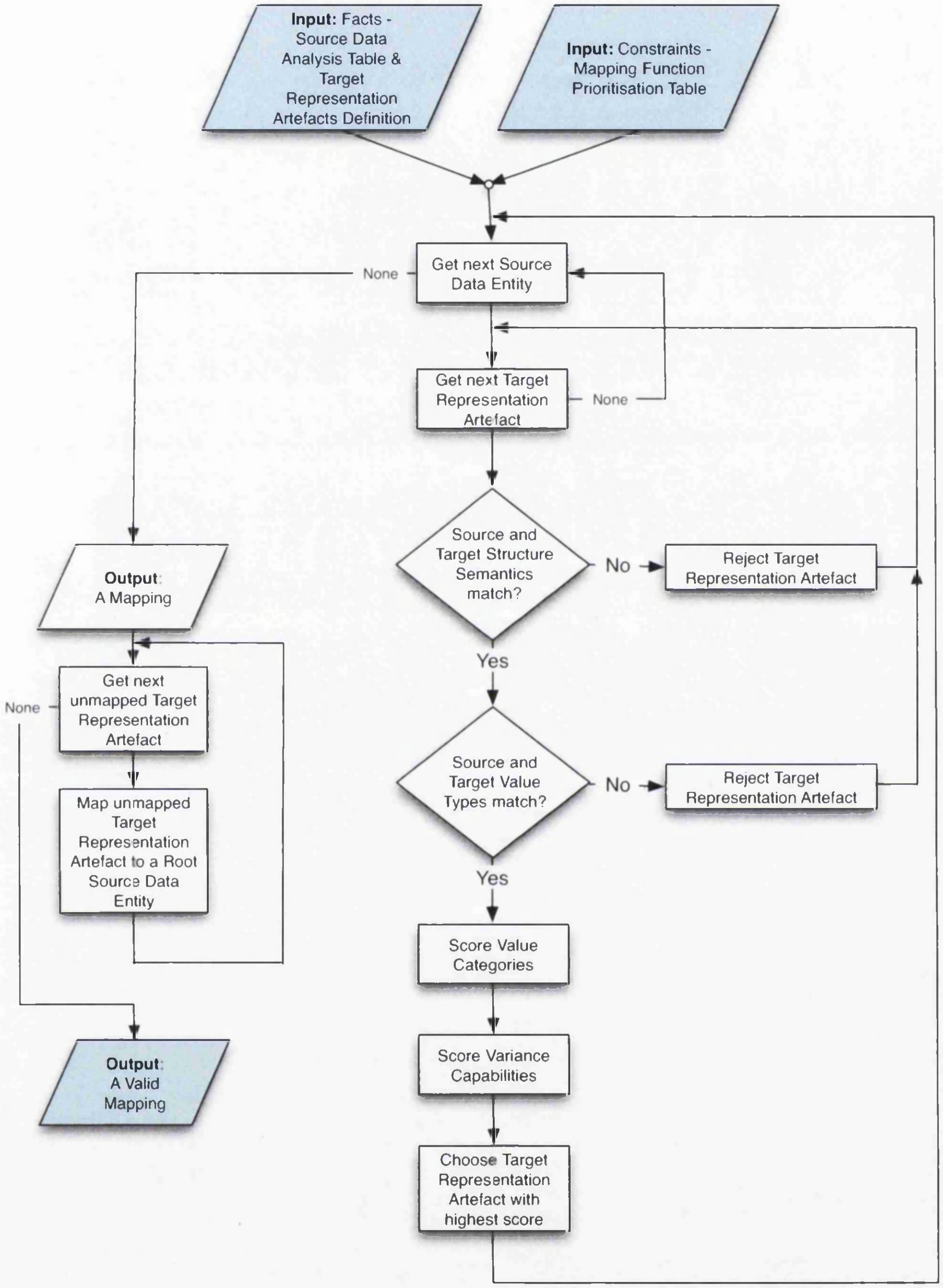

Input: Facts - Source Data Analysis Table & Target Representation Artefacts Definition
Input: Constraints - Mapping Function Prioritisation Table
Get next Source Data Entity
None
Get next Target Representation Artefact
None
Source and Target Structure Semantics match?
No
Reject Target Representation Artefact
Yes
Source and Target Value Types match?
No
Reject Target Representation Artefact
Yes
Score Value Categories
Score Variance Capabilities
Choose Target Representation Artefact with highest score
Output: A Mapping
Get next unmapped Target Representation Artefact
None
Map unmapped Target Representation Artefact to a Root Source Data Entity
Output: A Valid Mapping

Figure 5.2: The AutoMap algorithm

5.4.3.3 VizThis AutoMap Algorithm

The AutoMap algorithm takes a brute-force approach to finding a mapping between the source and target formats using the type semantics it has deduced or which have been provided by the user (see stage 1 in section 5.4.1). An acceptable mapping is one where the Entity to Artefact mappings have compatible characteristics. Where there are multiple target artefacts for which a source entity can be mapped to, the one with the greatest number of commonalities is chosen. These commonalities are derived from the Value Type, Value Category, Structure Semantic and Uniqueness / Variance Capability of each Entity and Artefact.

The AutoMap algorithm is shown in the flowchart in figure 5.2. The inputs, to the algorithm are: the *Facts* about the Source and the Target (the Source Data Analysis Table and Target Representation Artefacts Definition); and the Constraints (the Mapping Function Prioritisation Table).

For each source entity to target artefact possibility, the algorithm must ensure that the Source and Target Structure Semantics match. If not, then the possibility is rejected and an alternative possibility is tested. This is shown at the first decision point. Similarly, the Source and Target Value Types must match also and this is shown at the second decision point.

Next, the algorithm gives each potential Entity to Artefact mapping a score based on its Value Categories. The score is a value from 1 to 3 *'s depending on the function prioritisations given in table 5.4. For most artefacts, we only produce one score. For example, if the target artefact x is mapped to a Quantitative source entity, then a single score of *** is given. However, for some artefacts (namely, *fill* and *title*), two scores are given. This is because these artefacts can accept multiple value categories. For example, if *title* is given a Nominal value from an artefact which is Quantitative then it gets *** (because its a Nominal value) and * (because its a Quantitative artefact). This is explained in more detail in section 5.4.3.2.

A similar scoring process occurs for the Variance Capabilities. However, it is simpler in that only one score is given to each Entity to Artefact mapping. The variance values are either High (H), Medium (M), or Low (L). If the variance values are the same, then *** is given. If there is a difference of one (e.g., H and M), then ** is given. If there is a difference of two (e.g., H and L), then * is given.

AutoMap then performs a second and final pass through the mappings to ensure that all mandatory target representation artefacts are mapped. If any are not mapped, they are bound to a suitable source data entity and given a default value. This process is demonstrated in figure 5.9 with the `width` and `height` target representation artefacts which are unmapped in AutoMap's first pass, but are mapped to the `top-forty-chart` source data entity in the second pass because they are mandatory.

In this way, AutoMap uses a combination of a brute-force and scoring approaches. It finds acceptable mappings using a depth-first search (brute force) and then when multiple acceptable mappings exist, it evaluates each possible mapping using a scoring approach. The advantages of this approach are that the algorithm can find an acceptable mapping in the majority of cases. However, this relies on the source data schema and target representation schema being well designed as discussed in section 4.4.1. The limitation of the AutoMap algorithm are as follows:

Singularity The algorithm only gives one output. Therefore, multiple possible visualization are not shown.

Comparability The algorithm can not give meaningful scores to visualizations. In this way, it is not possible to compare and evaluate alternative visualizations.

Simplicity The algorithm can only handle well designed schemas. If a source and/or a target schema have multiple hierarchy levels, the algorithm will not find any acceptable mappings and in this circumstance, no visualization is shown to the user.

These limitations are discussed fully in the summary section of this chapter (see section 5.8).

The best mapping for the Fans dataset visualization is shown in table 5.6. The AutoMap algorithm gives this mapping a score of 39 based on the number of *'s awarded.

Source	Target	Val Type	Struc Sem	Value Category	Variance Capability
fans	svg		Root - Root		
person	rect		Obj - Obj		
games	x	num - num		quant - quant (***)	M - M (***)
tallness	y	num - num		quant - quant (***)	H - M (**)
nationality	fill	text - text		nom - nom (** + ***)	L - L (***)
scarves	width	num - num		quant - quant (***)	L - L (***)
age	height	num - num		quant - quant (***)	M - L (**)
name	title	text - text		nom - nom (*** + ***)	H - H (***)

Table 5.6: The results of the AutoMap analysis. This mapping is valid and has a score of 39 (indicated by the number of *'s). "num" = numeric; "quant" = quantitative; "nom" = nominal; "H" = High variance; "M" = Medium variance; "L" = Low variance.

5.4.4 Mapping Entity Values to Artefact Values (Stage 4)

In most cases the values of the source data cannot be used directly in the Target Representation. Instead a value mapping or translation process must occur. For example, the `nationality` entity is mapped to the `fill` colour representation artefact, therefore we must create a mapping between the nationality names and appropriate colours. In this case, `Welsh` is mapped to `red`, `Irish` is mapped to `green`, `Scottish` is mapped to `dark-blue` and `English` is mapped to `white`.

Alternatively, we may wish to use a function e.g., for mapping numeric values:

$[0, 1] \rightarrow \{0, ...10\}$

The specifics of how these techniques are used is described in Section 5.5.6.3.

5.4.5 Generate Target Representation (Stage 5)

The inputs to this stage are the Entity to Artefact mapping table (i.e., table 5.5), the Value mapping table (see "Lookup value table" in section 5.5.6.3) and the Source Data. We generate the Target Representation by creating Representation Artefacts (in the target representation format) with the values supplied by the Source Data and the Value mapping table. For our example, the output of this stage is the final Target Representation in SVG which is shown in figure 5.3a. Additionally, the X3D visualization for the same dataset is shown in figure 5.3b.

The goal of the VizThis tool is to allow users to easily produce cognitively useful visualizations using a Tree-centric mapping approach. The resulting visualizations, whether produced entirely automatically with the AutoMap feature or produced by manual tweaking of mappings, must give a greater level of cognitive insight than viewing the source data in its original form. Its original form may be a table of data or a spreadsheet, for example. When viewing the target visualizations in figures 5.3a and 5.3b, certain insights and patterns are clear. For example, all Welsh sports fans (represented by red rectangles) are shown on the left hand side of the visualization. As *age* is visualized using the *x* coordinate of each rectangle, this means that all Welsh fans are below a certain age. Additionally, we can see that the tallest fan is Irish (green rectangle) and the oldest fan is English (grey rectangle). This level of insight is greater than the source data in its original format provides. However, in order to test this hypothesis more rigorously, we conduct a user evaluation in section 5.6.

5.5 VizThis: User Interface

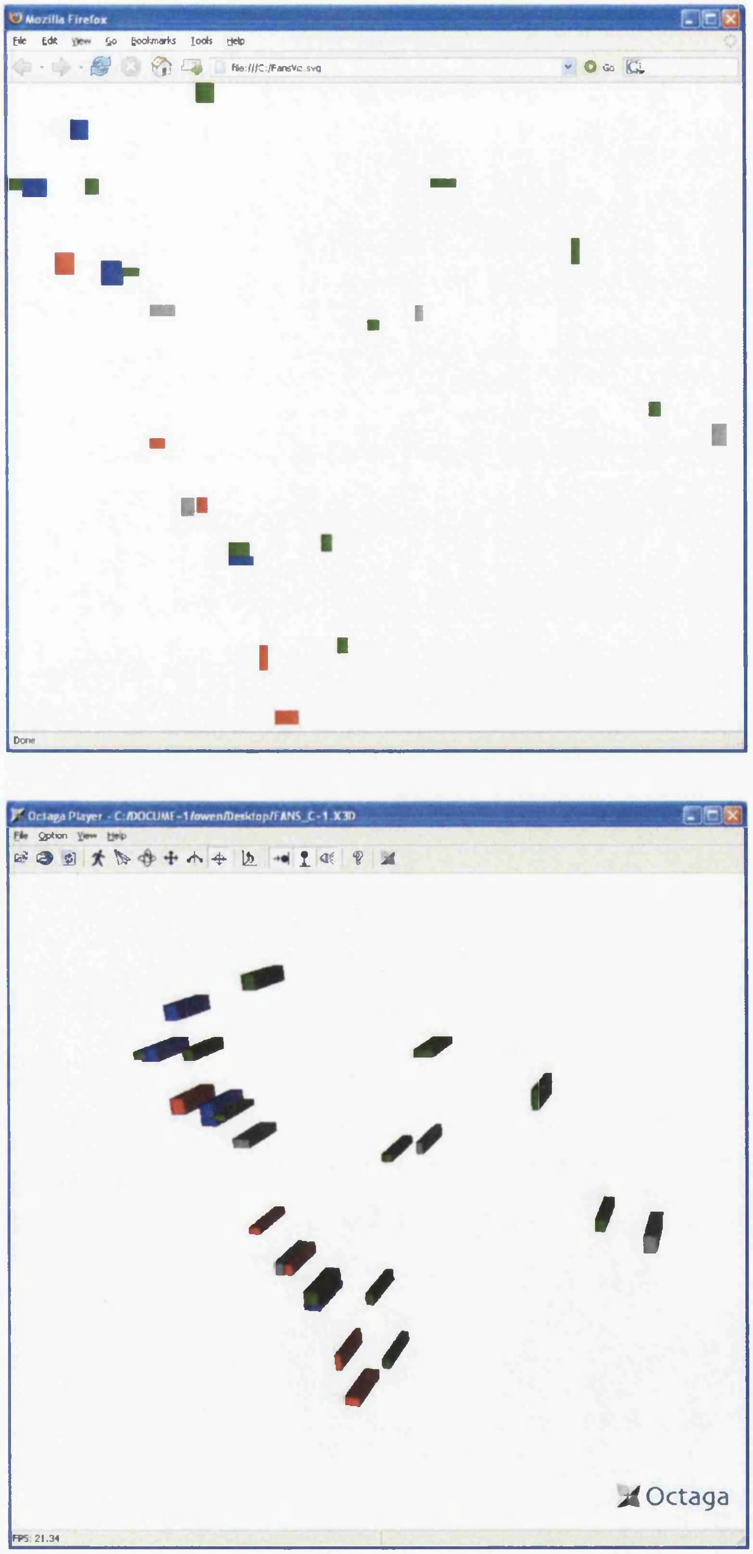
The VizThis tool consists of a User Interface and associated mapping/translation engine which is based on the Visualization as Mapping paradigm. We attempt to provide a tool which exploits the advantages of mapping, while keeping in mind the process of visualization. In this section we describe each of the features of the VizThis tool. We make extensive reference to the screenshot shown in figure 5.4. The screenshot shows the visualization of the sports fans data, expressed in XML.

5.5.1 Source File

The source file is shown in the top left hand corner of figure 5.4. The data in the source file remains editable with VizThis, so the user may perform value changes or simple data cleansing at any time. Data Analysis is performed on the source data and from this the Source Entities are populated.

5.5.2 Source Entities

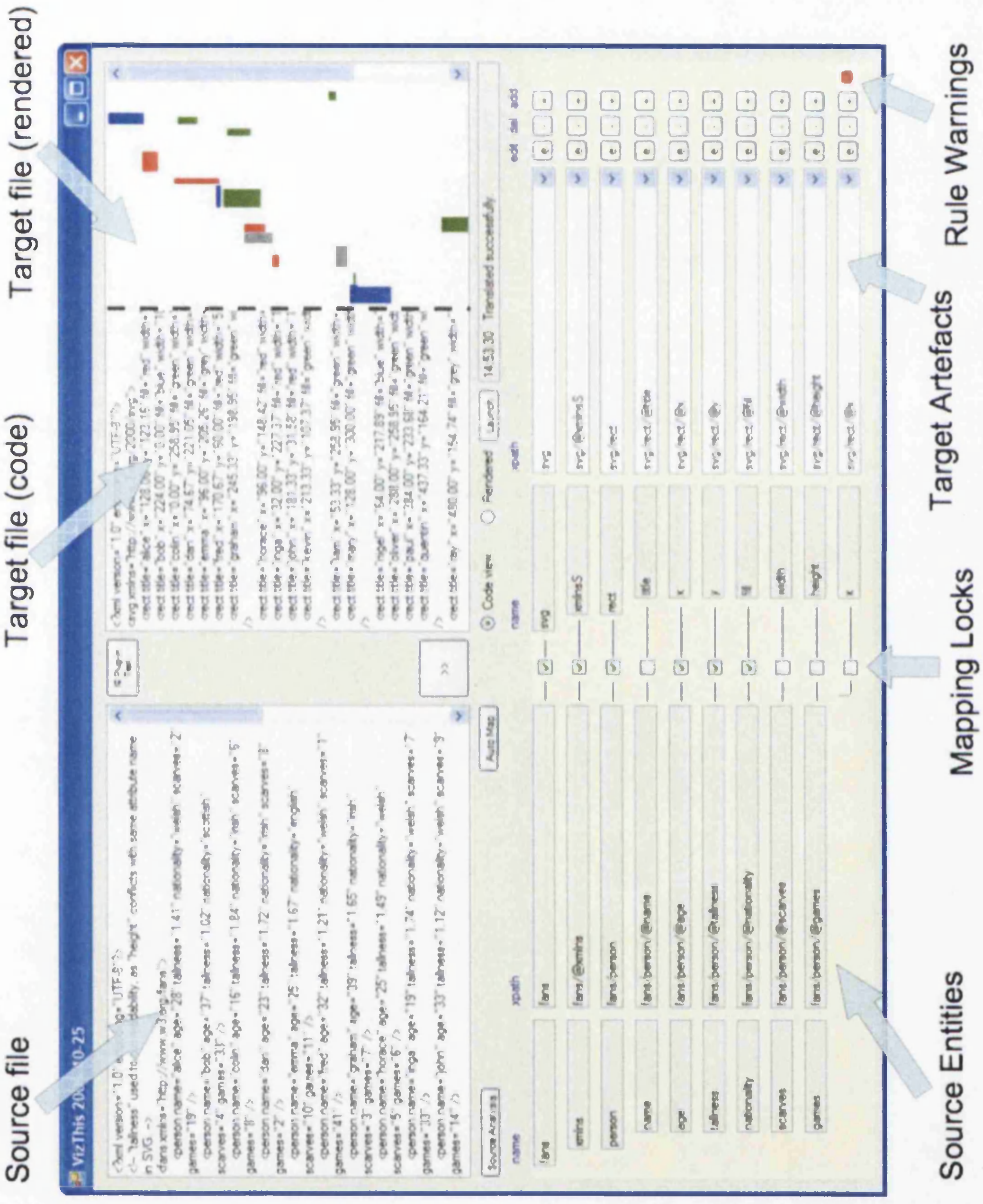
A source data entity is an element or attribute. Each source data entity has a name, a value type and a value category. An example of a source data entity in the sport fans information is `fans/person/@name`. A full description of the nature of Source Entities is given in section 5.4.1. VizThis shows each entity's name and XPath. The entity names are indented according to the entity's hierarchy level in the source file. This aids the user with schema comprehension. Note that VizThis does not require the XML source to have an associated schema file (DTD, XSD, Relax NG etc) in order to populate the Data Entities section of the VizThis user interface. This is because VizThis has the functionality to derive schema information as described in the following section.


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///C:/FansViz.svg
Go
Done
Octaga Player - C:/DOCUME~1/owen/Desktop/FANS_C-1.X3D
File Option View Help
Octaga
FPS: 21.34

(a) SVG

(b) X3D

Figure 5.3: Sports Fans dataset visualized in SVG and X3D in VizThis. For both visualizations: age is mapped to shape x position; tallness is mapped to shape y position; nationality is mapped to shape fill colour (red is Welsh, green is Irish, blue is Scottish, grey is English); number of scarves owned is mapped to width of shape; number of games attended is mapped to height of shape; the name of the sports fan is mapped to the title of the shape (seen when the user moves the mouse pointer over the rectangle - SVG only).



```
<?xml version="1.0" e      ng="UTF-8"?>
<!-- "tallness" used to        dability, as "height" conflicts with same attribute name
in SVG -->
<fans xmlns="http://www.w3.org/fans">
   <person name="alice" age="28" tallness="1.41" nationality="welsh" scarves="2"
games="19" />
   <person name="bob" age="37" tallness="1.02" nationality="scottish"
scarves="4" games="33" />
   <person name="colin" age="16" tallness="1.84" nationality="irish" scarves="6"
games="8" />
   <person name="dan" age="23" tallness="1.72" nationality="irish" scarves="8"
games="2" />
   <person name="emma" age="25" tallness="1.67" nationality="english"
scarves="10" games="11" />
   <person name="fred" age="32" tallness="1.21" nationality="welsh" scarves="1"
games="41" />
   <person name="graham" age="39" tallness="1.65" nationality="irish"
scarves="3" games="7" />
   <person name="horace" age="25" tallness="1.49" nationality="welsh"
scarves="5" games="6" />
   <person name="inga" age="19" tallness="1.74" nationality="welsh" scarves="7"
games="33" />
   <person name="john" age="33" tallness="1.12" nationality="welsh" scarves="9"
games="14" />
```

```
<?xml version="1.0" en        ="UTF-8"?>
<svg xmlns="http://w          g/2000/svg">
  <rect title="alice" x="128.00" y="123.15" fill="red" width=
  <rect title="bob" x="224.00" y="0.00" fill="blue" width="1
  <rect title="colin" x="0.00" y="258.95" fill="green" width=
  <rect title="dan" x="74.67" y="221.05" fill="green" width=
  <rect title="emma" x="96.00" y="205.26" fill="grey" width
  <rect title="fred" x="170.67" y="90.00" fill="red" width="5
  <rect title="graham" x="245.33" y="198.95" fill="green" w
/>
  <rect title="horace" x="96.00" y="148.42" fill="red" width=
  <rect title="inga" x="32.00" y="227.37" fill="red" width="1
  <rect title="john" x="181.33" y="31.58" fill="red" width="1
  <rect title="kevin" x="213.33" y="107.37" fill="green" wid
/>
  <rect title="lam" x="53.33" y="252.95" fill="green" width=
  <rect title="mary" x="128.00" y="330.00" fill="green" wid
/>
  <rect title="nigel" x="64.00" y="217.89" fill="blue" width="
  <rect title="oliver" x="288.00" y="258.95" fill="green" widt
  <rect title="paul" x="384.00" y="233.68" fill="green" width
  <rect title="quentin" x="437.33" y="164.21" fill="green" w
/>
  <rect title="ray" x="490.00" y="154.74" fill="grey" width="
```

◉ Code view ○ Rendered 14:53:30 Translated successfully

name	xpath		name	xpath
fans	fans	☑	svg	svg
xmlns	fans/@xmlns	☑	xmlnsS	svg/@xmlnsS
person	fans/person	☑	rect	svg/rect
name	fans/person/@name	□	title	svg/rect/@title
age	fans/person/@age	☑	x	svg/rect/@x
tallness	fans/person/@tallness	☑	y	svg/rect/@y
nationality	fans/person/@nationality	☑	fill	svg/rect/@fill
scarves	fans/person/@scarves	□	width	svg/rect/@width
games	fans/person/@games	□	height	svg/rect/@height
		□	x	svg/rect/@x

Figure 5.4: VizThis : Visualizing the Sports fans data set in SVG

5.5.3 Data Analysis

The Data Analysis routines are triggered when a change is made to either of the following:

1. The source data itself.
2. The regular expression or search and replace fields in the data cleansing section of the "Edit Value Mappings" dialogue box.

In this way, VizThis always has a judgement on the type semantics of the source data entities, allowing the AutoMap feature to be invoked at any point.

5.5.4 Target Artefacts

Target Representation Artefacts are objects and attributes in the target format which are combined to form a visualization. For example, in SVG, `svg/rect` is a representation object and `svg/rect/@width` is a representation attribute. The target artefact to which each source entity is mapped is shown on its right hand side. A full description of the nature of Target Artefacts is given in section 5.4.3. VizThis shows each Artefact's name and XPath. Like the Source Entities, each Target Artefact's name is indented according to the artefact's hierarchy level in the target schema. Each Target Entitiy's XPath is shown as a Drop-down box control. When selected, this drop-down shows a list of all Target Entities in the target file's schema. In this way, the user is able to change the Target Entity on that mapping line, therefore changing the mapping target of the source entity.

5.5.5 Delete and Add Target Artefacts

The "-" and "+" buttons on each mapping line allow the user to delete the current mapping (if present) or create a new one. VizThis allows a source entity to be mapped to multiple target artefacts. This is depicted as a mapping fork in the VizThis user interface.

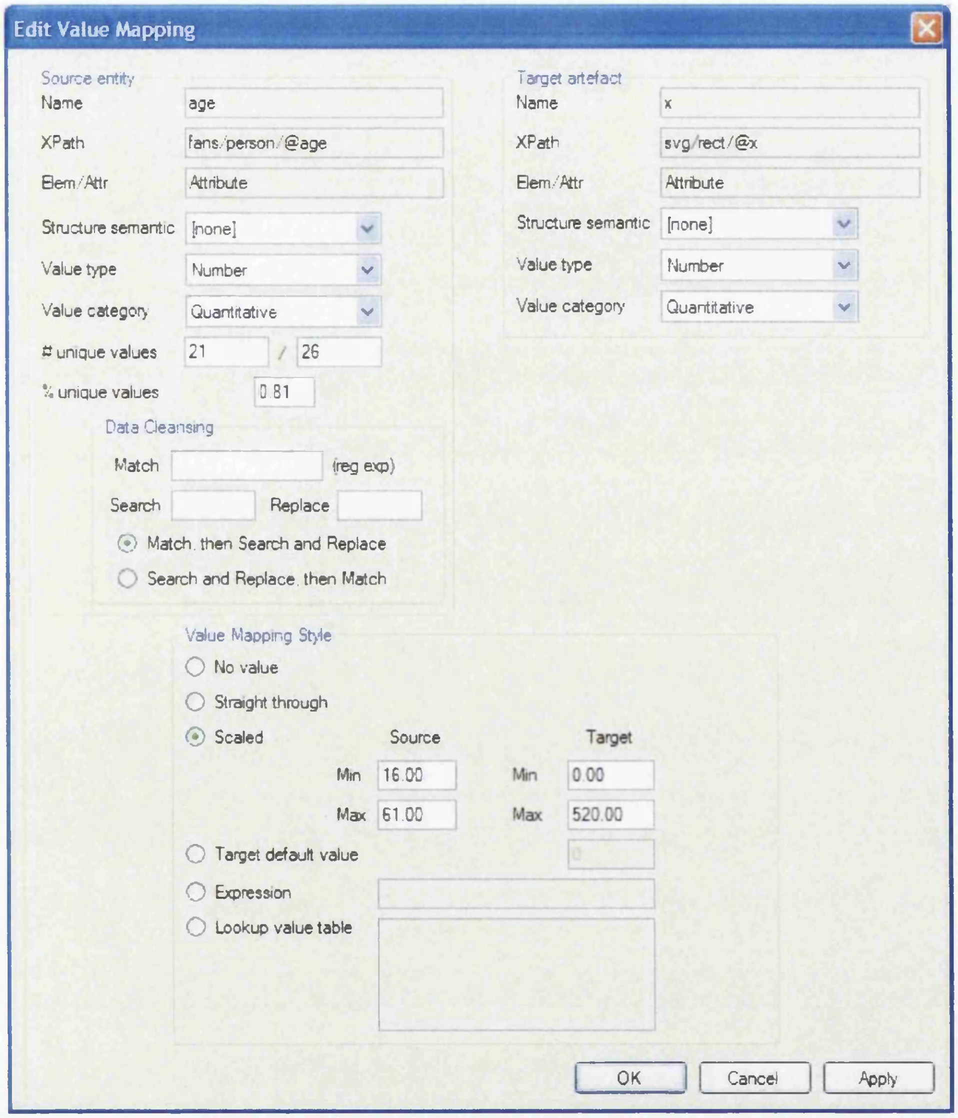
5.5.6 The Edit Value Mapping Dialogue Box

The parameters for how a Source Entity is mapped to a Target Artefact are displayed and edited in the "Edit Value Mapping" dialogue box (see figure 5.5). This is displayed by pressing the "e" button next to each mapping line.

The "Edit Value Mapping" dialogue box is split into 3 areas: Source entity; Target artefact; and Value Mapping Style.

5.5.6.1 Source entity settings

The first area, "Source entity" shows the properties of the source entity as described in section 5.4.1. This includes structure semantic, value type, value category, and number of unique values. These read-only values cannot be directly edited by the user. However, there is a sub-area which deals with Data Cleansing. This area is editable by the user. The user can specify a regular expression to be applied to the source entity. This is a standard regular expression and can be used for example to strip out numbers from a text string - the regular expression `[0-9]+` would do this. Also, the user can specify a search and replace expression for changing specific text strings. The user is able to select in which order the regular expression and search and replace processes are executed. When any data cleansing parameters are altered, the user can press the Apply button. This will perform a new Data Analysis process. As a result of this, the source entity parameters may be changed. For



Edit Value Mapping

Source entity

Name: age

XPath: fans/person/@age

Elem/Attr: Attribute

Structure semantic: [none]

Value type: Number

Value category: Quantitative

unique values: 21 / 26

% unique values: 0.81

Data Cleansing

Match: (reg exp)

Search: Replace:

● Match, then Search and Replace

○ Search and Replace, then Match

Target artefact

Name: x

XPath: svg/rect/@x

Elem/Attr: Attribute

Structure semantic: [none]

Value type: Number

Value category: Quantitative

Value Mapping Style

○ No value

○ Straight through

● Scaled

	Source		Target
Min	16.00	Min	0.00
Max	61.00	Max	520.00

○ Target default value: 0

○ Expression

○ Lookup value table

Figure 5.5: The VizThis Edit Value Mapping Dialogue Box

example, if the parameters entered for data cleansing result in a source entity being reclassified as having a Value type of Number, then this will be reflected in the Source entity settings.

5.5.6.2 Target artefact settings

The second area, "Target artefact" shows the properties of the target artefact as described in section 5.4.3. This includes structure semantic, value type and value category. These are read-only values and cannot be edited by the user. They are set when the available target formats are entered into the system (i.e., SVG and X3D). They are displayed as informational values only.

5.5.6.3 Value mapping mechanism

When a Source Entity is mapped to a Target Artefact, the value which the Target Artefact is given is usually based on the value of the Source Entity. There are 6 ways in which the value can be transferred:

No value The Target Artefact is always given a value of an empty string, no matter what the value of the Source Entity is.

Straight through (unchanged) The Target Artefact is given the exact same value as the Source Entity.

Scaled This is an arithmetic operation allowing a range of numeric source values to be scaled linearly to a range of target values. For example, if we have an `age` source entity whose values range from 0 to 120 and a `x` target artefact whose values range from 0 to 800 (representing the width of the screen in pixels), then we can scale any age value to an x screen coordinate linearly in order to represent any age in the range. The source entity range is calculated during the Data Analysis stage (see section 5.4.1). Any Data Cleansing parameters are applied before calculating this range. The target artefact range is set when a new target format is added to the VizThis system.

Target default value Each target artefact can have a default value. This is intended to be a sensible default for the visualization of that artefact. For example, the default for a the `svg/rect/@width` target artefact is 20.

Expression The user is able to enter an expression in C#. This is useful for applying arithmetic expressions or function transformations to entity values.

Lookup value table When the source entity is made up of discrete values, it is useful to have a lookup table of equivalent target artefacts. For example, if the source entity is a country, the lookup table can contain value mappings to each country's national colour.

5.5.7 Semantic Assistance - AutoMap

When a user is confident that the source entities are correctly categorised, either through Data Cleansing and then Data Analysis, or by overriding the system's categorisations, they can press the AutoMap button. The system will calculate the best mappings based on the algorithm described in section 5.4.3.3. Of course, the user can press the AutoMap button at any stage (e.g., before any data cleansing has taken place), and a valid mapping will be attempted. This mapping may not produce the most cognitively accurate visualization, but it will enable the user to see something with relatively little effort.

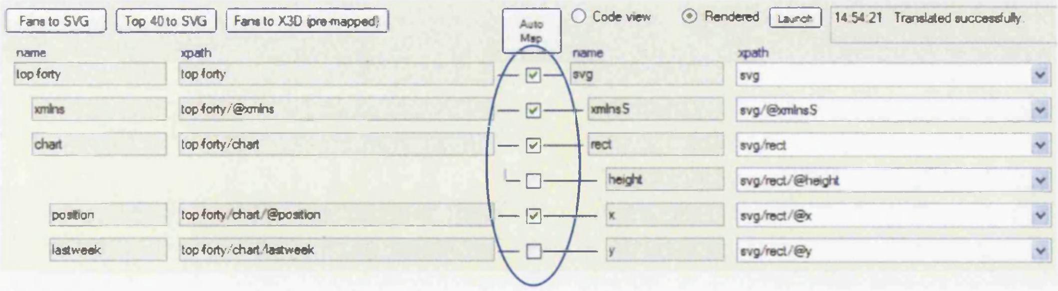
5.5.8 Execute Translation

When the user is happy with the mappings between the source entities and target artefacts, they can press the "Execute Translation" button (with the >> label) in order to have the target code generated. If the "Code view" radio button is selected, the user will see the target code, or if the "Rendered" radio button is selected the user will see the visualization rendered (currently in SVG or X3D). If there are any Absolute Constraint Warnings (see section 5.5.10), then the target code will not be generated and an error box will be displayed explaining the reason. Following a valid translation, the status box is updated to indicate a successful operation.

The VizThis application gives an embedded preview of the rendered visualization within the main application window. However, the user is able to press the "Launch" button at any time in order to see the visualization rendered in a full-screen external application (i.e., Mozilla Firefox for SVG or Octaga Player for X3D). The embedded preview is limited to what Microsoft Internet Explorer 6.0 can display (including any available plug-ins). However, the "Launch" button may be used to invoke any third party application.

5.5.9 Mapping Locks

While the user is tweaking the mapping choices, the user can tell the system to automatically re-generate the mappings using the AutoMap button. If the user has decided that there are some mappings with which they are happy and do not wish to alter, they can lock them by selecting a Lock checkbox. In this case, the constraint-system would consider such locks as definitive mappings and take them as anchors [NM01] for the rest of the mapping process. This is illustrated in figure 5.6 where we wish to keep all mappings except the two which are mapped to `svg/rect/@height` and `svg/rect/@y`. Note that the "AutoMap" button is enabled at this point, allowing the user to re-generate the un-locked mappings.



14:54:21 Translated successfully.

name	xpath		name	xpath
top forty	top forty	☑	svg	svg
xmlns	top forty/@xmlns	☑	xmlnsS	svg/@xmlnsS
chart	top forty/chart	☑	rect	svg/rect
		□	height	svg/rect/@height
position	top forty/chart/@position	☑	x	svg/rect/@x
lastweek	top forty/chart/lastweek	□	y	svg/rect/@y

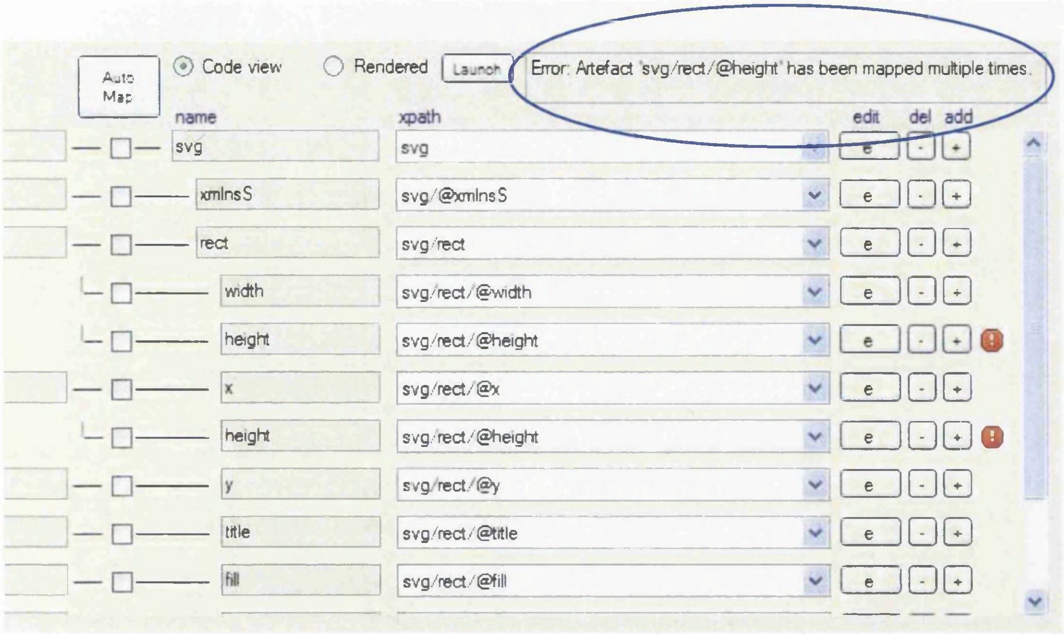
Figure 5.6: Mapping locks on four mapping lines

5.5.10 Constraint Warnings

When VizThis detects a problem with any mappings (constraint warnings), a red or yellow icon is displayed next to the offending mapping line together with a more detailed explanation in the Status Field.

There are 2 levels of constraints which are applicable to the system:

Absolute (red icon) Hard constraints which cannot be altered. An example is that there cannot be duplicate target entities. This is shown in figure 5.7, where there are two artefact attributes which are set to the same attribute (`svg/rect/@height`). This



● Code view ○ Rendered

Error: Artefact 'svg/rect/@height' has been mapped multiple times.

name	xpath	edit	del	add
svg	svg	e	-	+
xmlnsS	svg/@xmlnsS	e	-	+
rect	svg/rect	e	-	+
width	svg/rect/@width	e	-	+
height	svg/rect/@height	e	-	+
x	svg/rect/@x	e	-	+
height	svg/rect/@height	e	-	+
y	svg/rect/@y	e	-	+
title	svg/rect/@title	e	-	+
fill	svg/rect/@fill	e	-	+

Figure 5.7: VizThis notifying the user that there is an Absolute Constraint Warning.

is indicated by the two red warning icons. The Execute Translation stage cannot begin while any mapping lines are in this state.

Preferential (yellow icon) Soft constraints (or guidelines) which are set by a developer when a new representation format is added to the system. This is shown in figure 5.8. An example is the general principle that the SVG shape object `svg/rect` should have a `svg/rect/@height` attribute. This is indicated by a yellow warning icon. The Execute Translation stage can happen while there are mapping lines in this state. However, the quality of the resulting visualization may be low due to the warnings given not being considered.

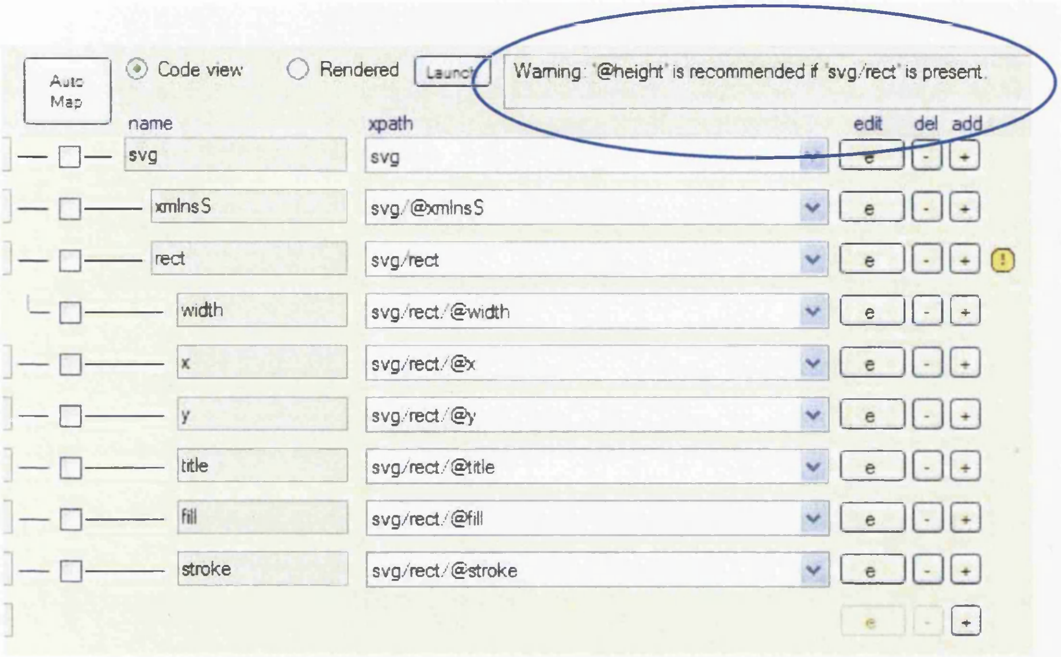
Constraints have 2 different scopes of influence:

Inter Artefact Concerned with the relationship between target artefacts. For example, no two source entities may be mapped to the same target representation artefact (see figure 5.7).

Stand Alone These are constraints which apply to individual artefacts. For example, the `svg/rect/@height` attribute should always have a value if there is a `svg/rect` object present (see figure 5.8).

5.5.11 Editing Mappings

Using the AutoMap algorithm, the VizThis tool can make assumptions about the source entities which should be mapped to the target artefacts. However, these mappings may not be the best available and the user is able to change them at any time. This can be done



Warning: '@height' is recommended if 'svg/rect' is present.

name	xpath	edit	del	add
svg	svg	e	-	+
xmlnsS	svg/@xmlnsS	e	-	+
rect	svg/rect	e	-	+
width	svg/rect/@width	e	-	+
x	svg/rect/@x	e	-	+
y	svg/rect/@y	e	-	+
title	svg/rect/@title	e	-	+
fill	svg/rect/@fill	e	-	+
stroke	svg/rect/@stroke	e	-	+
		e	-	+

Figure 5.8: VizThis notifying the user that there is a Preferential Constraint Warning.

by using the add (+) and delete (-) mappings buttons on the right of the main window (see section 5.5.5). Additionally, an already mapped target artefact can be changed by selecting a new artefact in the drop-down menu. If an invalid mapping is chosen, the user is immediately warned by the system.

5.6 VizThis: Worked Example - BBC Top 40 Music Chart to SVG

In this example, we take a real-life XML feed [Web08b] of the BBC Top 40 pop music chart [BBC08] and use the VizThis tool to produce an SVG representation. The main difference between this dataset and the Sports Fans dataset used in the example in section 5.4 is that it demonstrates the Data Cleansing stage and how it interacts with the Source Data Analysis stage. We also show the user interaction stages which a user may take in order to go from a completely automatically created visualization to a more evolved and cognitively useful visualization.

A snippet of the source showing the first chart song is below:

```
<top-forty>
    <chart position="1">
        <lastweek>1</lastweek>
        <weeks>(5)</weeks>
        <image>http://www.bbc.co.uk/radio1/media/images/artists/gnarlsbarkley/
        060323_cd_crazy_70.jpg</image>
        <artist>Gnarls Barkley</artist>
        <album>Crazy</album>
        <uri>http://www.gnarlsbarkley.com/</uri>
    </chart>
    . . .
<top-forty>
```

The source data consists of 40 chart songs (only one is shown above) with its current position, last week's position, number of weeks in the top 40, a URL to an image of the artist, the name of the artist, the name of the song, and a URL for more information about the artist. This particular source example provides a good set of data in which to demonstrate VizThis' features. Firstly, it uses a mixture of child elements and attributes to convey values, thus showing the idiosyncrasies of real-life XML data schema design. Secondly, some fields require data cleansing. Thirdly, it uses a variety of different data types.

5.6.1 Source Data Analysis

During this stage, each entity in the source is analysed. This results in the data shown in table 5.1.

XPath	**Value type**	**Value category / Structure semantic**	**Uniqueness**	**Min**	**Max**
top-forty	-	Container (root)	-	-	-
top-forty/chart	-	Container (object)	-	-	-
top-forty/chart/@position	Integer	Quantitative	40 values (1.0)	1	40
top-forty/chart/lastweek	**Integer(0.7), String(0.3)**	Quantitative	**29 val's (0.73)**	1	30
top-forty/chart/weeks	String	Nominal	**12 val's (0.3)**	-	-
top-forty/chart/image	URL	Nominal	**8 val's (0.2)**	-	-
top-forty/chart/artist	String	Nominal	40 values (1.0)	-	-
top-forty/chart/album	String	Nominal	40 values (1.0)	-	-
top-forty/chart/uri	URL	Nominal	40 values (1.0)	-	-

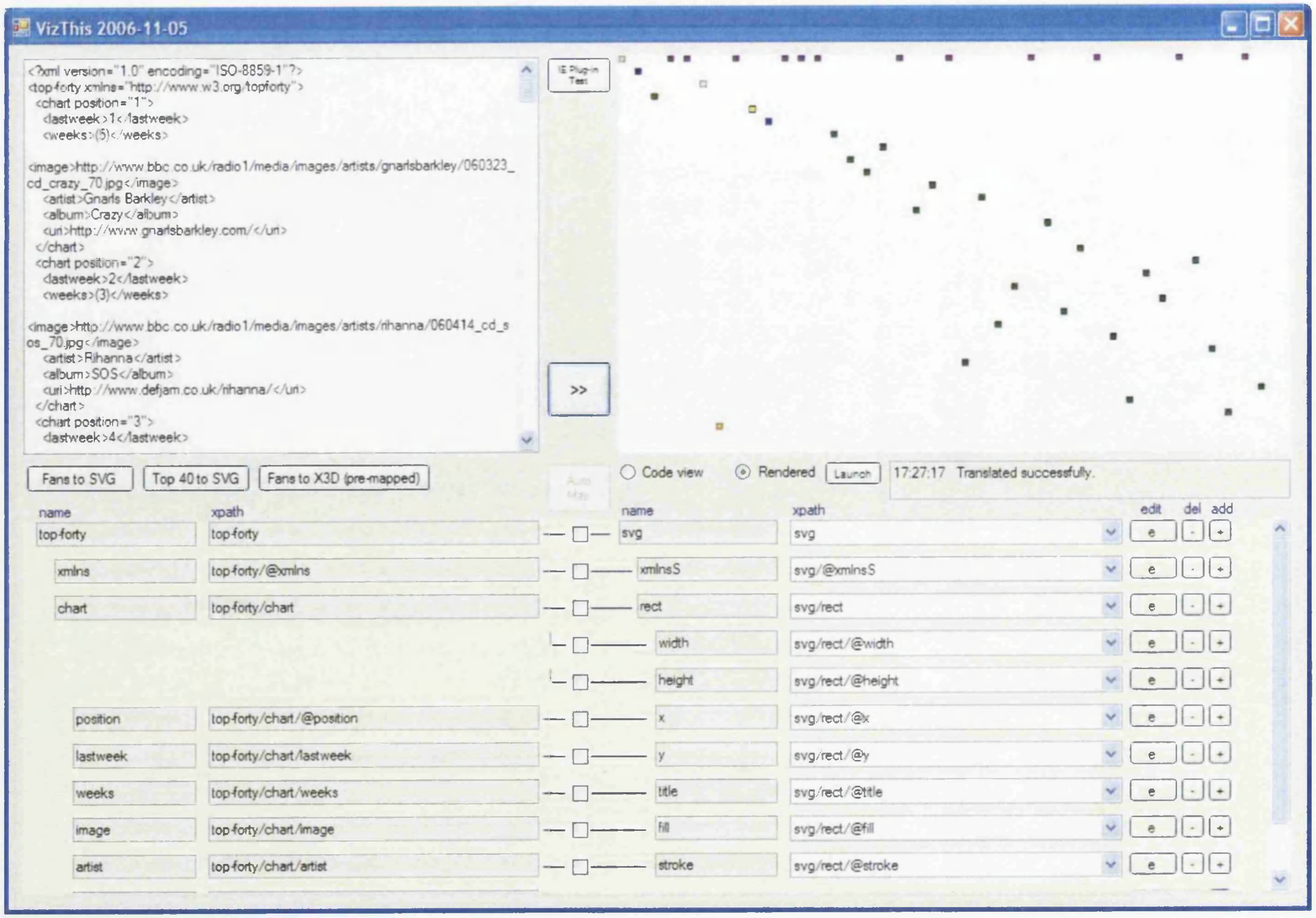
Table 5.7: Source data Entity analysis for the BBC Top 40 music chart dataset (before data cleansing)

It can be seen that the entities highlighted in bold have proportional values of less than 1. This often indicates ambiguities in the data, or areas where data cleansing may be needed. The entities `lastweek`, `weeks` and `image` have been given values less than 1.0 for their uniqueness. Also, `lastweek` has multiple possible Value Types. There are 3 possible actions which the user can take:

1. **Continue with the original source data**. No user interaction is necessary and the system will attempt to provide a "good enough" visualization with the information it already has.

2. **Alter the source data manually**. A good solution if there is a small volume of source data as it is a relatively quick process.

3. **Apply data cleansing functions**. Create data cleansing functions in order to fix ambiguous data. Functions can be recorded and applied again when the same data format is detected on future occasions.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<top-forty xmlns="http://www.w3.org/topforty">
  <chart position="1">
    <lastweek>1</lastweek>
    <weeks>(5)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/gnarlsbarkley/060323_
cd_crazy_70.jpg</image>
    <artist>Gnarls Barkley</artist>
    <album>Crazy</album>
    <uri>http://www.gnarlsbarkley.com/</uri>
  </chart>
  <chart position="2">
    <lastweek>2</lastweek>
    <weeks>(3)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/rihanna/060414_cd_s
os_70.jpg</image>
    <artist>Rihanna</artist>
    <album>SOS</album>
    <uri>http://www.defjam.co.uk/rihanna/</uri>
  </chart>
  <chart position="3">
    <lastweek>4</lastweek>
```

17:27:17 Translated successfully.

name	xpath	name	xpath
top-forty	top-forty	svg	svg
xmlns	top-forty/@xmlns	xmlnsS	svg/@xmlnsS
chart	top-forty/chart	rect	svg/rect
		width	svg/rect/@width
		height	svg/rect/@height
position	top-forty/chart/@position	x	svg/rect/@x
lastweek	top-forty/chart/lastweek	y	svg/rect/@y
weeks	top-forty/chart/weeks	title	svg/rect/@title
image	top-forty/chart/image	fill	svg/rect/@fill
artist	top-forty/chart/artist	stroke	svg/rect/@stroke

Figure 5.9: BBC Top 40 data visualised using VizThis's AutoMap feature only. No data cleansing, or manual tweaking has been performed.

Firstly, we will assume that the user takes action number 1 - Continue with the original source data. In this case, the AutoMap feature can be used to produce an automatic mapping between the source entities and target artefacts. This is shown in figure 5.9. It can be seen that the visualization produced is certainly not useless and does contain some cognitive value. Each square represents a chart song. It can be observed that some squares sit along the top of the visualization (representing new songs in the chart) with the remainder forming an correlation along the diagonal line from top-left to bottom-right hand corner (representing how much each song has risen or fallen in the charts). A more thorough assessment of the cognitive value of this particular visualization is conducted during the User Evaluation in section 5.7.

We will next assume that the user desires a better visualization and so decides to perform some data cleansing on the data.

5.6.2 Data Cleansing

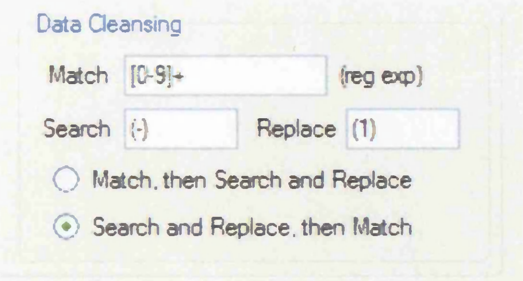
A code snippet representing one song ("Steady As She Goes" by "Raconteurs") illustrates the two data problems on lines 4 and 5:

```
 1:   <top-forty>
 2:       . . .
 3:       <chart position="4">
 4:           <lastweek>NEW</lastweek>
 5:           <weeks>(-)</weeks>
 6:           <image>/radio1/chart/media/chart_star.gif</image>
 7:           <artist>Raconteurs</artist>
 8:           <album>Steady As She Goes</album>
 9:           <uri>http://www.theraconteurs.com/</uri>
10:       </chart>
11:       . . .
12:   </top-forty>
```

The two problems with the original source data are:

1. Firstly, the `lastweek` entity is numeric most of the time (28 of 40 records), except when the song is a new entry (12 of 40 records), in which case it is set to the string "NEW". The Source Data Analyser gives a proportional value to the Value Type based on how many records meet each criteria. As can be seen, the Source Data Analyser gives more likelihood to the entity being of value type Integer. So in this case all values are assumed to be integers. Fortunately, the tool interprets any non-numeric value as a zero which is a good alternative value for `lastweek` to give songs when they are new entries.

2. Secondly, the `weeks` entity is deemed to be a String value by the system. This is because every number is surrounded by parenthesis. In order to gain the true semantics from this entity, we must remove the parenthesis. Additionally, when the chart song is a new entry, the value is set to a dash (–). Again, in order to gain most semantic value from this entity, we need to change each dash to a 1. This is still semantically correct, since it accurately represents the number of weeks the song has been in the charts. In order to deal with these two aspects of data cleansing for `weeks`, we use a regular expression, together with simple search and replace strings. This is illustrated in figure 5.10. The regular expression `[0-9]+` is used to extract the numeric values from an alpha-numeric string. The search string of (–) is to specified along with the replace string of (1). Finally, the radio buttons specify that the search and replace should be carried out before the regular expression matching.



Data Cleansing

Match [0-9]+ (reg exp)

Search (-) Replace (1)

○ Match, then Search and Replace

● Search and Replace, then Match

Figure 5.10: The Regular Expression and Search and Replace strings entered in the Edit Value Mapping dialogue box in VizThis. This relates to the `weeks` source entity.

After defining operations for data cleansing, we re-analyse the data to produce table 5 (only relevant entities are shown).

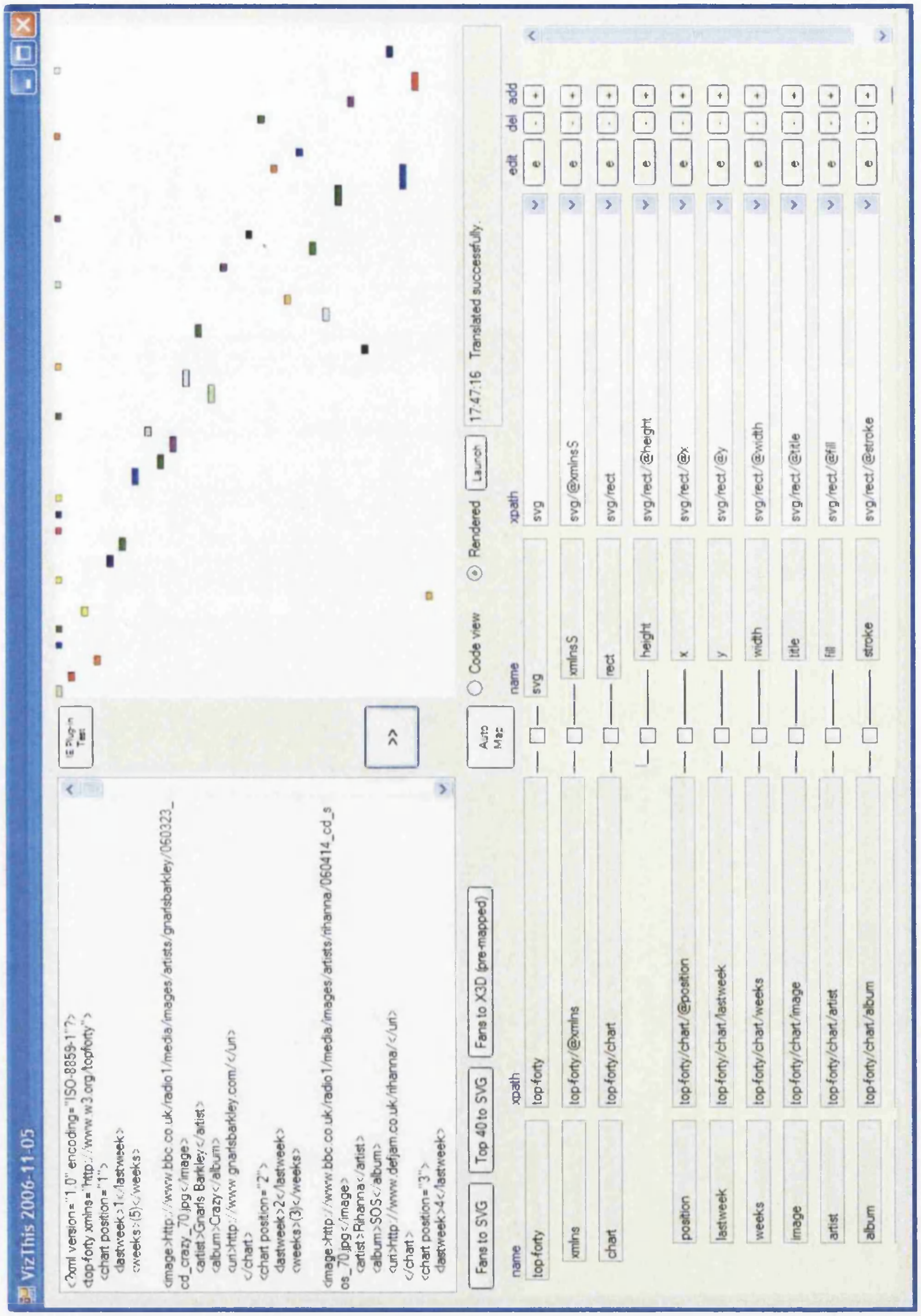
XPath	Value type	Value category / Structure semantic	Uniqueness	Min	Max
...	...	...	...	...	...
top-forty/chart/lastweek	**Integer**	**Quantitative**	**29 values (0.73)**	**1**	**30**
top-forty/chart/weeks	**Integer**	**Quantitative**	**12 values (0.3)**	**2**	**15**
top-forty/chart/image	URL	Nominal	8 values (0.2)	-	-
...	...	...	...	...	...

Table 5.8: Source data Entity analysis for the BBC Top 40 music chart dataset (after data cleansing and normalisation). Only relevant entities are shown. The `image` entity is unchanged.

It can be seen that the Entities, `lastweek` and `weeks` have been updated in the table (highlighted in bold). The ambiguities which existed before have been resolved. This will improve the quality of the semantics derived by the Entity to Artefact Matcher. However, the *image* entity still has unresolved ambiguities. We will see that this will not have a large impact on the final visualization since we know that the entity is a URL. For the purposes of this example we choose to ignore entities of type URL. Note that the *image* entity has few unique values because the source data only provides images for the top 10 songs and also those which are new entries. New entries all have the same image. This means the *image* entity has only 8 unique values.

If the user now uses the AutoMap feature to re-map between the source and target entities, VizThis is able to produce a better mapping and therefore a cognitively more useful visualization. This is illustrated in figure 5.11. Notice that the rectangles which represent songs have varying widths. The width represents the number of weeks the song has been in the charts. The VizThis visualization is therefore providing additional cognitive value over the original visualization which used the AutoMap feature with no user-specified data cleansing functions. Again, a more thorough assessment of the cognitive value of this visualization is conducted during the User Evaluation in section 5.7.

The user may now perform some manual tweaking of the mappings. Since the `weeks`



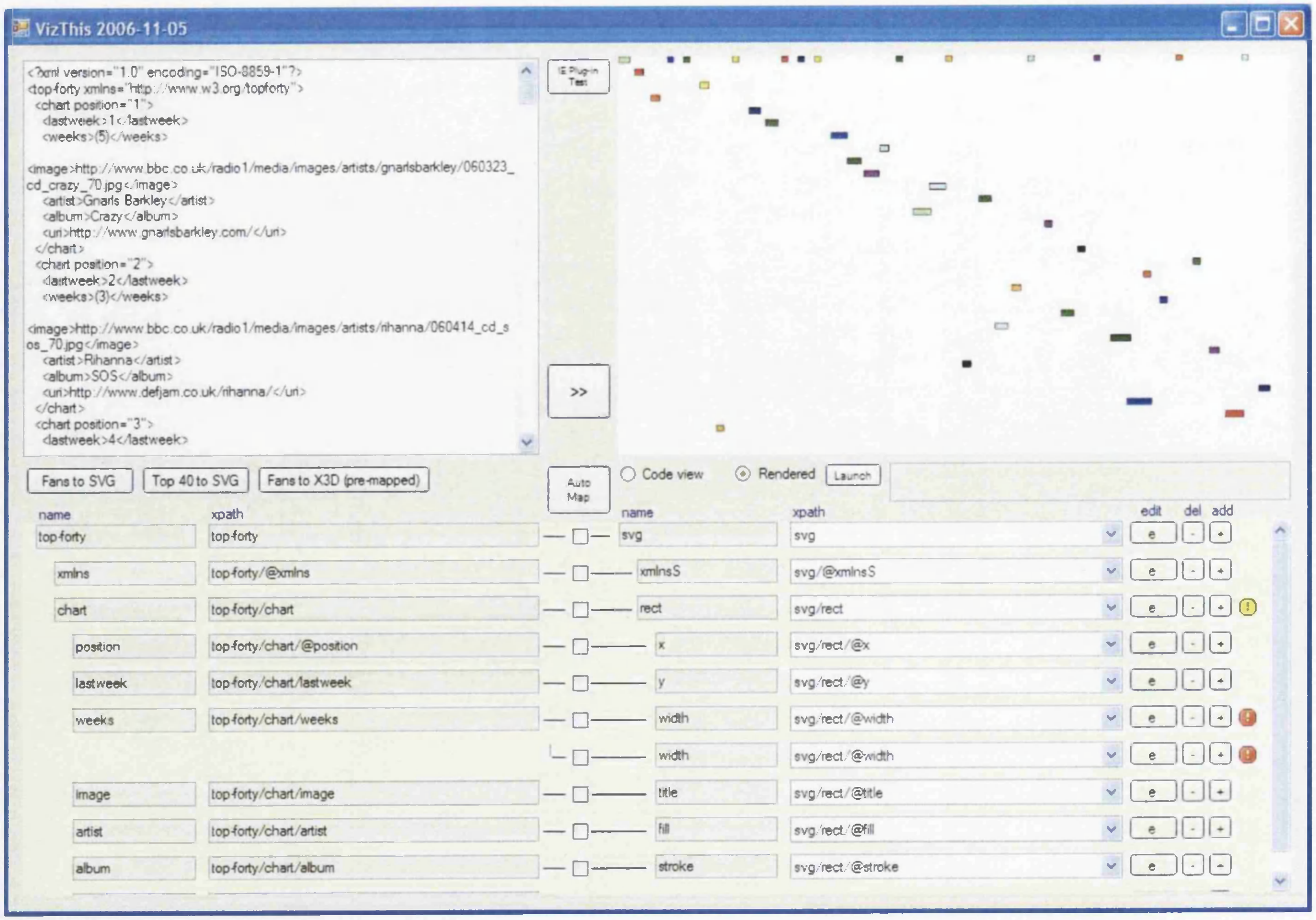
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<top-forty xmlns="http://www.w3.org/topforty">
  <chart position="1">
   <lastweek>1</lastweek>
   <weeks>(5)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/gnarlsbarkley/060323_
cd_crazy_70.jpg</image>
   <artist>Gnarls Barkley</artist>
   <album>Crazy</album>
   <uri>http://www.gnarlsbarkley.com/</uri>
  </chart>
  <chart position="2">
   <lastweek>2</lastweek>
   <weeks>(3)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/rihanna/060414_cd_s
os_70.jpg</image>
   <artist>Rihanna</artist>
   <album>SOS</album>
   <uri>http://www.defjam.co.uk/rihanna/</uri>
  </chart>
  <chart position="3">
   <lastweek>4</lastweek>
```

17:47:16 Translated successfully.

name	xpath	name	xpath
top-forty	top-forty	svg	svg
xmlns	top-forty/@xmlns	xmlnsS	svg/@xmlnsS
chart	top-forty/chart	rect	svg/rect
		height	svg/rect/@height
position	top-forty/chart/@position	x	svg/rect/@x
lastweek	top-forty/chart/lastweek	y	svg/rect/@y
weeks	top-forty/chart/weeks	width	svg/rect/@width
image	top-forty/chart/image	title	svg/rect/@title
artist	top-forty/chart/artist	fill	svg/rect/@fill
album	top-forty/chart/album	stroke	svg/rect/@stroke

Figure 5.11: The BBC Top 40 data visualised after data cleansing functions have been specified. Note that the `weeks` source entity has been mapped to the `svg/rect/@width` target artefact.

source entity has been mapped to the `svg/rect/@width` target artefact by the AutoMap feature, we are gaining additional insight into the data. However, it can be seen that the `svg/rect/@height` target artefact has been set at a single default value. The



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<top-forty xmlns="http://www.w3.org/topforty">
 <chart position="1">
  <lastweek>1</lastweek>
  <weeks>(5)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/gnarlsbarkley/060323_
cd_crazy_70.jpg</image>
  <artist>Gnarls Barkley</artist>
  <album>Crazy</album>
  <uri>http://www.gnarlsbarkley.com/</uri>
 </chart>
 <chart position="2">
  <lastweek>2</lastweek>
  <weeks>(3)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/rihanna/060414_cd_s
os_70.jpg</image>
  <artist>Rihanna</artist>
  <album>SOS</album>
  <uri>http://www.defjam.co.uk/rihanna/</uri>
 </chart>
 <chart position="3">
  <lastweek>4</lastweek>
```

name	xpath	name	xpath
top-forty	top-forty	svg	svg
xmlns	top-forty/@xmlns	xmlnsS	svg/@xmlnsS
chart	top-forty/chart	rect	svg/rect
position	top-forty/chart/@position	x	svg/rect/@x
lastweek	top-forty/chart/lastweek	y	svg/rect/@y
weeks	top-forty/chart/weeks	width	svg/rect/@width
		width	svg/rect/@width
image	top-forty/chart/image	title	svg/rect/@title
artist	top-forty/chart/artist	fill	svg/rect/@fill
album	top-forty/chart/album	stroke	svg/rect/@stroke

Figure 5.12: The BBC Top 40 data visualised. Notice the Absolute (red icon) and the Preferential (yellow icon) constraint warnings.

svg/rect/@height artefact is therefore not being used to convey any cognitively useful information. The user could manually map svg/rect/@height to the same source entity as svg/rect/@width, i.e., the weeks source entity. This would give more prominence to the weeks source entity because the shapes would therefore be squares of varying sizes. In order to do this the user adds `svg/rect/@height` as the second target artefact which the `weeks` source entity is mapped to. This is done using the "del" and "add" buttons next to each mapping line. Notice in figure 5.12 where we show an intermediate stage where there are constraint warnings. The Preferential (yellow icon) warning shows that target artefact `svg/rect` should have a `svg/rect/@height` child entity. The two Absolute (red icon) warnings show that the same target artefact (in this case `svg/rect/@width`) cannot be mapped in multiple lines. More information about the Absolute and Preferential warnings is provided in section 5.5.10.

After the user has manually mapped `svg/rect/@height` to the same source entity as `svg/rect/@width`, i.e., the `weeks` source entity, then the visualization can be regenerated as shown in figure 5.13.

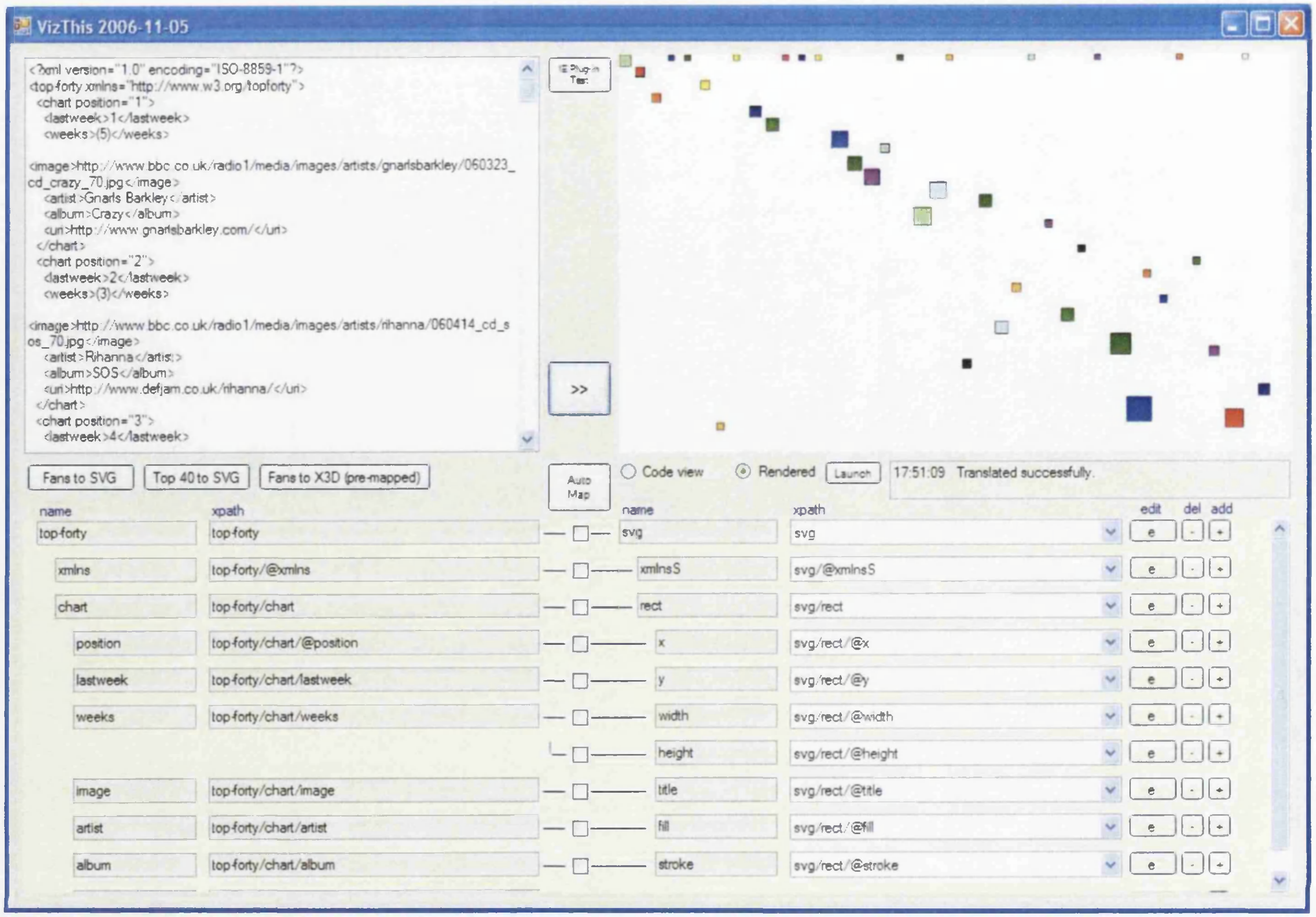
Finally, the user can make another alteration to the mappings to improve the visualization. In figure 5.13, it can be seen that there are different coloured shapes. Each colour represents a different artist. Since there are 40 different artists in the source data (see table 5.7), then this is providing little cognitive assistance to the user. If there were fewer different artists (e.g., 6 different artists in the whole data set of 40 songs), then the mapping of `artist` to `svg/rect/@fill` (shape colour) would be beneficial. However, with 40 different values of artist, the mapping is confusing since humans are not capable of differentiating 40 different colours. Therefore, the user can change the `svg/rect/@fill` (shape colour) target artefact to a set value. This can be the artefact's default value (in this case green). This final visualization is shown in figure 5.14.

5.7 VizThis: User Evaluation

To measure the quality (or cognitive value) of the VizThis visualizations produced from the BBC Top 40 Music Chart data, we conducted an informal usability test of three different visualizations. The user tests were conducted according to the principles of discount usability engineering [Nie95].

We conducted an informal user evaluation on 6 subjects who came from a technical but non-computer science background. The purpose of this test was to evaluate how well the VizThis approach can produce cognitively useful visualizations with varying levels of human involvement (thus measuring the value of the semantic assistance which is provided by the system). The first visualization (figure 5.15) was produced by the system with no human involvement (AutoMap facility only). The second visualization (figure 5.16) was produced with a human user performing some data cleansing. The third visualization (figure 5.17) was produced with human involvement for data cleansing and the tweaking of mappings. Each of the visualizations shows the chosen source entity to target artefact mappings in the bottom left hand corner. This is provided in an attempt to aid the subjects' cognition. We used data from the BBC Top 40 chart music [BBC08] XML feed [Web08b].

We asked each subject to evaluate the quality of the 3 visualizations. The subjects were shown each visualization (A, B and C) in turn and given 4 minutes to explain what they thought was being represented. We found that subjects were able to comprehend some aspects of each of the visualizations. This was helped by the subjects being able to consult



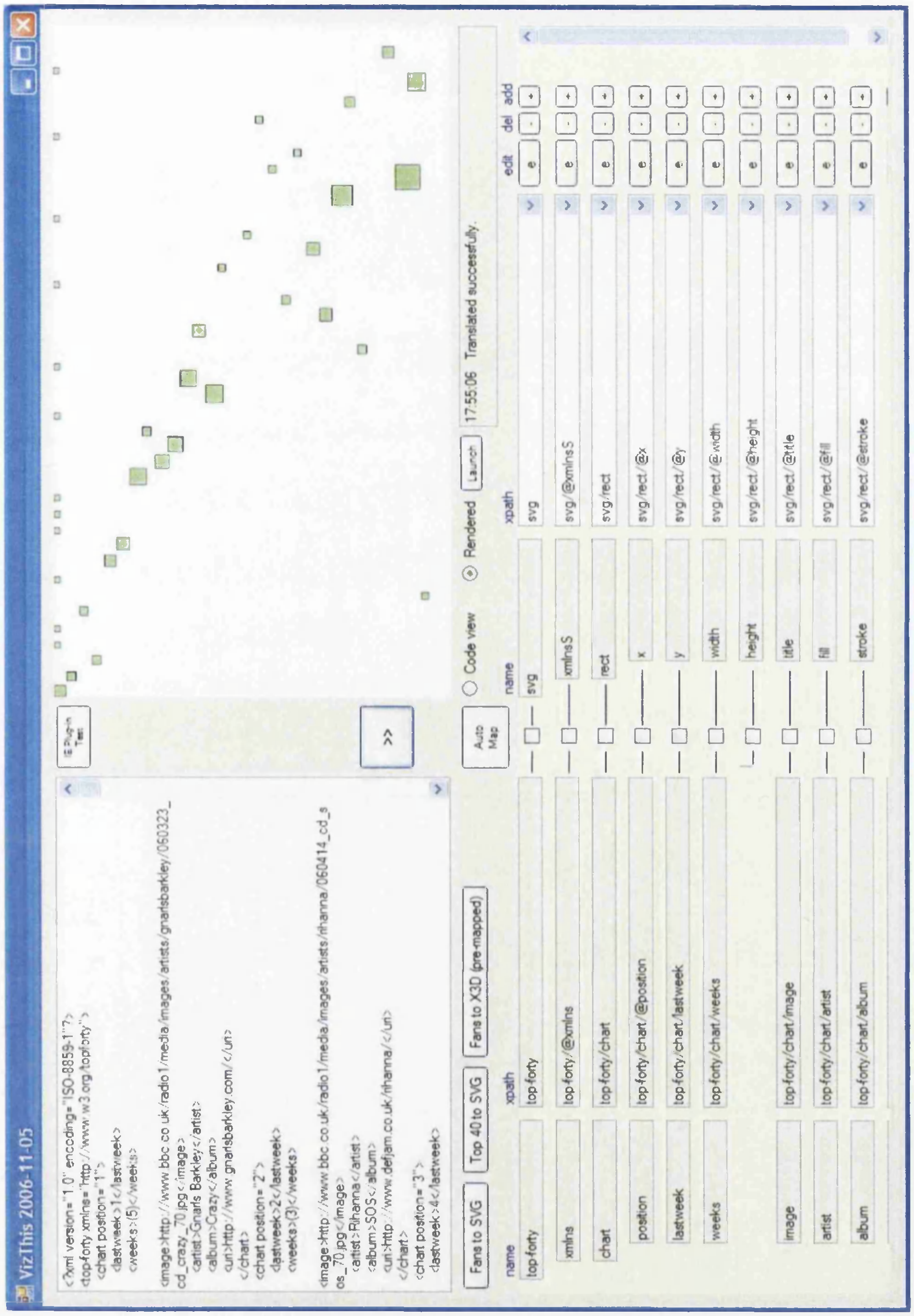
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<top-forty xmlns="http://www.w3.org/topforty">
 <chart position="1">
  <lastweek>1</lastweek>
  <weeks>(5)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/gnarlsbarkley/060323_
cd_crazy_70.jpg</image>
  <artist>Gnarls Barkley</artist>
  <album>Crazy</album>
  <uri>http://www.gnarlsbarkley.com/</uri>
 </chart>
 <chart position="2">
  <lastweek>2</lastweek>
  <weeks>(3)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/rihanna/060414_cd_s
os_70.jpg</image>
  <artist>Rihanna</artist>
  <album>SOS</album>
  <uri>http://www.defjam.co.uk/rihanna/</uri>
 </chart>
 <chart position="3">
  <lastweek>4</lastweek>
```

17:51:09 Translated successfully.

name	xpath		name	xpath	edit	del	add
top-forty	top-forty	□	svg	svg	e	-	+
xmlns	top-forty/@xmlns	□	xmlnsS	svg/@xmlnsS	e	-	+
chart	top-forty/chart	□	rect	svg/rect	e	-	+
position	top-forty/chart/@position	□	x	svg/rect/@x	e	-	+
lastweek	top-forty/chart/lastweek	□	y	svg/rect/@y	e	-	+
weeks	top-forty/chart/weeks	□	width	svg/rect/@width	e	-	+
		□	height	svg/rect/@height	e	-	+
image	top-forty/chart/image	□	title	svg/rect/@title	e	-	+
artist	top-forty/chart/artist	□	fill	svg/rect/@fill	e	-	+
album	top-forty/chart/album	□	stroke	svg/rect/@stroke	e	-	+

Figure 5.13: The BBC Top 40 data visualised. Notice that the number of weeks the song has been in the charts is now conveyed as the length of the side of each square.

the mapping table of data entities to representation artefacts (shown in the bottom left corner of each visualization).



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<top-forty xmlns="http://www.w3.org/topforty">
  <chart position="1">
    <lastweek>1</lastweek>
    <weeks>(5)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/gnarlsbarkley/060323_
cd_crazy_70.jpg</image>
    <artist>Gnarls Barkley</artist>
    <album>Crazy</album>
    <uri>http://www.gnarlsbarkley.com/</uri>
  </chart>
  <chart position="2">
    <lastweek>2</lastweek>
    <weeks>(3)</weeks>

<image>http://www.bbc.co.uk/radio1/media/images/artists/rihanna/060414_cd_s
os_70.jpg</image>
    <artist>Rihanna</artist>
    <album>SOS</album>
    <uri>http://www.defjam.co.uk/rihanna/</uri>
  </chart>
  <chart position="3">
    <lastweek>4</lastweek>
```

Fans to SVG | Top 40 to SVG | Fans to X3D (pre-mapped)

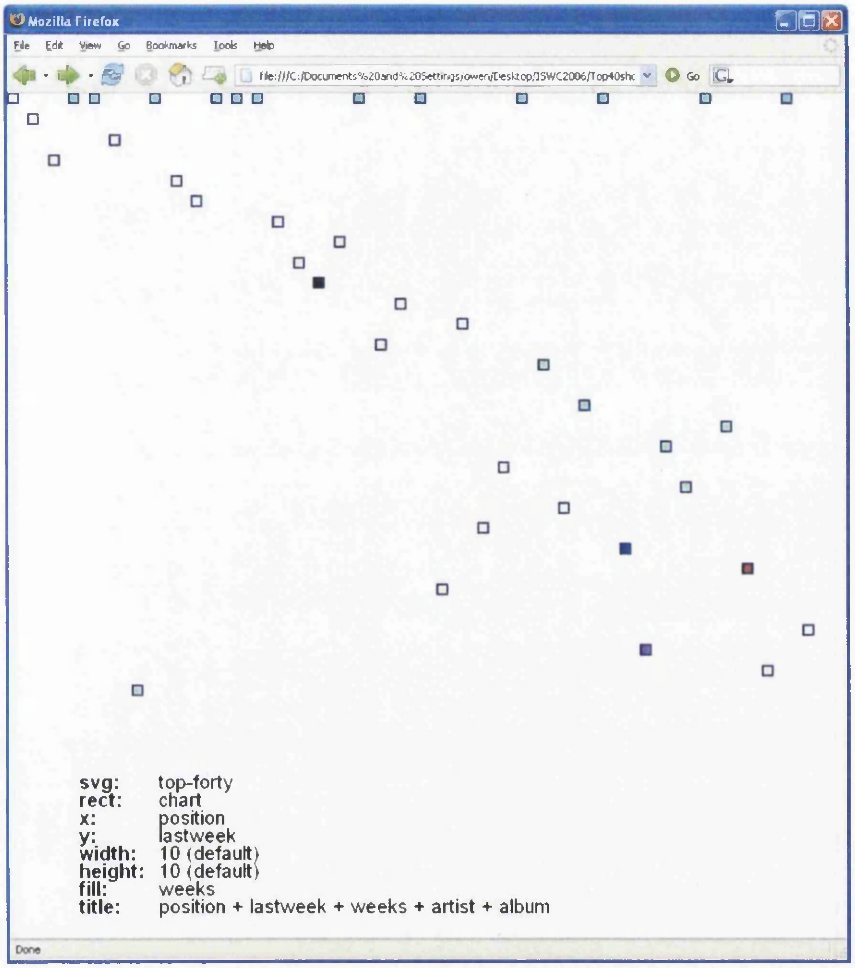
17:55:06 Translated successfully.

name	xpath	name	xpath	edit	del	add
top-forty	top-forty	svg	svg	e	-	+
xmlns	top-forty/@xmlns	xmlnsS	svg/@xmlnsS	e	-	+
chart	top-forty/chart	rect	svg/rect	e	-	+
position	top-forty/chart/@position	x	svg/rect/@x	e	-	+
lastweek	top-forty/chart/lastweek	y	svg/rect/@y	e	-	+
weeks	top-forty/chart/weeks	width	svg/rect/@width	e	-	+
		height	svg/rect/@height	e	-	+
image	top-forty/chart/image	title	svg/rect/@title	e	-	+
artist	top-forty/chart/artist	fill	svg/rect/@fill	e	-	+
album	top-forty/chart/album	stroke	svg/rect/@stroke	e	-	+

Figure 5.14: The BBC Top 40 data visualised. Notice that we have removed the confusing colouring of each square and instead have one colour for each square.

5.7.1 Visualization A

The following cognitive insights were observed by subjects:


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/owen/Desktop/ISWC2006/Top40shc
Go
Done

svg:	top-forty
rect:	chart
x:	position
y:	lastweek
width:	10 (default)
height:	10 (default)
fill:	weeks
title:	position + lastweek + weeks + artist + album

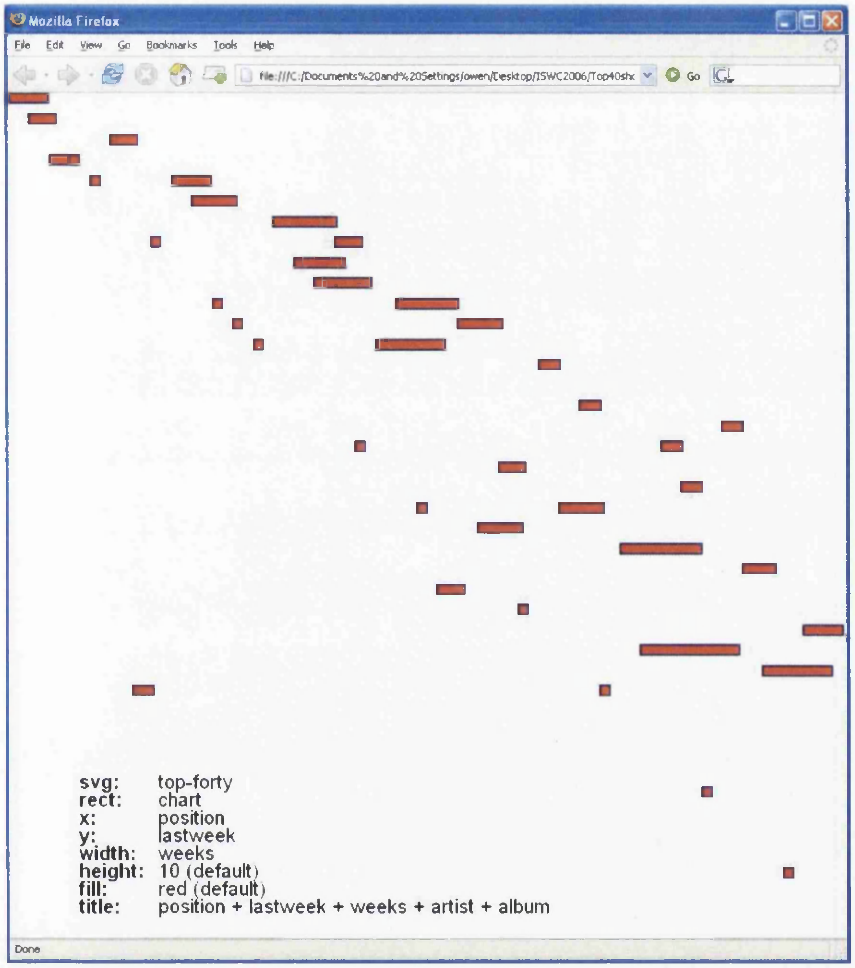
Figure 5.15: Visualization A - AutoMap only. No human intervention

- Four subjects said they could see a linear trend along the line x=y (when the origin is the top left hand corner). This represents the variation between the current chart position and last week's chart position.
- Three subjects noticed the outlier object in the bottom left corner which represents the chart's highest-climber.
- Five subjects noticed that there were certain objects at the top of the visualization. These represent songs which are new entries.

5.7.2 Visualization B

Subjects also noticed the following:

- Five subjects noticed small squares exactly on the line x=y representing new entries.


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/owen/Desktop/ISWC2006/Top40shx
Go
Done

svg:	top-forty
rect:	chart
x:	position
y:	lastweek
width:	weeks
height:	10 (default)
fill:	red (default)
title:	position + lastweek + weeks + artist + album

Figure 5.16: Visualization B - Data Cleansed, followed by AutoMap

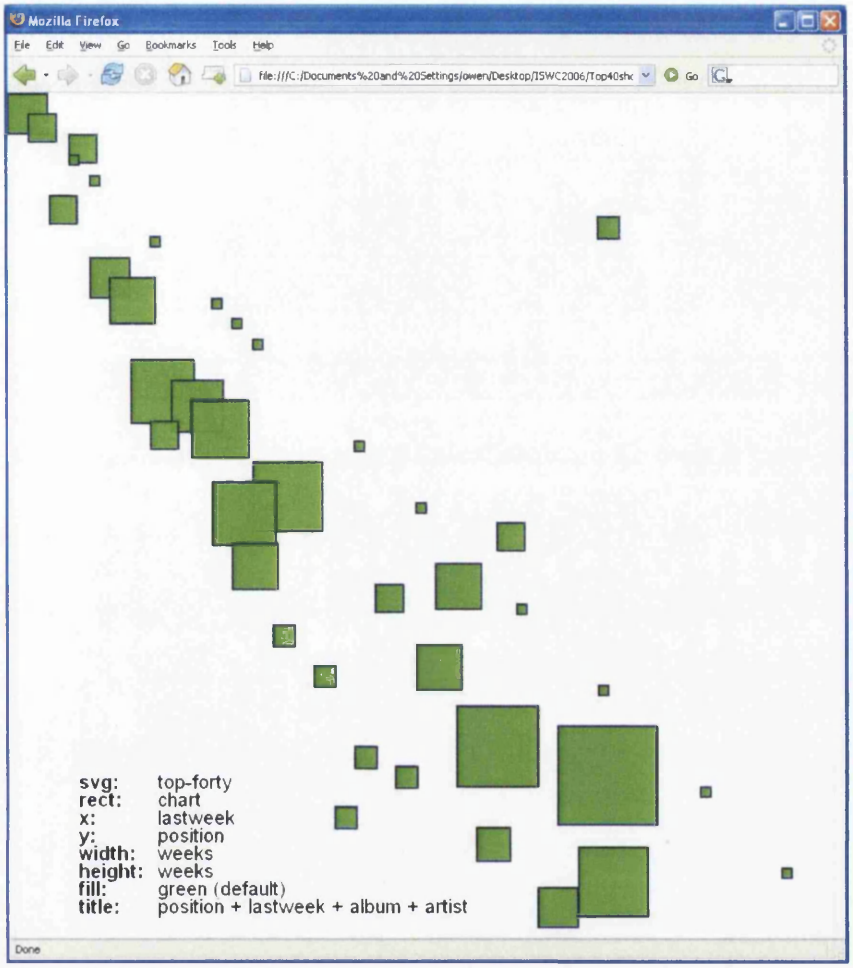
- Five subjects noticed that different widths of the bars indicated the number of weeks the song has been in the charts.

5.7.3 Visualization C

Subjects additionally noticed the following:

- Six (all) subjects said that the size of the squares represented weeks in the chart.
- Six (all) subjects noticed the outlier object.
- Three subjects noticed that the majority of objects were now on the other side of the x=y line.

These results are positive since they indicate that, although not perfect, "Visualization as Mapping" produces results which have cognitive value, even with no, or limited human involvement. This is a good indication of the accuracy of the semantic assistance provided


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/owen/Desktop/ISWC2006/Top40shc
Go
Done

svg:	top-forty
rect:	chart
x:	lastweek
y:	position
width:	weeks
height:	weeks
fill:	green (default)
title:	position + lastweek + album + artist

Figure 5.17: Visualization C - Data Cleansed, followed by AutoMap and mappings tweaked.

by the system.

5.8 Summary

In this chapter, we have described a user interaction paradigm which has been created from the application of Tree-centric mapping techniques to the area of Information Visualization in an XML-based environment. Through the formalisation of this process with Visualization As Mapping techniques, a number of advantages have been achieved. These advantages reduce the work burden on the user. We discuss the advantages and disadvantages of this approach below together with the implications in relation to the next chapter.

5.8.1 Advantages

Automaticity The mapping of source data entities to target representation artefacts will always involve human intervention in order to produce the best visualizations. However, much of the mapping process can be automated, or at least a "semantically intelligent" guess made.

Constraints When source entities and target artefacts are considered within a mapping context, we can define their nature and behavior in a way which allows us to derive constraints. This allows us to constrain which entities can be mapped to which artefacts, thus decreasing the number of possibilities, simplifying the system and reducing users' work.

Generality Since our technique is valid for any semantically rich XML markup language, a number of different source formats from many domains can be supported. This generality also applies to the target format used.

Multi-modal target Our technique generalises the source file into data entities and the target file into representation artefacts. This allows us to represent any data in any media representation. This is not limited to graphical representations, but also textual and audial too.

However, there are also some disadvantages associated with viewing Information Visualization in terms of Tree-centric mapping:

5.8.2 Disadvantages

Specificity When we view Information Visualization using Mapping concepts, there is a danger of forgetting about the objectives of visualization. Also we must be careful to not confuse the user by using mapping related terminology.

Exclusivity Our technique will only handle well-defined formats which are expressed in common markup languages (XML, XML/XLink). If they are not expressed in one of these formats, either they must be converted, or a proprietary tool built for producing the visualization. For example, many graphics formats are binary based, or are only accessible through a programming API.

AutoMap limitations AutoMap only produces one possible mapping. It tries to give a definitive answer. There is no facility for the system to produce multiple possibilities which are ranked according to their fitness score.

Visualization techniques The techniques which can be produced using visual object primitives (at least in SVG and X3D) is limited. Therefore, the sophistication of the visualization techniques which can be employed is also limited. Unless we consider additional process stages before generating the target file, we can only generate relatively primitive visualization styles (e.g., 2D charts). Visualization techniques which are more complicated (e.g., TreeMaps or Parallel Coordinates) require a level of sophistication of SVG or X3D which may not be possible with simple mapping techniques and may require a procedural approach.

XML idiosyncrasies When dealing with a Tree-centric pipeline based on XML, we must deal with the implementation specific nuances associated with each format. Since the XML specification does not prescribe a particular data modelling style, there can be many techniques to convey semantics. Any mapping system must be able to deal with these different techniques and be able to map between them. For example, child entities can be conveyed as attributes of an element, or new child elements. Also, some XML formats encode multiple values in a single XML element value pair. This is common in X3D (e.g., the RGB="255 255 0" style). An effort has been made to standardise data modelling in XML through the XML Normal Form (XNF)[Tho01]. However, this has gained little traction.

The disadvantage of Specificity can be reduced by carefully crafting a user interface, with frequent user feedback and a good focus on the problem of Information Visualization. The disadvantages of Exclusivity will be naturally reduced as the trend towards producing data in semantically rich formats such as XML and RDF continues.

However, there are some fundamental design problems with the approach we have taken in implementing VizThis and taking the Tree-centric mapping approach with XML. This is illustrated by the following scenario.

The AutoMap algorithm works well when creating an SVG visualization due to the relative simplicity of the SVG schema design. However, when creating an X3D visualization the algorithm produces poor results. This is because AutoMap has to deal with the idiosyncrasies of X3D's design before it can even begin to deal with the best mappings from a visualization perspective. This is too complex a task to achieve good results. VizThis is trying to solve the problem at a level of abstraction which is too low (i.e. at the level of individual XML entities and primitive graphical artefacts). Therefore, we need a higher level of abstraction before we can solve non-trivial problems.

In summary, the VizThis interaction metaphor has produced some encouraging results which have been evaluated in a small-scale, informal, qualitative user evaluation. However, for the system to be a success we need to consider mapping at a higher level of abstraction and also have the ability to produce more sophisticated visualizations. We address these limitations in a new model which we describe in the next chapter (chapter 6).

Chapter 6

SemViz : From Web Data to Visualization via Ontology Mapping

Contents

6.1 Introduction

In this chapter, we introduce SemViz which takes an Ontology-centric approach to the visualization pipeline using Ontologies with Certainty Factors (OCF). Preliminary results from this work were presented at EuroVis 2008, Eindhoven where the paper won Best Paper Award [GSGC08]. Before describing the details of SemViz, it is beneficial to summarise the work from the previous chapter (chapter 5).

In the previous chapter we described VizThis which takes an XML-centric approach to mapping from source data to target visualization. We discussed the merits of this approach together with the disadvantages. There were 3 main disadvantages:

1. The VizThis system deals with XML primitives (elements and attributes), rather than higher level data and visualization objects. This means that there is a large overhead in dealing with mapping low-level primitives between the source and target formats.

2. The visual objects supported by the graphical display languages used by VizThis (SVG and X3D) are too low-level to easily create more sophisticated information visualization techniques such as TreeMaps and Parallel Coordinates.

3. VizThis attempts to provide the user with a definitive answer as to the best visualization. There may be additional "just as good" visualizations which the user will miss out on seeing because of the tool's rigid constraint-based system.

In this chapter we present SemViz which deals with the mapping process at a higher level of abstraction. SemViz addresses the three disadvantages above as follows:

1. The whole visualization pipeline uses ontologies as its core data structure. As such, the idiosyncrasies of the use of different XML primitives are eliminated because ontologies are more disciplined in their use of parent and child relationships. These are represented as concepts and attributes (see section 3). Also, ontologies have the ability to define more formally the semantics conveyed.

2. We utilise existing public domain visualization toolkits (i.e., ILOG Discovery and Prefuse) to create more sophisticated visualizations such as TreeMaps and Parallel Coordinates.

3. SemViz does not attempt to provide the user with a definitive "best visualization". Instead, it ranks all possible visualizations and presents users with a selection of the best visualizations which they are free to explore further. Therefore, SemViz is not based on a rigid constraint-based system, but a scoring system which produces prioritised results.

6.1.1 Usage Scenario

In this chapter we demonstrate the application of SemViz as a visualiser of information on the web. There is a constant growth in the volume of information on the web which can be usefully visualized. This often takes the form of tables of information and lists. Ideally, this information would be presented in a semantically rich format with a well defined schema and an accompanying ontology (commonly known as the Semantic Web [BLHL01]). However, in practice, the information is mixed into visually appealing, but semantically non-machine readable web pages, with no defined schema and certainly no ontology. However, our need to visualize the information on these pages remains strong - the information is useful, and there is a large volume. We must therefore create a more pragmatic solution based on existing technologies and available tools.

6.1.2 System Overview

The novel design of this pipeline features three ontologies, namely a *domain ontology* (DO) for storing domain knowledge about the source data (i.e., music charts in this chapter), a *visual representation ontology* (VRO) for storing the knowledge about visualization tools, styles and parameter space, and a *semantic bridging ontology* (SBO) for storing the knowledge about the mapping from DO to VRO. We use an ontology mapping technique

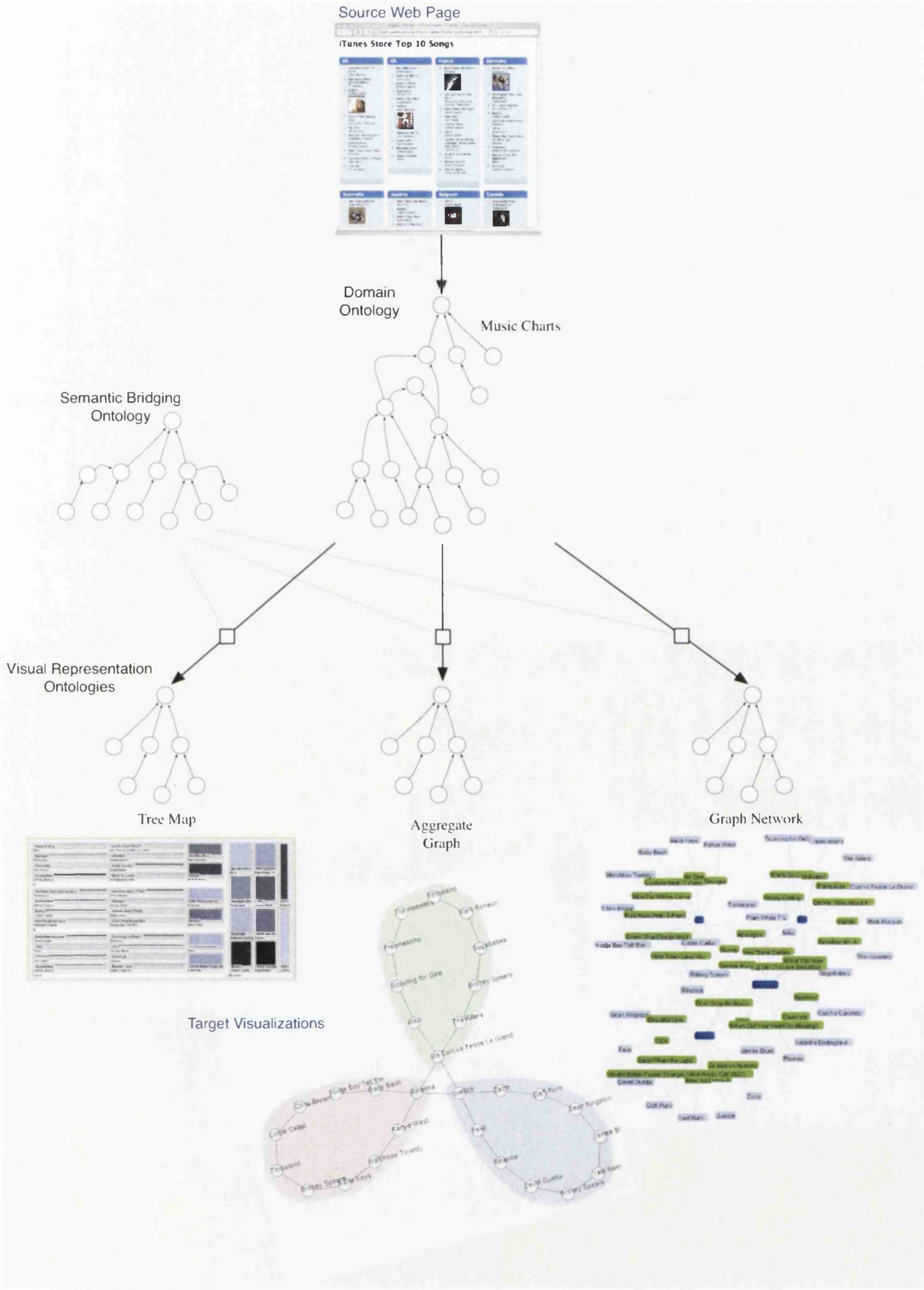
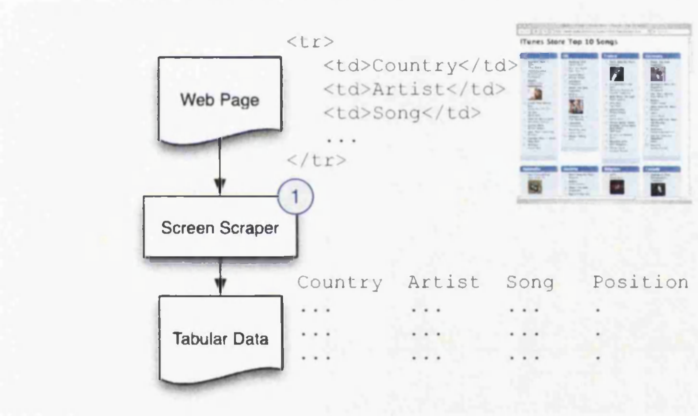


Figure 6.1: SemViz pipeline showing: Domain Ontology (DO); Semantic Bridging Ontology (SBO); and Visual Representation Ontology (VRO).

inspired by principles from OMEN (Ontology Mapping Enhancer) [MNJ05] to realise the automatic data mapping within the pipeline, and create interfaces between the pipeline and two popular visualization tools, ILOG Discovery [BHS03] and Prefuse [HCL05]. We also use the concept of Certainty Factors [Cha08] to weight relationships in the ontologies. Figure 6.1 shows an example web page containing the iTunes Store song chart, and visualizations generated by ILOG Discovery and Prefuse automatically via our pipeline.

6.2 SemViz: Pipeline Stages

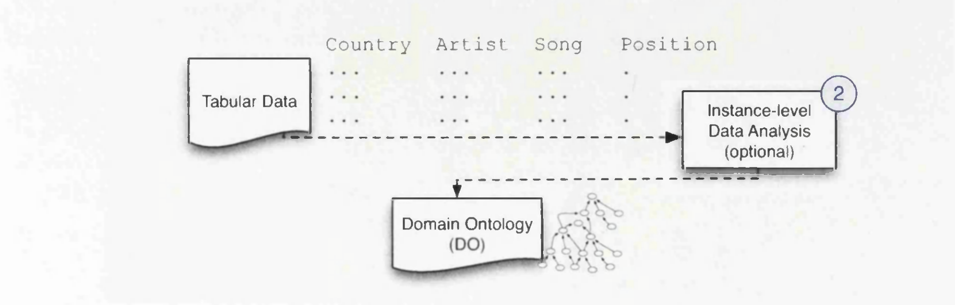
We have developed a prototype, SemViz which is able to produce an end-to-end automatic visualization of tabulated data from a selection of music chart web pages. SemViz allows the mapping algorithm's parameters to be adjusted and includes custom code for interfacing to the visualization toolkits. We choose to output visualizations using either the ILOG Discovery or Prefuse visualization toolkits. These public domain tools are relatively easy to interface with and also provide a variety of visualization styles. The SemViz pipeline stages are as follows:



```
<tr>
   <td>Country</td>
   <td>Artist</td>
   <td>Song</td>
   ...
</tr>
```

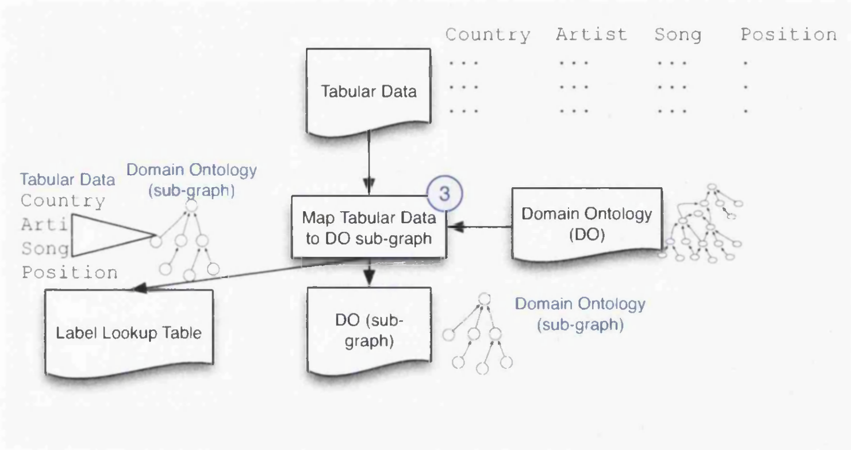
```
Country  Artist  Song   Position
...      ...     ...    .
...      ...     ...    .
...      ...     ...    .
```

Stage 1. Extract Tabular Data from Web Page. If an XML or CSV link to the tabulated data is not provided, a screen-scraper/data extractor such as Solvent and Piggy Bank [HMK07] can be applied. The system needs the source data to be in a tabular format. For example, the *Web Page* code is in HTML and therefore contains formatting data which is superfluous to the visualization pipeline. Therefore, the tabular data must be extracted from the HTML. An example of this is shown to the right of the *Tabular Data* output.

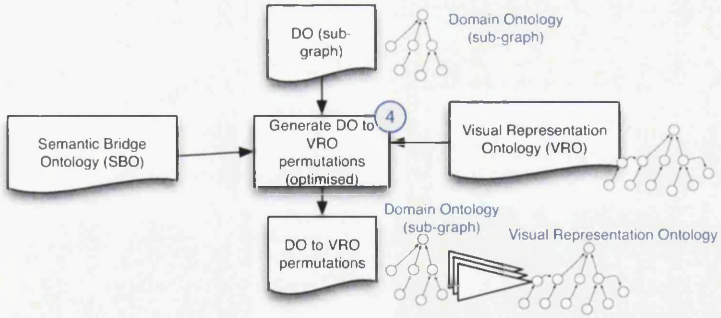


```
Country  Artist  Song   Position
...      ...     ...    .
...      ...     ...    .
...      ...     ...    .
```

Stage 2. Perform Instance-level Data Analysis on Source Data. This stage is optional, but can be used to augment the Domain Ontology, particularly if there is a large amount of data where valuable semantics can be usefully extracted.

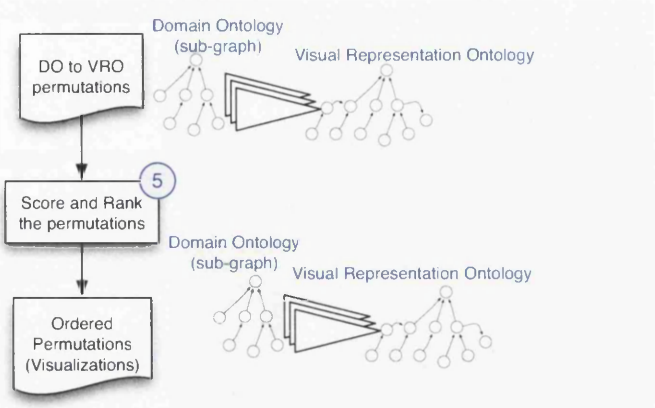


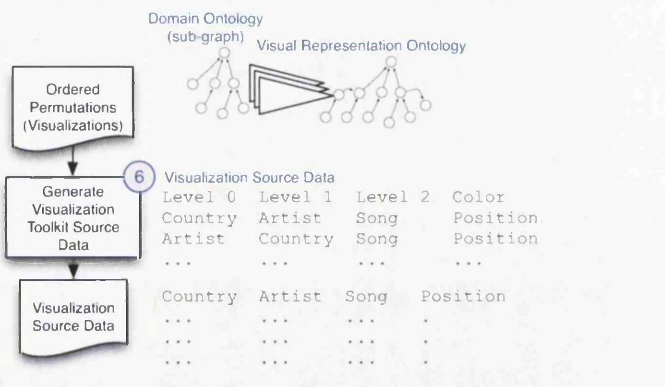
Stage 3. Map Tabular Data to Domain Ontology sub-graph. This component uses string similarity measures of the data column and domain ontology concept names and also the instance data to probabilistically reason on the most likely mappings. Each mapping permutation is scored and the top n of the possible permutations are stored. This is a schema mapping process.


DO (sub-graph)
Domain Ontology (sub-graph)
Semantic Bridge Ontology (SBO)
Generate DO to VRO permutations (optimised)
4
Visual Representation Ontology (VRO)
DO to VRO permutations
Domain Ontology (sub-graph)
Visual Representation Ontology

Stage 4. Create the Domain Ontology to VRO Permutations. Depending on which concepts in the Domain Ontology have been stimulated by the Source Data, the Mapper uses the rules stored in the Semantic Bridging Ontology to create mappings which aim to result in useful visualizations. Each mapping permutation is scored, and the top m of the permutations are stored. This is a schema mapping process. The Ontology Mapping algorithm is described in section 6.7.

Stage 5. Score and Rank the Permutations. With the top n permutations from stage 3 and m permutations from stage 4, this results in $n \times m$ possible mapping permutations. Each permutation is given a score, they are ranked, and the highest 10 scoring permutations are combined with the original tabulated source data to form 10 VRO instances. In SemViz, 10 are chosen as a good trade-off but this value is a parameter. The 10 VRO instances are converted into the specific files necessary for each Visualization toolkit supported by the system.

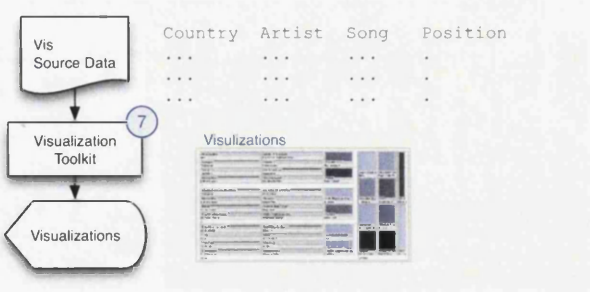

DO to VRO permutations
Domain Ontology (sub-graph)
Visual Representation Ontology
5
Score and Rank the permutations
Domain Ontology (sub-graph)
Visual Representation Ontology
Ordered Permutations (Visualizations)


Domain Ontology (sub-graph)
Visual Representation Ontology
Ordered Permutations (Visualizations)
6
Generate Visualization Toolkit Source Data
Visualization Source Data
Visualization Source Data

Level 0	Level 1	Level 2	Color
Country	Artist	Song	Position
Artist	Country	Song	Position
...	...	...	...

Country	Artist	Song	Position
...	...	...	.
...	...	...	.
...	...	...	.

Stage 6. Generate Visualization Toolkit Source Data. The VRO instances are converted into the specific files necessary for each Visualization toolkit supported by the system.

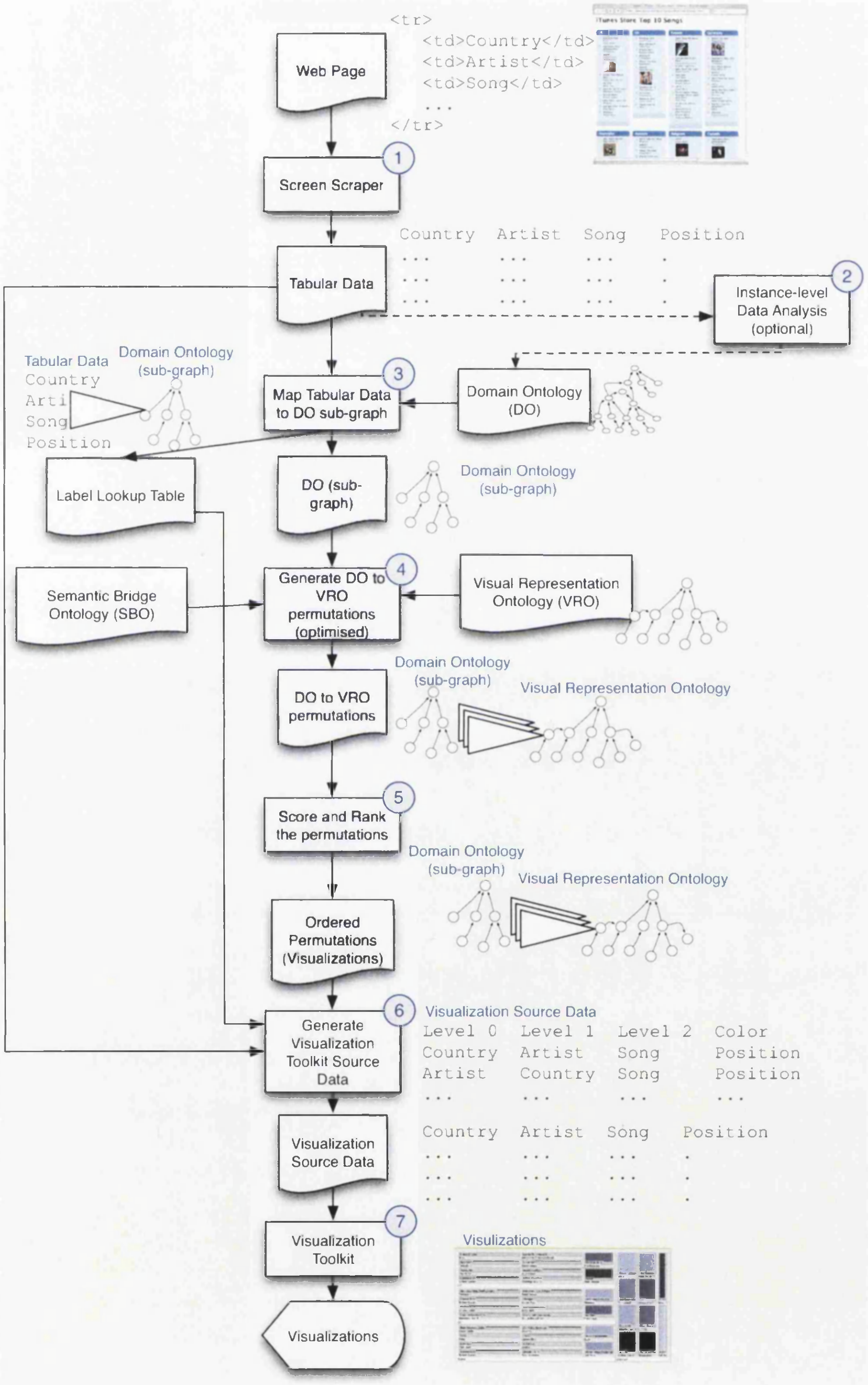


Country	Artist	Song	Position
...	...	...	.
...	...	...	.
...	...	...	.

Stage 7. Invoke Visualization Toolkits. The toolkits are invoked and the visualizations are generated before being presented to the user.

All of the stages above (except stage 1) are automatic in that no user intervention is required. Stage 1 requires that the user present the SemViz tool with source data in CSV format (with the first row representing the column heading names). Most web pages do not provide their data in this format, therefore the user must use a screen-scraper. The amount of manual interaction involved depends on the tool and the exact nature of the source webpage.

The full SemViz pipeline is shown in figure 6.2.



```
<tr>
   <td>Country</td>
   <td>Artist</td>
   <td>Song</td>
   ...
</tr>
```

```
Country  Artist  Song   Position
...      ...     ...    .
...      ...     ...    .
...      ...     ...    .
```

```
Visualization Source Data
Level 0  Level 1  Level 2  Color
Country  Artist   Song     Position
Artist   Country  Song     Position
...      ...      ...      ...
```

```
Country  Artist  Song   Position
...      ...     ...    .
...      ...     ...    .
...      ...     ...    .
```

Web Page
1
Screen Scraper
Tabular Data
2
Instance-level Data Analysis (optional)
Tabular Data
Domain Ontology (sub-graph)
Country
Arti
Song
Position
3
Map Tabular Data to DO sub-graph
Domain Ontology (DO)
Label Lookup Table
DO (sub-graph)
Domain Ontology (sub-graph)
Semantic Bridge Ontology (SBO)
4
Generate DO to VRO permutations (optimised)
Visual Representation Ontology (VRO)
Domain Ontology (sub-graph)
Visual Representation Ontology
DO to VRO permutations
5
Score and Rank the permutations
Domain Ontology (sub-graph)
Visual Representation Ontology
Ordered Permutations (Visualizations)
6
Generate Visualization Toolkit Source Data
Visualization Source Data
7
Visualization Toolkit
Visulizations
Visualizations

Figure 6.2: The SemViz technology pipeline, from Source web page to Target visualizations.

6.3 SemViz: Pipeline Ontologies

As discussed in section 3.1, an ontology provides an explicit conceptualisation (i.e., meta-information) that describes the semantics of data [Fen01]. A language for defining ontologies is syntactically and semantically richer than other common approaches (e.g., databases).

The concepts, relationships and attributes of the ontologies can be seen in Figure 6.4. Concepts (circles) are related via relations (arrows). For example, an "Artist" concept is related to a "Song" concept via a "has" relation. A concept can also have attributes (rectangles in the diagram). For example, an "Artist" concept has an "isPrimaryKey" attribute. The difference between an attribute and a relation is that an attribute points to a single primitive entity (e.g., an integer), whereas a relation points to another concept. When modelling domains using ontologies, there is a trade-off in deciding whether to use attributes or relations. Usually, a "policy decision" is made depending on the particular domain. For SemViz, we assert that attributes are only used to convey meta-information about concepts (i.e., isPrimaryKey, isQuantitative).

In previous work [GSG$^+$06], we have defined ontologies which capture the semantics of common textual and graphical XML formats. These ontologies focussed on the primitive type of the data types (e.g., integer or string) as opposed to the rich semantics of the concept (e.g., artist and album). The ontologies were implemented in the VizThis tool as described in chapter 5. The work in this chapter (SemViz) uses ontologies which capture the richer semantics of the concepts.

6.3.1 Certainty Factors

In SemViz, we employ a novel type of ontology which we call Ontologies with Certainty Factors (OCF). Traditional Ontologies are based on absolute knowledge with binary relationships (they are either present or not present). Any reasoning operations on Traditional Ontologies typically provide only one solution. This process is similar as that provided by VizThis (see section 5). Ontologies with Certainty Factors have weighted relationships known as Certainty Factors [Cha08]. The strength of these relationships varies between 0.0 and 1.0 (this differs slightly from MYCIN which used values between -1 and +1). In this way, we can represent uncertain knowledge. We believe that it is more realistic to capture a lot of knowledge with certainty factors than a small amount of absolute knowledge. Certainty factors are a hybrid of an ordinal scale that humans use to specify and uncertainty which is more or less suggestive; an interval scale for computers, numbers like 0.6 and 0.4; and a ratio scale for the actual arithmetic [Cha04]. A typical table of human readable descriptions (ordinal scale) mapped to a certainty factor (interval scale) is shown in table 6.1.

Every relation in a OCF has a certainty factor associated with it. This may be a default value provided by the system in cases where a human has not explicitly defined one. In this way, every OCF is a fully connected graph of relationships between all concepts. This allows us to take two ontologies and for any mapping permutation of concepts between those two

Description	Certainty Factor
strongly suggestive	0.8
suggestive	0.6
weakly suggestive	0.4
slight hint	0.2

Table 6.1: Human readable descriptions (ordinal scale) mapped to a certainty factor (interval scale).

	TO	OCF
Graph Representation of Domain Knowledge	yes	yes
Syntactically and Semantically rich	yes	can be
Hierarchy of concepts and properties	yes	can be
Example	Music Ontology Specification (MOS)	Visual Representation Ontology (VRO)
Relationship type	Absolute (0 or 1)	Certainty Factors (0.0 to 1.0)
Graph type	Partial	Fully connected
Format	RDF / OWL	RDF (with weighted properties through reification)

Table 6.2: Traditional Ontologies (TO) versus Ontologies with Certainty Factors (OCF).

ontologies, we can calculate a score. We discuss the SemViz algorithm in detail in section 6.7.

In table 6.2, we compare Traditional Ontologies (TO) and Ontologies with Certainty Factors (OCF). It can be seen that the two types of ontologies are very similar, with the main difference being that an OCF is a fully-connected graph which has relationships weighted with Certainty Factors. This difference is shown visually in the example in figure 6.3. This example captures the relationships between a Person, Address and Pet. On the left, we can see that the TO is rigid in the semantics conveyed. This is good in that very clear reasoning can be made about the relations between concepts. However, this rigidity does not allow the flexibility required for a scoring algorithm such as SemViz. With the OCF (right), the same schema is represented with a fully connected graph, representing all possible relationships between concepts. These relationships have certainty values ranging from 0.01 (very low certainty) to 0.9 (high certainty). In this way, the semantics captured by the OCF are more expressive and more flexible.

We choose Certainty Factors over Fuzzy Logic or probabilistic methods due to their simplicity. Certainty Factors are easier to comprehend than fuzzy logic or probability measures. This is important since the users of SemViz will need to "prime" the ontologies with values. Although these users may be domain experts, their domain of knowledge may not be computer science. As such, in order to accurately capture the knowledge of many

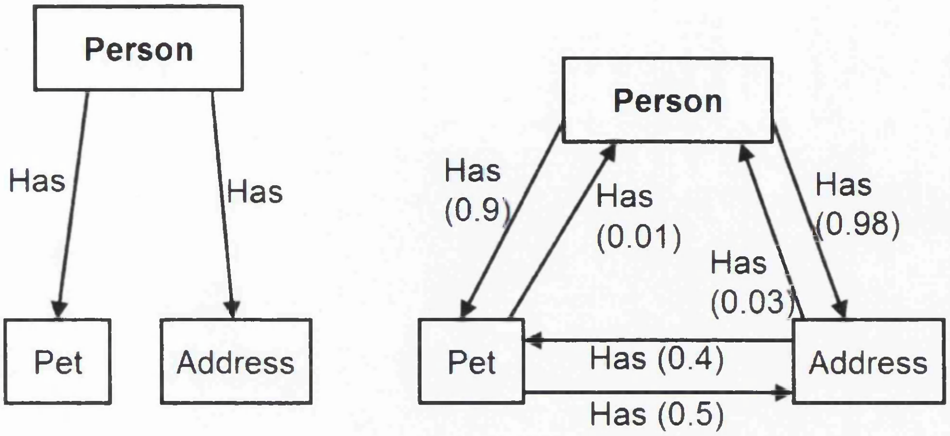


Figure 6.3: A simple ontology captured using Traditional Ontologies (TO) (left) and Ontologies with Certainty Factors (OCF) (right).

DO	VRO	Relationship / Attribute
synonyms	-	A
instanceHistory	-	A
has	contains	R
compliments	compliments	R
priorityWrt	priorityWrt	R
isQualitative	isQualitative	A
isQuantitative	isQuantitative	A
isPrimaryKey	isInformational	A
-	isMandatory	A

Table 6.3: Semantic Equivalence of relationships and attributes as used in the DO and VRO.

domains, we must make the process as straight forward as possible. This simplicity also allows Ontologies with Certainty Factors to be examined while they are an integral part of a working system. This allows a certain amount of alteration and "debugging" by domain experts.

The ontologies used in SemViz were prototyped using Stanford's Protege tool [NSD^{+}01] and expressed in RDF/OWL [LS98]. In the following worked example, we use the BBC's top 40 web page visualized as a 2D Graph. We restrict the diagrams to show only stimulated nodes and also relationships with significant weightings. The disadvantage of a fully connected graph is that the number of connections is $O(n^2)$. However, we also provide the complete ontology with all values in tabular form.

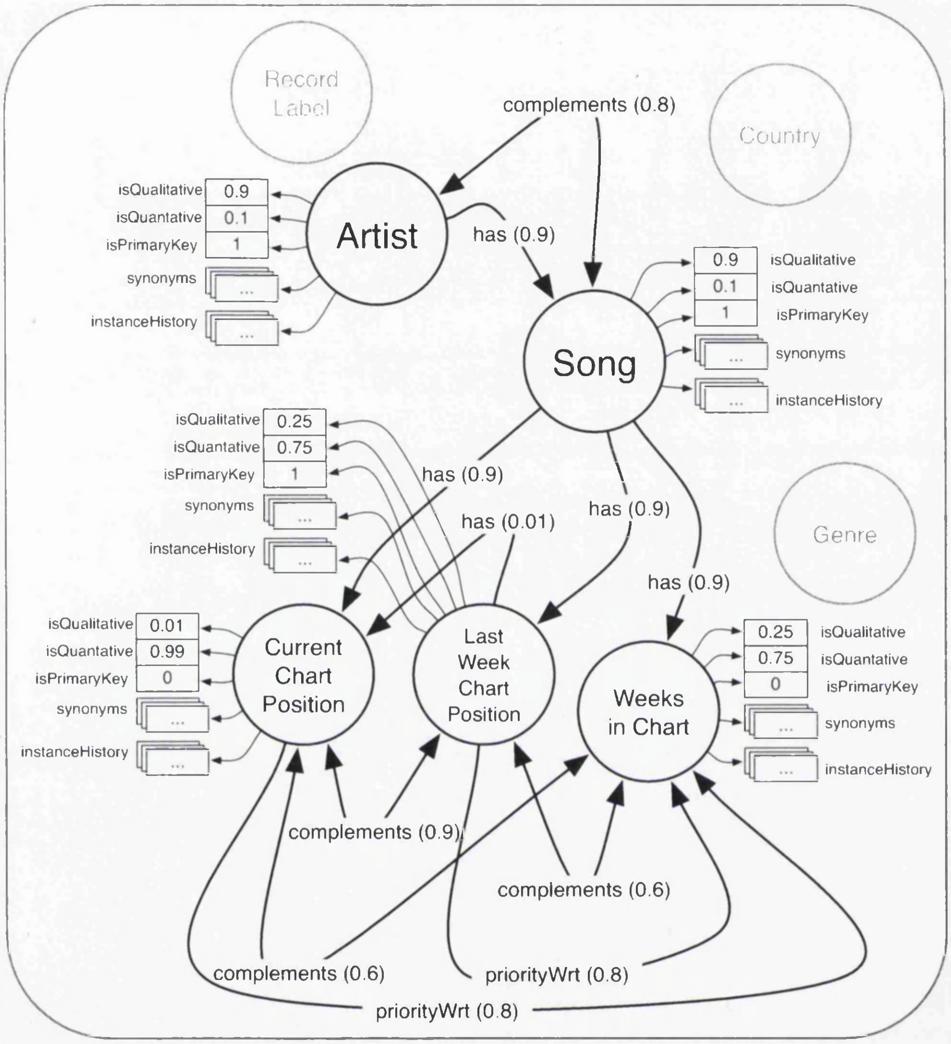


Figure 6.4: The Domain Ontology instance for the music charts subject area (as mapped to the BBC top 40 charts web page).

6.4 SemViz: Domain Ontology (DO)

The purpose of the Domain Ontology is to store the semantics of the subject area which the source web page covers. These semantics are derived in such a way that they can be easily mapped to artefacts in the Visual Representation Ontology (VRO) (see section 6.5). This is done by defining a controlled set of relationships and attributes for use in both the DO and VRO. Some of the relationships and attributes have *semantic equivalence*. This forms the basis of our ability to map between DO concepts (i.e., data entities) and VRO concepts (i.e., visual artefacts) and therefore produce cognitively useful visualizations. Table 6.3 lists the relationships and attributes used in the DO and VRO, together with their semantic equivalence (if applicable). Note that these relationships and attributes are general in that they can be applied to any new DO instance (e.g., car records) or VRO instance (e.g., 3D

Attribute	isQuantitative	isQualitative	isPrimaryKey	isPresent
Current Chart Position	0.97	0.01	0	1
Last Week Chart Position	0.91	0.12	0	1
Weeks in Chart	0.7	0.12	0	1
Artist	0.08	0.93	1	1
Song	0.05	0.91	1	1

has	Current CP	Last Week CP	Weeks in C	Artist	Song
Current Chart Position	n/a	0	0	0	0
Last Week Chart Position	0	n/a	0	0	0
Weeks in Chart	0	0	n/a	0	0
Artist	0.1	0.1	0.1	n/a	0.9
Song	0.6	0.6	0.6	0.2	n/a

complements	Current CP	Last Week CP	Weeks in C	Artist	Song
Current Chart Position	n/a	0.9	0.7	0.01	0.01
Last Week Chart Position	0.9	n/a	0.7	0.01	0.01
Weeks in Chart	0.7	0.7	n/a	0.01	0.01
Artist	0.01	0.01	0.01	n/a	0.9
Song	0.01	0.01	0.01	0.9	n/a

priorityWrt	Current CP	Last Week CP	Weeks in C	Artist	Song
Current Chart Position	n/a	0.6	0.8	0.3	0.3
Last Week Chart Position	0.4	n/a	0.6	0.3	0.3
Weeks in Chart	0.2	0.4	n/a	0.3	0.3
Artist	0.7	0.7	0.7	n/a	0.6
Song	0.7	0.7	0.7	0.4	n/a

Table 6.4: All weightings for the Music Domain Ontology (BBC Top 40 Music Chart subgraph).

Graph) which may be added to the SemViz system.

The DO for the Music Charts domain is shown in Figure 6.4. Each relationship and attribute has a *certainty value* which is a real number between 0.0 (weakest) and 1.0 (strongest). The only exception to this is the "priorityWrt" (priority with respect to) relationship which is 0.5 if the two linked concepts are of equal priority, or greater than 0.5 if the source concept has a higher priority than the target concept. The system records the certainty factor of all relationships and attributes, no matter how strong or weak. In fact, there are relationships between every concept. These are present because the pipeline is based on certainty factor reasoning where we score permutations in order to decide on the best mapping. In general, a DO is initially created by a domain expert who "primes" the ontology with appropriate values for the relationship strengths. However, as relationships and attributes are strengthened or weakened by user feedback, then the strength values will alter to reflect this.

The first mapping stage of SemViz is between the Tabulated Source Data and the DO (stage 3 in Figure 6.2). This process uses string similarity (Levenshtein distance [Lev65]) to measure the likelihood of a column in the source data having a match with a concept in the DO. String similarity is performed on both the column/concept names and the instance data. The DO keeps a record of concept name history (*synonyms* attribute) and instance value history (*instanceHistory* attribute). The score for the mapping of a column to a concept is based on the top similarities of the concept name synonyms plus the proportion of historical instances which are the same as the instances in the source. The second mapping stage between DO and VRO is based on Ontology Mapping concepts and is discussed in Section 6.7.

In Figure 6.4 it can be seen that the BBC Top 40 charts web page has been mapped to the "Artist", "Song", "Current Chart Position", "Last Week Chart Position", and "Weeks In Chart" concepts in the DO. These 5 mapped concepts are known as the *stimulated* concepts of the DO. They therefore represent a sub-graph of the full DO.

In table 6.4, we show the weightings of all relationships in the Music Domain Ontology sub-graph. The table is split into 4 smaller tables: Attribute Weightings; *has* Weightings; *complements* Weightings; and *priorityWrt* Weightings.

Each value in the table represents a Certainty Value [Cha08]. For example, the certainty that *Current Chart Position* is a quantitative value (i.e., *isQuantitative*) is 0.97 (very high). Conversely, the certainty that *Artist isQuantitative* is 0.08 (very low). We provide similar certainty values for *isQualitative*, *isPrimaryKey*, and *isPresent.* The certainty values for *isPrimaryKey* are either 0 or 1 since we know with a high degree of accuracy whether a source data entity is a primary key or not. Note that *isPresent* is a "control attribute". It is set to 1 (true) if the source data entity has a target visual artefact mapped to it. Otherwise it is set to 0 (false). We use this "control attribute" together with the *isMandatory* attribute in order to ensure that all mandatory target artefacts have a mapping. If not, they are invalid mappings (i.e., scored as zero). Also note, we state that *isPrimaryKey* in the DO has semantic equivalence with *isInformational* in the VRO (see table 6.3). We state this because primary key entities often have values which are of a high importance and therefore need to be mapped to visual artefacts which show their value directly.

The lower three sub-tables in 6.4 represent the certainty weightings for *has*, *complements* and *priorityWrt* respectively. The table is read by first selecting an entity from the rows on the left ("Current Chart Position", "Last Week Chart Position", ... , "Song") and then choosing a column ("Current CP", "Last Week CP", ... , "Song"). Note that some abbreviation is used for the column names due to space constraints. For example, an *Artist has* a *Song* with certainty 0.9 (high). However, *Current Chart Position has Last Week Chart Position* with certainty 0 (low). The *has* relationship represents a containment relationship between entities. The *complements* relationship is used to signify to what degree entities' values change in relationship with each other (and if the nature of how this relationship differs is of interest). For example, the *age* of a person and their *height* would be two entities with a high *complements* certainty value (since, to a certain degree, their values increase with each other). However, the *name* of a person and their *weight* would have a low *certainty* value (since the values are not typically related). The *priorityWrt* relationship is used to capture the relative priorities between entities. For example, a person's *Date Of Birth* has a higher priority than *Hair Colour.* The *priorityWrt* certainty is 0.5 if the two linked concepts are of equal priority, or greater than 0.5 if the source concept has a higher

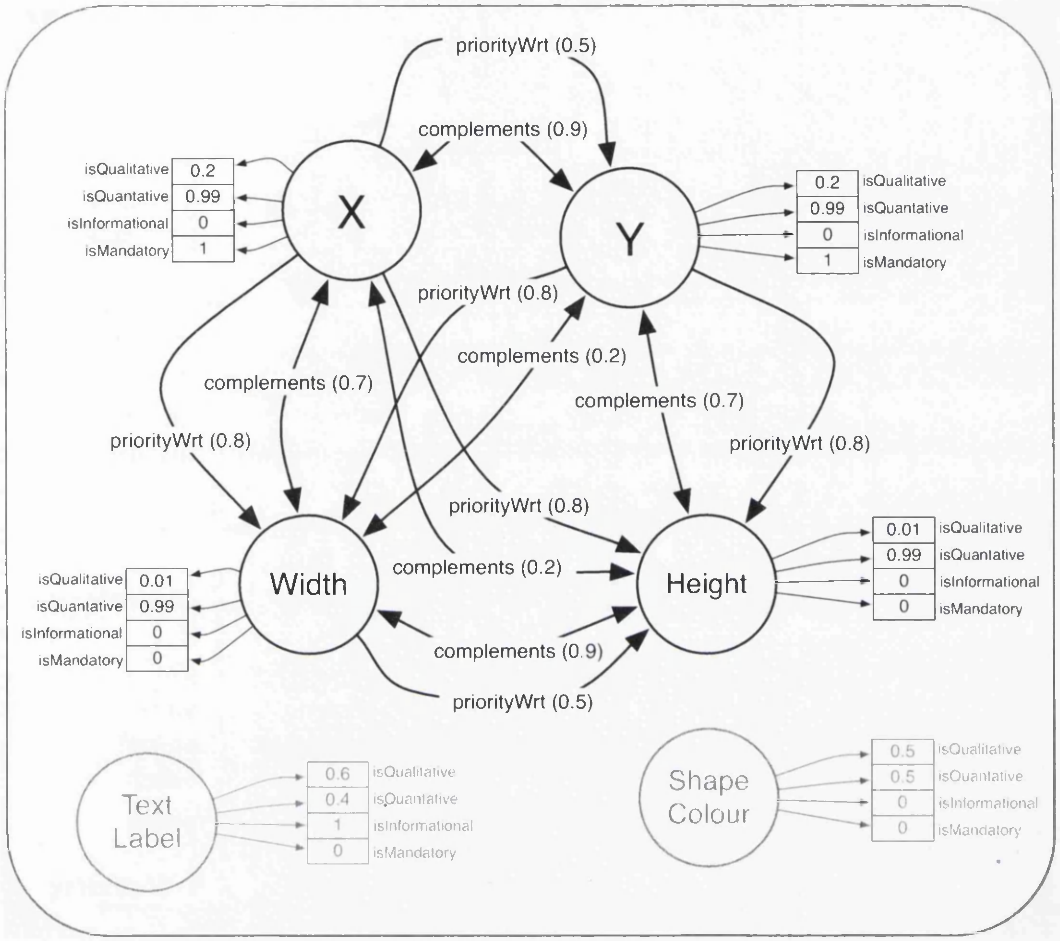


Figure 6.5: The Visual Representation Ontology instance for a 2D Graph (as mapped to the DO instance in Figure 6.4).

priority than the target concept.

6.5 SemViz: Visual Representation Ontology (VRO)

The VRO captures the semantics of a particular visual representation style (e.g., 2D Graph). It does this by modelling visual artefacts (e.g., X coordinate, Y coordinate, Colour, etc.) as concepts and the relationships between them. In this way, we can match relationships in the DO with relationships which have semantic equivalence in the VRO. We can also perform a similar task with semantically equivalent attributes.

We have built VRO's for 2D graphs (see Figure 6.5), TreeMaps, Parallel Coordinates and Graph Networks. The major source of information during the domain modelling exercise was ILOG Discovery. Its user interface has a Projection Inspector (see Figure 6.6) which allows users to control the mappings between source data entities (e.g., "Current chart position") and target visualization artefacts (e.g., "X coordinate"). ILOG

Attribute	isQuantitative	isQualitative	isInformational	isMandatory
X	0.9	0.1	0.01	1
Y	0.9	0.1	0.01	1
Width	0.9	0.1	0.01	0
Height	0.9	0.1	0.01	0
Label	0.1	0.9	0.9	0
Color	0.9	0.5	0.1	0

Contains	X	Y	Width	Height	Label	Color
X	n/a	0.01	0.01	0.01	0.01	0.01
Y	0.01	n/a	0.01	0.01	0.01	0.01
Width	0.01	0.01	n/a	0.01	0.01	0.01
Height	0.01	0.01	0.01	n/a	0.01	0.01
Label	0.01	0.01	0.01	0.01	n/a	0.01
Color	0.01	0.01	0.01	0.01	0.01	n/a

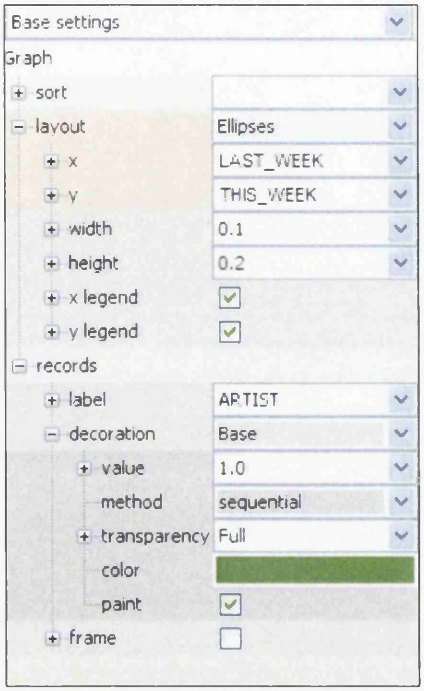
Complements	X	Y	Width	Height	Label	Color
X	n/a	0.9	0.5	0.2	0.01	0.01
Y	0.9	n/a	0.2	0.5	0.01	0.01
Width	0.5	0.2	n/a	0.9	0.01	0.01
Height	0.2	0.5	0.9	n/a	0.01	0.01
Label	0.01	0.01	0.01	0.01	n/a	0.01
Color	0.01	0.01	0.01	0.01	0.01	n/a

priorityWrt	X	Y	Width	Height	Label	Color
X	n/a	0.5	0.8	0.8	0.5	0.5
Y	0.5	n/a	0.8	0.8	0.5	0.5
Width	0.2	0.8	n/a	0.5	0.5	0.5
Height	0.2	0.2	0.5	n/a	0.5	0.5
Label	0.5	0.5	0.5	0.5	n/a	0.5
Color	0.5	0.5	0.5	0.5	0.5	n/a

Table 6.5: All weightings for the Visual Representation Ontology for 2D graphs.

Discovery's Projection Inspector therefore provides a good source of *executable* and *pragmatic* semantics covering different visual representations. The 2D Graph VRO is also able to capture the semantics used by 2D Graphs in other visualization toolkits such as Prefuse[HCL05].

The VRO in Figure 6.5 and table 6.5 shows the 2D graph concepts (visual artefacts) which have been mapped to the concepts (data entities) in the Music Charts DO from Figure 6.4. The "Text Label" and "Shape Colour" concepts have no visible relationships in the diagram since they are all of low strength. Note that the *isMandatory* attribute in the VRO has no semantic equivalence to any attribute in the DO. However, it is used as a control feature to ensure that visualizations are valid through having all mandatory DO concepts (i.e., visual artefacts) mapped.



Base settings	
Graph	
+ sort	
− layout	Ellipses
+ x	LAST_WEEK
+ y	THIS_WEEK
+ width	0.1
+ height	0.2
+ x legend	☑
+ y legend	☑
− records	
+ label	ARTIST
− decoration	Base
+ value	1.0
method	sequential
+ transparency	Full
color	🟩
paint	☑
+ frame	□

Figure 6.6: ILOG Discovery's Projection Inspector.

Where possible, we use terminology from the Information visualization reference model [CMS99] and Data State Model [Chi02] as the terminology is independent of the visual representation and particular visualization toolkit used.

Note that the certainty values for the VRO are read in the same way as for the DO (see section 6.4). There are some differences in the names of the relationships and attributes. These differences and the *semantic equivalences* are shown in table 6.3.

6.6 SemViz: Semantic Bridging Ontology (SBO)

The purpose of the SBO is to capture and store the available expert knowledge about how various subject domains can be usefully visualized by different visual representations. This allows the complexity of the number of mapping permutations to be reduced. It also allows the accuracy of the scoring algorithm to be increased (see Section 6.7). The SBO is made up of Semantic Bridge concepts (or "semantic bridges"). Each semantic bridge records a single mapping between a DO concept (source data entity) and a VRO concept (target visual artefact), together with its appropriateness value. In this way, the SBO is a fully-connected graph of all possible permutations between the DO(s) and the VRO(s) in the system. By default, the appropriateness given to each semantic bridge is 100. However, this value can be increased or decreased to reflect specific expert knowledge. The SBO shown in Figure 6.7 highlights the semantic bridges which have non-default appropriateness values for mappings between the Music Domain Ontology and the 2D graph Visual Representation Ontology.

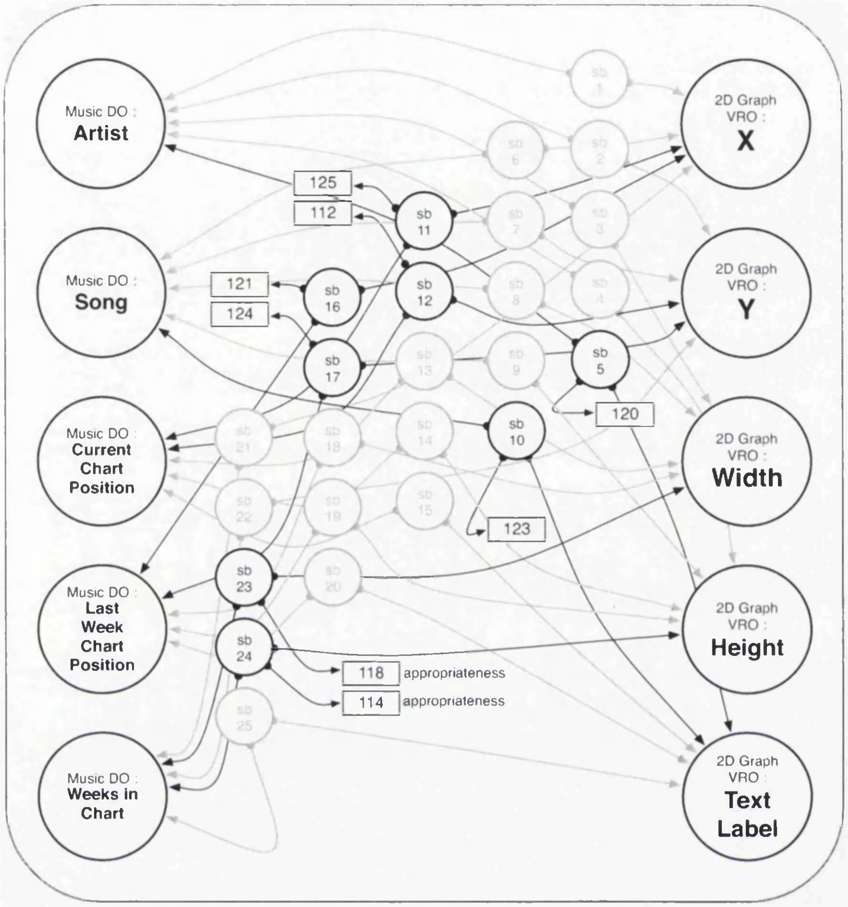


Figure 6.7: The Semantic Bridging Ontology containing the domain and visualization knowledge for mappings between the Music DO and the 2D Graph VRO.

As an example, semantic bridge number 10 (*sb10*) represents the mapping between the source data entity *Song* and the target representation artefact *Text Label*. This semantic bridge has an appropriateness value of 123. This indicates that there is a strong mapping affinity between these two concepts. The benefits of this are two-fold. Firstly, the complexity of the SemViz algorithm is reduced because we can discount mapping permutations where the source data entity *Song* is not mapped to the target representation artefact *Text Label*. We can do this for all semantic bridges with an appropriateness value over a certain threshold. Secondly, we can use the appropriateness value as a factor in the SemViz scoring algorithm. This has the effect of increasing the accuracy of the scoring by taking into account the stored expert knowledge in the SBO. The SemViz scoring algorithm which takes the appropriateness value into consideration is detailed in section 6.7 as the second version of $totalw_{\theta}$.

6.7 SemViz: Ontology Mapping Algorithm

The algorithm we employ to score the mapping permutations from the DO to VRO is loosely based on a version of the OMEN (Ontology Mapping ENhancer) algorithm described in [MNJ05]. OMEN uses a set of meta-rules that capture the influence of the ontology structure and the semantics of ontology relations to match nodes that are neighbours of already matched nodes in two ontologies. The approach of using meta-rules and semantics of concept relations has inspired the use of semantic equivalence (see table 6.3) and the comparison of relationship values (see section 6.3.1). Instead of OMEN's Bayesian network (which cannot easily be defined by experts), we use the SBO to manage the complexity and scalability of the mapping process. We also use the principle of Certainty Factors as described in [Cha08] in order to weight the relationships between concepts in the DO and VRO.

It is possible to consider all permutations between concepts from DO to VRO. However, this leads to an algorithm with a factorial computational complexity. Therefore, for non-trivial examples, the number of permutations to check quickly becomes unmanageable. To reduce the number of permutations, we use the SBO to ensure that only a subset of the permutations will be considered - those with semantic bridge appropriateness values over a pre-determined threshold. This expert knowledge can also be used by the scoring algorithm.

With respect to the base example in Figure 6.8, let θ be the mapping from DO to VRO, so:

$$V = \theta(D)$$
$$V' = \theta(D')$$

For a given D in DO, if we wish to find w_D (the weighting of the concept pair mapping in this permutation in θ), we known that D to D' has a relationship of type $q_{D'}$. We also known that V to V' has a relationship of type $q'_{V'}$. If $q_{D'}$ and $q'_{V'}$ have semantic equivalence, $q_{D'} \sim q'_{V'}$ (see table 6.3) then we can compare the certainty strength values: $s_{D'}$ and $t_{V'}$. The closer these certainty strength values are to each other, the higher the probability of them being equivalent. In order to get w_D, we apply a "fitness function" which takes the two strength values as parameters (s and t). In the example visualizations in this chapter, we use the first fitness function (f_1).

The overall score given to the whole permutation, $totalw_\theta$ (indicating the calculated cognitive value of the visualization) is the sum of all concept pair weight values. This is formalised as:

$$w_D = \sum_{D' \neq D} f_1(s_{D'}, t_{V'})$$

$$totalw_\theta = \sum_{D \in DO} w_D$$

where

$$f_1(s,t) := 1 - |s - t|$$

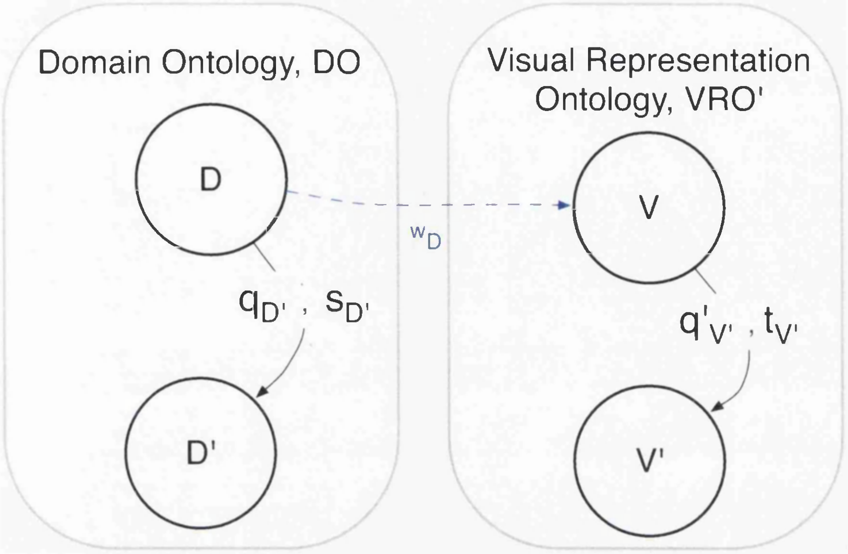


Figure 6.8: The DO and VRO of a base example.

Sub-table	Attribute or Relationship
isQuantative_isQuantative	A
isQualitative_isQualitative	A
isPrimaryKey_isInformational	A
isPresent_isMandatory	A (Control Attribute)
has_contains	R
complements_complements	R
priorityWrt_priorityWrt	R

Table 6.6: The sub-tables used in the SemViz algorithm (either Attribute or Relationship).

Other fitness functions are possible, such as f_2, which takes into account the size of the values of s and t.

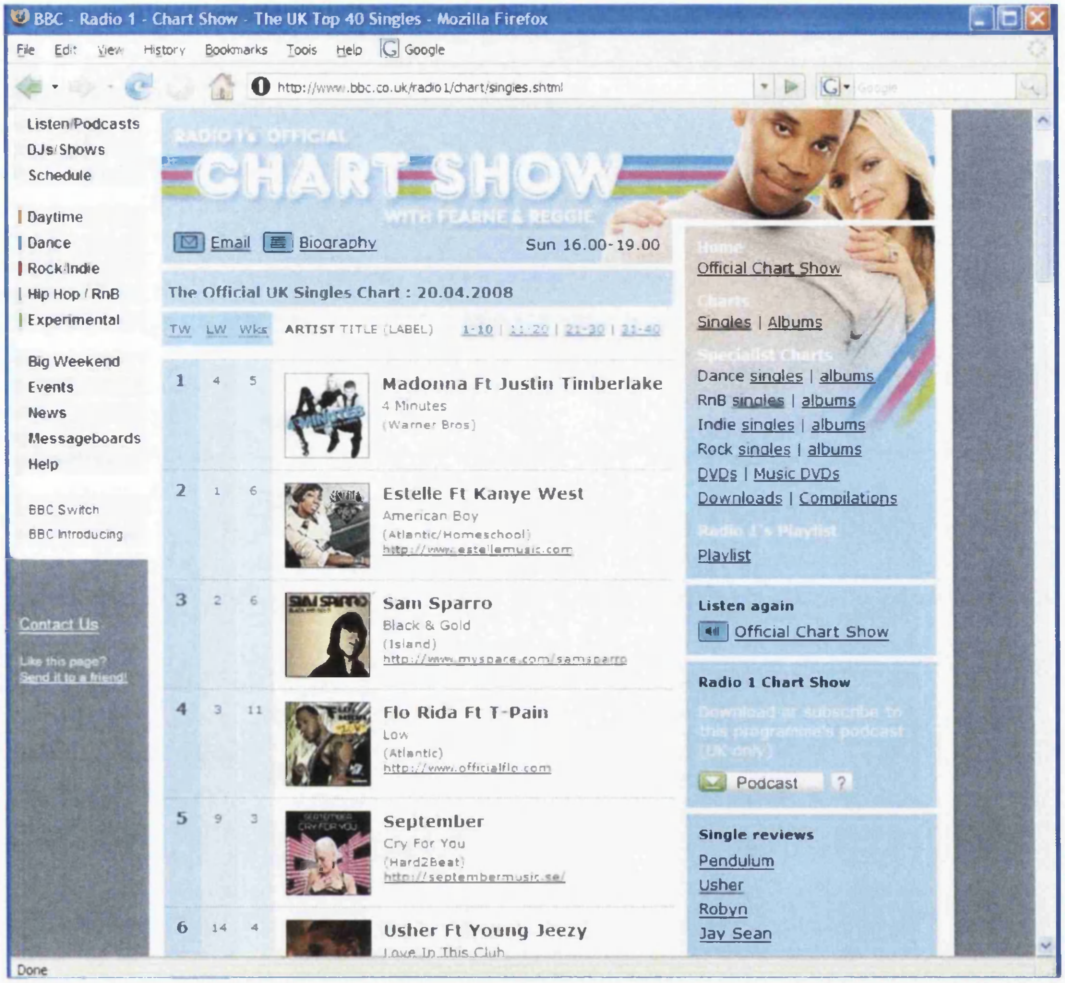
$$f_2(s,t) := (1 - |s - t|) \cdot \frac{s+t}{2}$$

An alternative version of $totalw_\theta$ takes into account the appropriateness value ($a_{DV'}$) stored in the SBO.

$$totalw_\theta = \sum_{D \in DO} w_D \cdot \frac{a_{D\theta(D)}}{100}$$

The approach of using a SBO allows us to reduce the permutation search space while utilising existing domain and visualization knowledge.

For the first example, we will use the scoring of the BBC Top 40 music chart dataset. The original web page is shown in figure 6.9. In figure 6.10 we can see each of the 7 sub-tables which have been used to calculate the total score for the highest ranking visualization. The 7 sub-tables are shown in table 6.6.



The Official UK Singles Chart : 20.04.2008

TW	LW	Wks	ARTIST TITLE (LABEL)
			1-10 | 11-20 | 21-30 | 31-40
1	4	5	Madonna Ft Justin Timberlake 4 Minutes (Warner Bros)
2	1	6	Estelle Ft Kanye West American Boy (Atlantic/Homeschool) http://www.estellemusic.com
3	2	6	Sam Sparro Black & Gold (Island) http://www.myspace.com/samsparro
4	3	11	Flo Rida Ft T-Pain Low (Atlantic) http://www.officialflo.com
5	9	3	September Cry For You (Hard2Beat) http://septembermusic.se/
6	14	4	Usher Ft Young Jeezy Love In This Club

Figure 6.9: The BBC Top 40 music chart web page source data.

There are two types of sub-table: attribute sub-tables; and relationship sub-tables. Attribute sub-tables are simpler, so we will consider them first.

Each *attribute sub-table* is split into 3 columns. The first column represents the values for the source entity, the second column represents the values for the target entity, and the third and final column (shaded green) represents the score for that row (calculated from the source and target value using f_1 from section 6.7). The format is illustrated in the first attribute sub-table (*isQuantative_isQuantative*) where the relationship between columns is highlighted with orange and yellow. The final row in the third column (shaded blue) is the sum of all rows above it and represents the total score for that attribute sub-table.

The format of *relationship sub-tables* is similar, except that instead of one column each for the source values, target values and total values, there are 4 (we call these a column blocks). The reason for this is so that we can calculate the certainty weightings of each relationship between all the entities. This format is illustrated in the *complements_complements* relationship sub-table. The certainty weighting between the target entity (highlighted red) and all the other target entities (highlighted green) are represented as the numbers in the target column block (highlighted purple). The third column-block shows the scores (highlighted orange) for each row's certainty weightings (calculated using f_1 from section 6.7). The relationship sub-table has a final column which is the sum of the four scores to it's left. The last row in this final column (shaded blue) is the sum of all five values above it and represents the total score for this relationship sub-table.

Finally, the total score for that mapping permutation is the sum of the scores of the (seven) sub-tables which in the example is 25.42 resulting in a ranking of 1. The resulting visualizations and scores are shown in figure 6.11.

Please note that the structure of the tables described above is specific to this particular example (BBC Top 40 music chart visualized as a 2D Graph). In particular, the number of column blocks in each relationship sub-table is 4 because $|VRO| = 5$ (i.e., the number of column blocks in each relationship sub-table is $|VRO| - 1$).

6.8 SemViz: Interaction Design

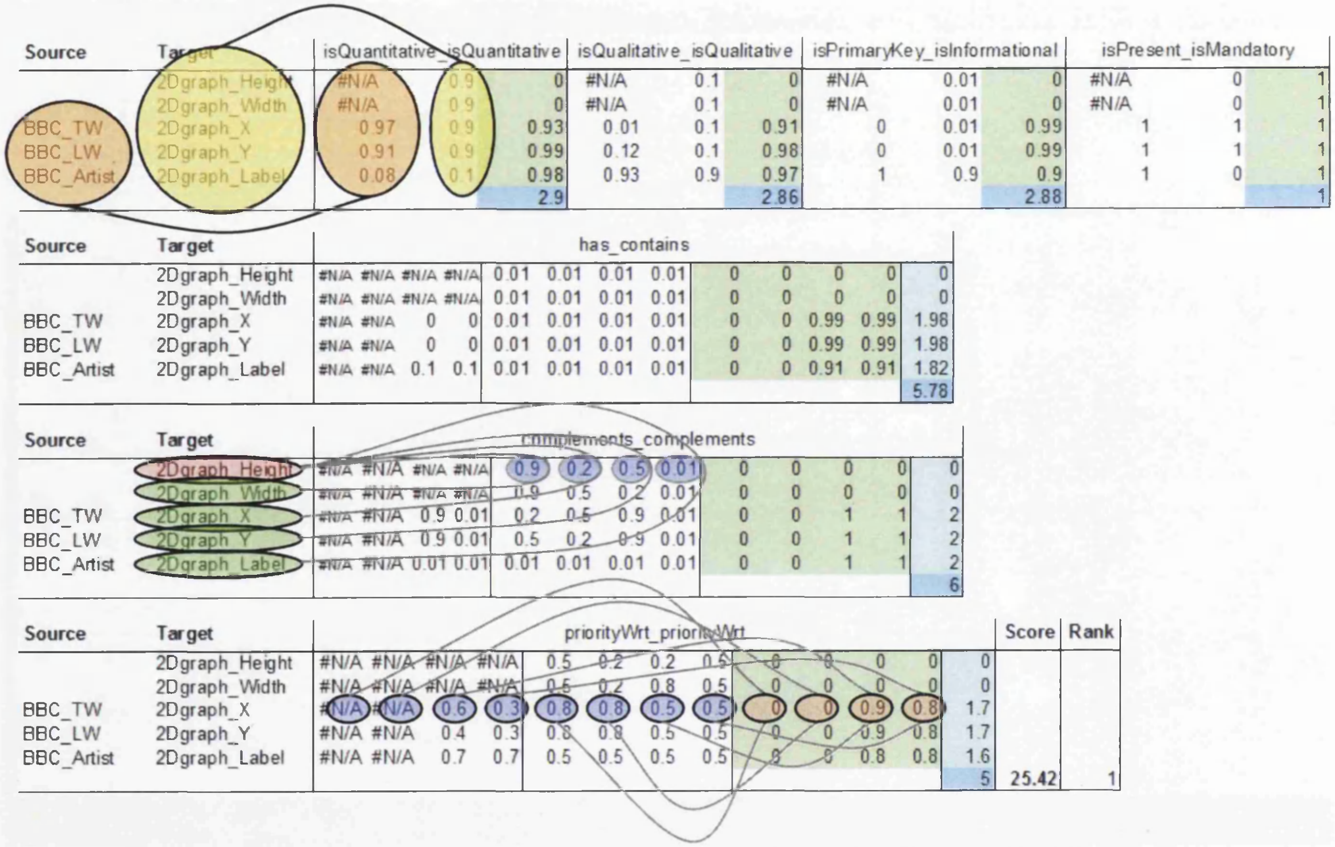
The SemViz user interaction design allows novice visualization users to view thumbnails of the visualizations generated. Each thumbnail is scored and ranked. The user can at any time select a thumbnail to see a larger view of it. This is shown in figure 6.12. Each visualization thumbnail is shown together with details of how the source data entities are mapped to the target representation artefacts. Below each thumbnail is a bar indicator showing the relative rank of each visualization.

The interaction design is based on the gallery interaction methodology [MAB$^+$97]. This presents the user with multiple visualizations for one data set. This is based on the principle of the user being able to choose which visualization is most applicable for their needs. The ontology-based pipeline in SemViz lends itself to this style of interaction. Since the result of the pipeline is a score for each possible mapping permutation, we can present the user with a manageable set of the best visualizations.

This interaction methodology is particularly useful if there is a low degree of certainty about the scores given to each visualization possibility. This uncertainty is caused by a poor mapping certainty between the source data and the domain ontology (due to the source data not having a good match with the DO, or there being limited records in the source data).

The benefits of the gallery selection methodology are:

- the user gets "something" to see, even if the certainty of its appropriateness is low.
- the visualization thumbnails which the user selects provides the system with feedback on the mapping decisions which were made. This provides the basis for a learning system based on user feedback.



Source	Target	isQuantitative_isQuantitative			isQualitative_isQualitative			isPrimaryKey_isInformational			isPresent_isMandatory		
	2Dgraph_Height	#N/A	0.9	0	#N/A	0.1	0	#N/A	0.01	0	#N/A	0	1
	2Dgraph_Width	#N/A	0.9	0	#N/A	0.1	0	#N/A	0.01	0	#N/A	0	1
BBC_TW	2Dgraph_X	0.97	0.9	0.93	0.01	0.1	0.91	0	0.01	0.99	1	1	1
BBC_LW	2Dgraph_Y	0.91	0.9	0.99	0.12	0.1	0.98	0	0.01	0.99	1	1	1
BBC_Artist	2Dgraph_Label	0.08	0.1	0.98	0.93	0.9	0.97	1	0.9	0.9	1	0	1
				2.9			2.86			2.88			1

Source	Target	has_contains												
	2Dgraph_Height	#N/A	#N/A	#N/A	#N/A	0.01	0.01	0.01	0.01	0	0	0	0	0
	2Dgraph_Width	#N/A	#N/A	#N/A	#N/A	0.01	0.01	0.01	0.01	0	0	0	0	0
BBC_TW	2Dgraph_X	#N/A	#N/A	0	0	0.01	0.01	0.01	0.01	0	0	0.99	0.99	1.98
BBC_LW	2Dgraph_Y	#N/A	#N/A	0	0	0.01	0.01	0.01	0.01	0	0	0.99	0.99	1.98
BBC_Artist	2Dgraph_Label	#N/A	#N/A	0.1	0.1	0.01	0.01	0.01	0.01	0	0	0.91	0.91	1.82
														5.78

Source	Target	complements_complements												
	2Dgraph_Height	#N/A	#N/A	#N/A	#N/A	0.9	0.2	0.5	0.01	0	0	0	0	0
	2Dgraph_Width	#N/A	#N/A	#N/A	#N/A	0.9	0.5	0.2	0.01	0	0	0	0	0
BBC_TW	2Dgraph_X	#N/A	#N/A	0.9	0.01	0.2	0.5	0.9	0.01	0	0	1	1	2
BBC_LW	2Dgraph_Y	#N/A	#N/A	0.9	0.01	0.5	0.2	0.9	0.01	0	0	1	1	2
BBC_Artist	2Dgraph_Label	#N/A	#N/A	0.01	0.01	0.01	0.01	0.01	0.01	0	0	1	1	2
														6

Source	Target	priorityWrt_priorityWrt													Score	Rank
	2Dgraph_Height	#N/A	#N/A	#N/A	#N/A	0.5	0.2	0.2	0.5	0	0	0	0	0		
	2Dgraph_Width	#N/A	#N/A	#N/A	#N/A	0.5	0.2	0.8	0.5	0	0	0	0	0		
BBC_TW	2Dgraph_X	#N/A	#N/A	0.6	0.3	0.8	0.8	0.5	0.5	0	0	0.9	0.8	1.7		
BBC_LW	2Dgraph_Y	#N/A	#N/A	0.4	0.3	0.8	0.8	0.5	0.5	0	0	0.9	0.8	1.7		
BBC_Artist	2Dgraph_Label	#N/A	#N/A	0.7	0.7	0.5	0.5	0.5	0.5	0	0	0.8	0.8	1.6		
														5	25.42	1

Figure 6.10: The SemViz scoring algorithm figures for the first visualization (Annotated).

6.9 SemViz: Results and Remarks

6.9.1 Instance vs. Schema-level Categorisation

In this example, we use the iTunes music chart web page as source data. The data set used is shown in figure 6.13 and consists of: Country name, Song name, Artist name

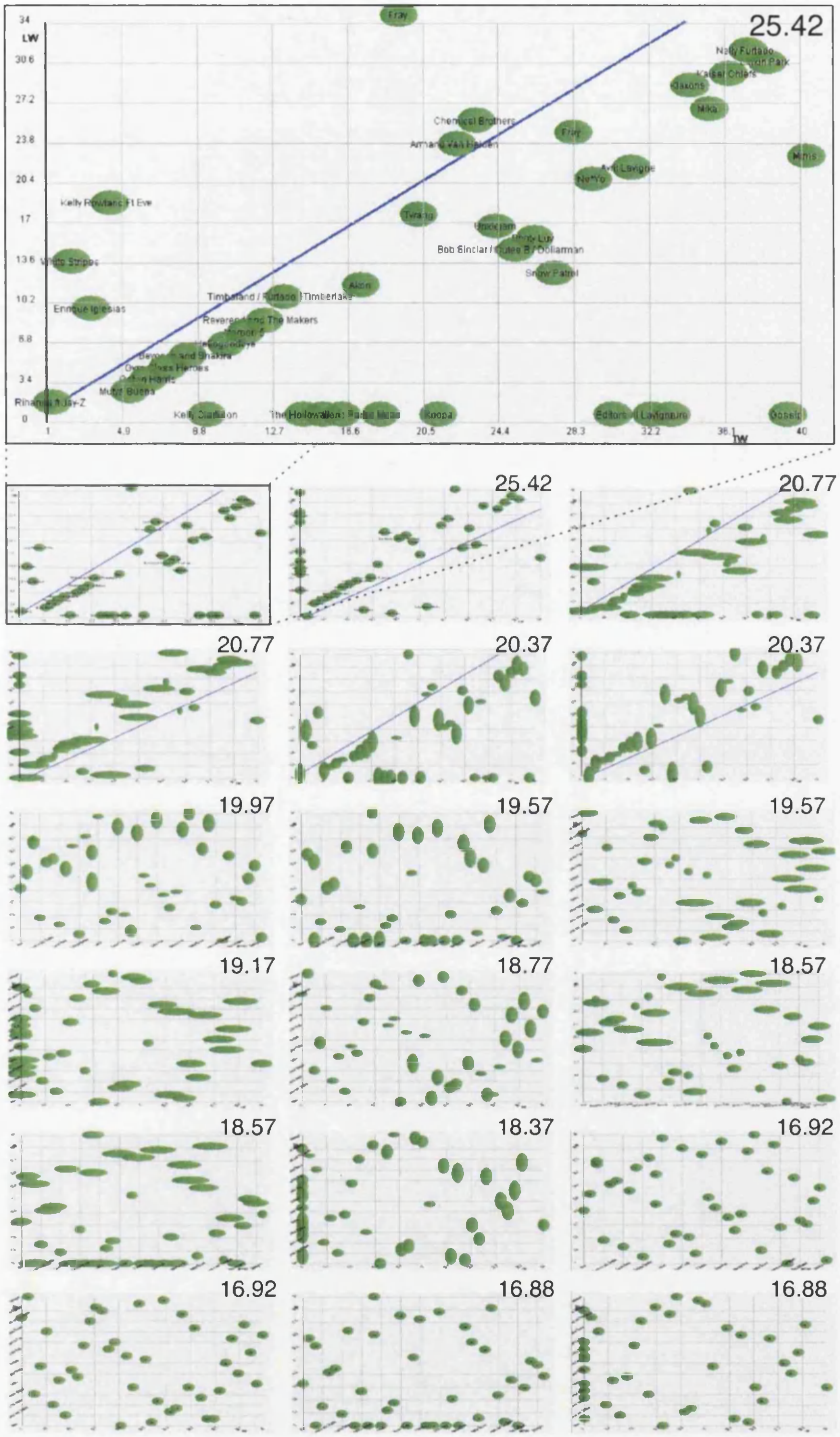
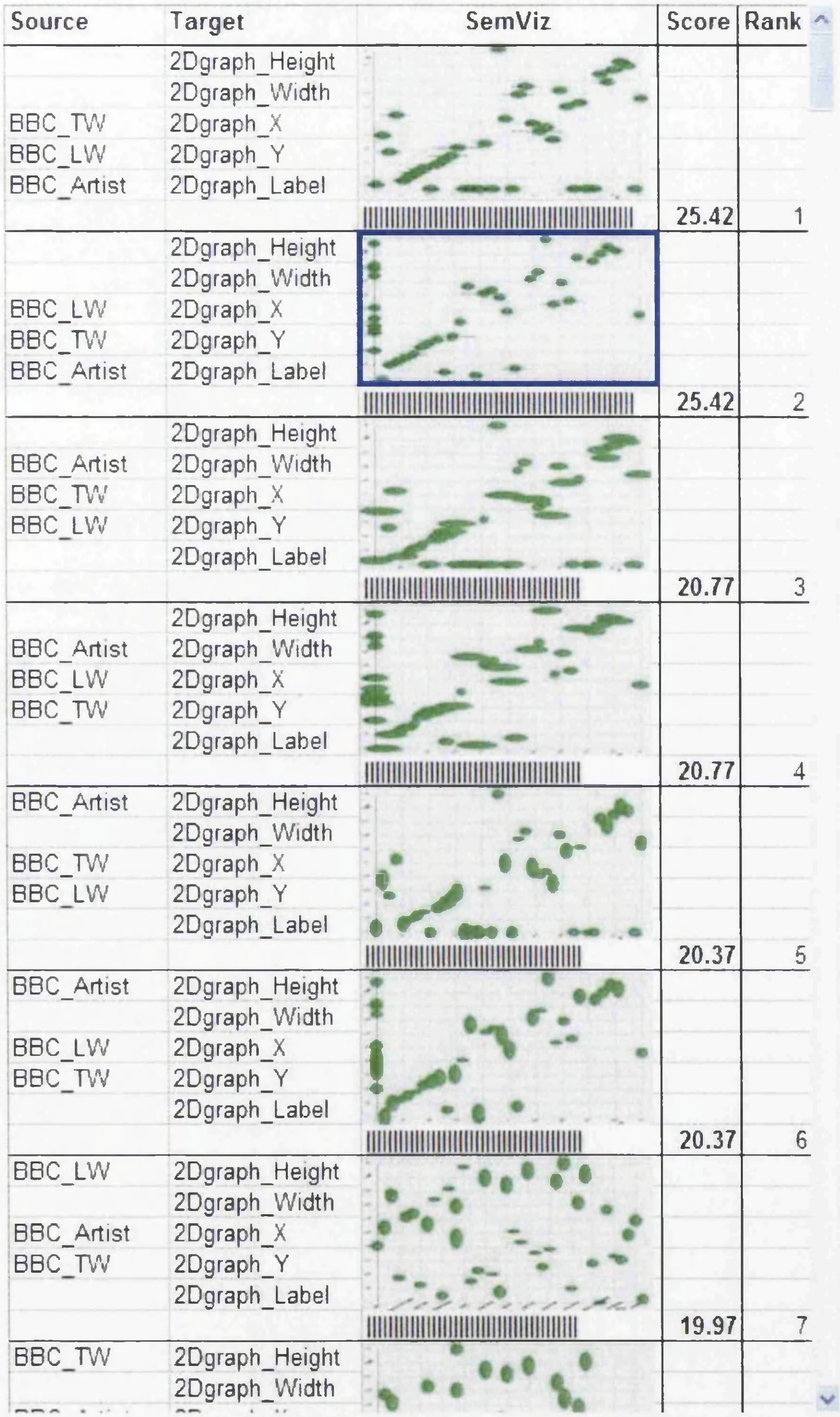


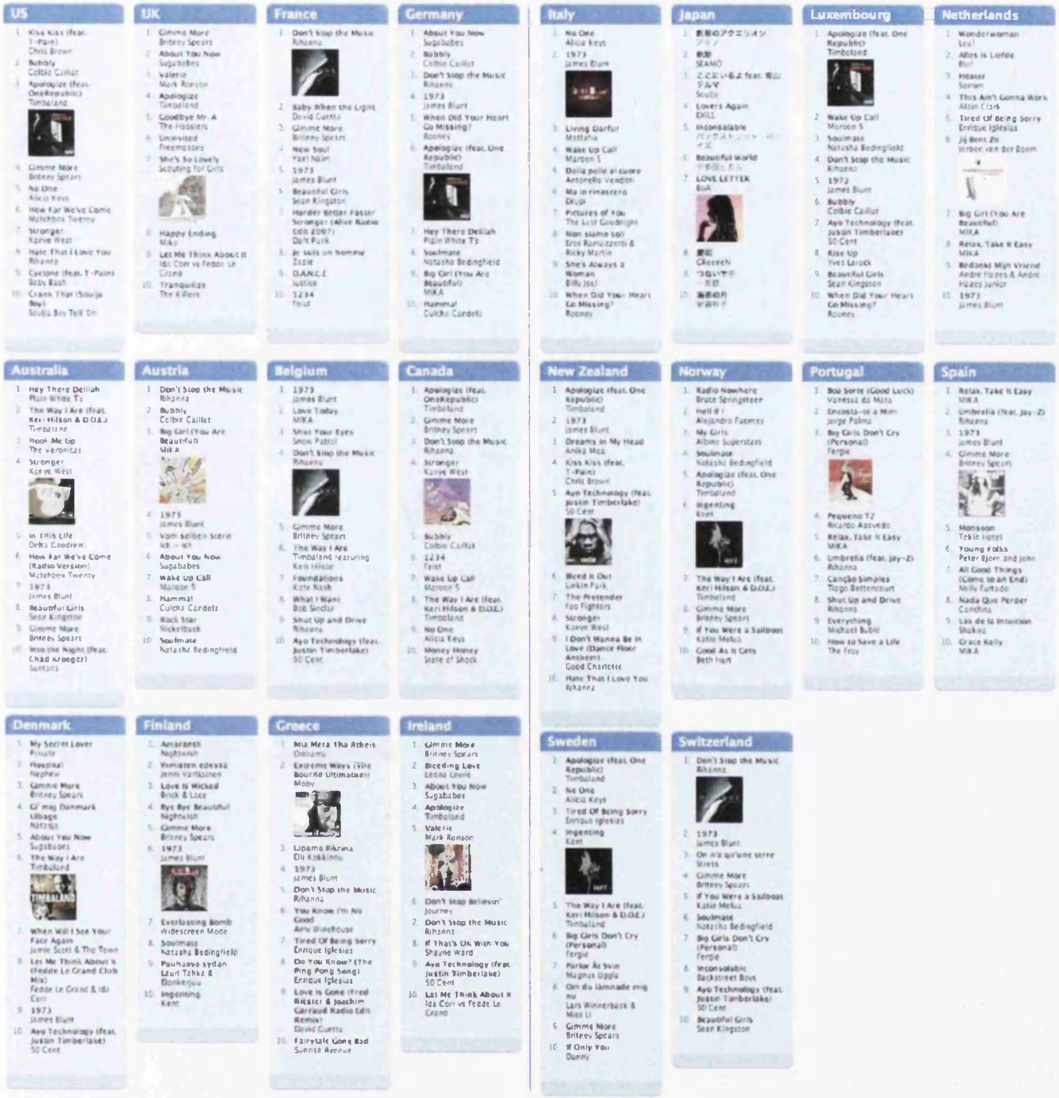
Figure 6.11: Top: The highest scoring visualization. Thumbnails: Images showing all (usable) permutations of the BBC Top 40 web page to the 2D Graph VRO in ILOG Discovery.



Source	Target	SemViz	Score	Rank
	2Dgraph_Height			
	2Dgraph_Width			
BBC_TW	2Dgraph_X			
BBC_LW	2Dgraph_Y			
BBC_Artist	2Dgraph_Label			
			25.42	1
	2Dgraph_Height			
	2Dgraph_Width			
BBC_LW	2Dgraph_X			
BBC_TW	2Dgraph_Y			
BBC_Artist	2Dgraph_Label			
			25.42	2
	2Dgraph_Height			
BBC_Artist	2Dgraph_Width			
BBC_TW	2Dgraph_X			
BBC_LW	2Dgraph_Y			
	2Dgraph_Label			
			20.77	3
	2Dgraph_Height			
BBC_Artist	2Dgraph_Width			
BBC_LW	2Dgraph_X			
BBC_TW	2Dgraph_Y			
	2Dgraph_Label			
			20.77	4
BBC_Artist	2Dgraph_Height			
	2Dgraph_Width			
BBC_TW	2Dgraph_X			
BBC_LW	2Dgraph_Y			
	2Dgraph_Label			
			20.37	5
BBC_Artist	2Dgraph_Height			
	2Dgraph_Width			
BBC_LW	2Dgraph_X			
BBC_TW	2Dgraph_Y			
	2Dgraph_Label			
			20.37	6
BBC_LW	2Dgraph_Height			
	2Dgraph_Width			
BBC_Artist	2Dgraph_X			
BBC_TW	2Dgraph_Y			
	2Dgraph_Label			
			19.97	7
BBC_TW	2Dgraph_Height			
	2Dgraph_Width			

Figure 6.12: The SemViz user interaction design showing the scoring and ranking of the BBC Top 40 to 2D graph visualizations.

and Chart position of songs in top 10 music charts in 22 different countries. This dataset differs from the first example dataset (BBC Top 40) in that the data fields primarily have



US

1. Kiss Kiss (feat. T-Pain)
Chris Brown
2. Bubbly
Colbie Caillat
3. Apologize (feat. OneRepublic)
Timbaland
4. Gimme More
Britney Spears
5. No One
Alicia Keys
6. How Far We've Come
Matchbox Twenty
7. Stronger
Kanye West
8. Hate That I Love You
Rihanna
9. Cyclone (feat. T-Pain)
Baby Bash
10. Crank That (Soulja Boy)
Soulja Boy Tell 'Em

UK

1. Gimme More
Britney Spears
2. About You Now
Sugababes
3. Valerie
Mark Ronson
4. Apologize
Timbaland
5. Goodbye Mr. A
The Hoosiers
6. Uninvited
Freemasons
7. She's So Lovely
Scouting for Girls
8. Happy Ending
Mika
9. Let Me Think About It
Ida Corr vs Fedde Le Grand
10. Tranquilize
The Killers

France

1. Don't Stop the Music
Rihanna
2. Baby When the Light
David Guetta
3. Gimme More
Britney Spears
4. New Soul
Yael Naim
5. 1973
James Blunt
6. Beautiful Girls
Sean Kingston
7. Harder Better Faster Stronger (Alive Radio Edit 2007)
Daft Punk
8. Je suis un homme
Zazie
9. D.A.N.C.E
Justice
10. 1234
Feist

Germany

1. About You Now
Sugababes
2. Bubbly
Colbie Caillat
3. Don't Stop the Music
Rihanna
4. 1973
James Blunt
5. When Did Your Heart Go Missing?
Rooney
6. Apologize (feat. One Republic)
Timbaland
7. Hey There Delilah
Plain White T's
8. Soulmate
Natasha Bedingfield
9. Big Girl (You Are Beautiful)
MIKA
10. Hammal
Culcha Candela

Italy

1. No One
Alicia Keys
2. 1973
James Blunt
3. Living Darfur
Mattafix
4. Wake Up Call
Maroon 5
5. Dalla pelle al cuore
Antonello Venditti
6. Ma in rinascerò
Drupi
7. Pictures of You
The Last Goodnight
8. Non siamo soli
Eros Ramazzotti & Ricky Martin
9. She's Always a Woman
Billy Joel
10. When Did Your Heart Go Missing?
Rooney

Japan

1. 創聖のアクエリオン
アキノ
2. 軌跡
SEAMO
3. ここにいるよ feat. 青山テルマ
SoulJa
4. Lovers Again
EXILE
5. inconsolable
バックストリート・ボーイズ
6. Beautiful World
宇多田ヒカル
7. LOVE LETTER
BoA
8. 愛唄
GReeeeN
9. つないだ手
[illegible]
10. [illegible]
[illegible]

Luxembourg

1. Apologize (feat. One Republic)
Timbaland
2. Wake Up Call
Maroon 5
3. Soulmate
Natasha Bedingfield
4. Don't Stop the Music
Rihanna
5. 1973
James Blunt
6. Bubbly
Colbie Caillat
7. Ayo Technology (feat. Justin Timberlake)
50 Cent
8. Rise Up
Yves Larock
9. Beautiful Girls
Sean Kingston
10. When Did Your Heart Go Missing?
Rooney

Netherlands

1. Wonderwoman
Lea!
2. Alles is Liefde
Blof
3. Heater
Samim
4. This Ain't Gonna Work
Alain Clark
5. Tired Of Being Sorry
Enrique Iglesias
6. Jij Bent Zo
Jeroen van der Boom
7. Big Girl (You Are Beautiful)
MIKA
8. Relax, Take It Easy
MIKA
9. Bedankt Mijn Vriend
André Hazes & André Hazes Junior
10. 1973
James Blunt

Australia

1. Hey There Delilah
Plain White T's
2. The Way I Are (feat. Keri Hilson & D.O.E.)
Timbaland
3. Hook Me Up
The Veronicas
4. Stronger
Kanye West
5. In This Life
Delta Goodrem
6. How Far We've Come (Radio Version)
Matchbox Twenty
7. 1973
James Blunt
8. Beautiful Girls
Sean Kingston
9. Gimme More
Britney Spears
10. Into the Night (feat. Chad Kroeger)
Santana

Austria

1. Don't Stop the Music
Rihanna
2. Bubbly
Colbie Caillat
3. Big Girl (You Are Beautiful)
MIKA
4. 1973
James Blunt
5. Vom selben Stern
Ich + Ich
6. About You Now
Sugababes
7. Wake Up Call
Maroon 5
8. Hammal
Culcha Candela
9. Rock Star
Nickelback
10. Soulmate
Natasha Bedingfield

Belgium

1. 1973
James Blunt
2. Love Today
MIKA
3. Shut Your Eyes
Snow Patrol
4. Don't Stop the Music
Rihanna
5. Gimme More
Britney Spears
6. The Way I Are
Timbaland featuring Keri Hilson
7. Foundations
Kate Nash
8. What I Want
Bob Sinclar
9. Shut Up and Drive
Rihanna
10. Ayo Technology (feat. Justin Timberlake)
50 Cent

Canada

1. Apologize (feat. OneRepublic)
Timbaland
2. Gimme More
Britney Spears
3. Don't Stop the Music
Rihanna
4. Stronger
Kanye West
5. Bubbly
Colbie Caillat
6. 1234
Feist
7. Wake Up Call
Maroon 5
8. The Way I Are (feat. Keri Hilson & D.O.E.)
Timbaland
9. No One
Alicia Keys
10. Money Honey
State of Shock

New Zealand

1. Apologize (feat. One Republic)
Timbaland
2. 1973
James Blunt
3. Dreams in My Head
Anika Moa
4. Kiss Kiss (feat. T-Pain)
Chris Brown
5. Ayo Technology (feat. Justin Timberlake)
50 Cent
6. Bleed It Out
Linkin Park
7. The Pretender
Foo Fighters
8. Stronger
Kanye West
9. I Don't Wanna Be in Love (Dance Floor Anthem)
Good Charlotte
10. Hate That I Love You
Rihanna

Norway

1. Radio Nowhere
Bruce Springsteen
2. Hell If I
Alejandro Fuentes
3. My Girls
Albino Superstars
4. Soulmate
Natasha Bedingfield
5. Apologize (feat. One Republic)
Timbaland
6. Ingenting
Kent
7. The Way I Are (feat. Keri Hilson & D.O.E.)
Timbaland
8. Gimme More
Britney Spears
9. If You Were a Sailboat
Katie Melua
10. Good As It Gets
Beth Hart

Portugal

1. Boa Sorte (Good Luck)
Vanessa da Mata
2. Encosta-te a Mim
Jorge Palma
3. Big Girls Don't Cry (Personal)
Fergie
4. Pequeno T2
Ricardo Azevedo
5. Relax, Take It Easy
MIKA
6. Umbrella (feat. Jay-Z)
Rihanna
7. Canção Simples
Tiago Bettencourt
8. Shut Up and Drive
Rihanna
9. Everything
Michael Bublé
10. How to Save a Life
The Fray

Spain

1. Relax, Take It Easy
MIKA
2. Umbrella (feat. Jay-Z)
Rihanna
3. 1973
James Blunt
4. Gimme More
Britney Spears
5. Monsoon
Tokio Hotel
6. Young Folks
Peter Bjorn and John
7. All Good Things (Come to an End)
Nelly Furtado
8. Nada Que Perder
Conchita
9. Las de la Intuición
Shakira
10. Grace Kelly
MIKA

Denmark

1. My Secret Lover
Private
2. Hospital
Nephew
3. Gimme More
Britney Spears
4. Gi' mig Danmark tilbage
Natasja
5. About You Now
Sugababes
6. The Way I Are
Timbaland
7. When Will I See Your Face Again
Jamie Scott & The Town
8. Let Me Think About It (Fedde Le Grand Club Mix)
Fedde Le Grand & Ida Corr
9. 1973
James Blunt
10. Ayo Technology (feat. Justin Timberlake)
50 Cent

Finland

1. Amaranth
Nightwish
2. Ihmisten edessä
Jenni Vartiainen
3. Love Is Wicked
Brick & Lace
4. Bye Bye Beautiful
Nightwish
5. Gimme More
Britney Spears
6. 1973
James Blunt
7. Everlasting Bomb
Widescreen Mode
8. Soulmate
Natasha Bedingfield
9. Pauhaava sydän
Lauri Tähkä & Elonkerjuu
10. Ingenting
Kent

Greece

1. Mia Mera Tha Atheis
Onirama
2. Extreme Ways (The Bourne Ultimatum)
Moby
3. Lipame Mikrena
Eli Kokkinou
4. 1973
James Blunt
5. Don't Stop the Music
Rihanna
6. You Know I'm No Good
Amy Winehouse
7. Tired Of Being Sorry
Enrique Iglesias
8. Do You Know? (The Ping Pong Song)
Enrique Iglesias
9. Love Is Gone (Fred Riester & Joachim Garraud Radio Edit Remix)
David Guetta
10. Fairytale Gone Bad
Sunrise Avenue

Ireland

1. Gimme More
Britney Spears
2. Bleeding Love
Leona Lewis
3. About You Now
Sugababes
4. Apologize
Timbaland
5. Valerie
Mark Ronson
6. Don't Stop Believin'
Journey
7. Don't Stop the Music
Rihanna
8. If That's OK With You
Shayne Ward
9. Ayo Technology (feat. Justin Timberlake)
50 Cent
10. Let Me Think About It
Ida Corr vs Fedde Le Grand

Sweden

1. Apologize (feat. One Republic)
Timbaland
2. No One
Alicia Keys
3. Tired Of Being Sorry
Enrique Iglesias
4. Ingenting
Kent
5. The Way I Are (feat. Keri Hilson & D.O.E.)
Timbaland
6. Big Girls Don't Cry (Personal)
Fergie
7. Pärlor Åt Svin
Magnus Uggla
8. Om du lämnade mig nu
Lars Winnerbäck & Miss Li
9. Gimme More
Britney Spears
10. If Only You
Danny

Switzerland

1. Don't Stop the Music
Rihanna
2. 1973
James Blunt
3. On n'a qu'une terre
Stress
4. Gimme More
Britney Spears
5. If You Were a Sailboat
Katie Melua
6. Soulmate
Natasha Bedingfield
7. Big Girls Don't Cry (Personal)
Fergie
8. Inconsolable
Backstreet Boys
9. Ayo Technology (feat. Justin Timberlake)
50 Cent
10. Beautiful Girls
Sean Kingston

Figure 6.13: The iTunes music chart web page source data.

a hierarchical relationship, rather than being a mixture of quantitative numeric values and qualitative textual values as in the BBC Top 40 dataset.

There are two methods of deducing the semantics of the the web page source data:

1. Schema-level Categorisation. The semantics of each concept in the Domain Ontology are pre-defined. We use these semantics to calculate an effective mapping between the DO and VRO.
2. Instance-level Categorisation. An analysis of the actual values of the source data provides semantics. This analysis can optionally be performed by the Data Analysis module (see stage 2 in Figure 6.2) during the first pass through the visualization pipeline.

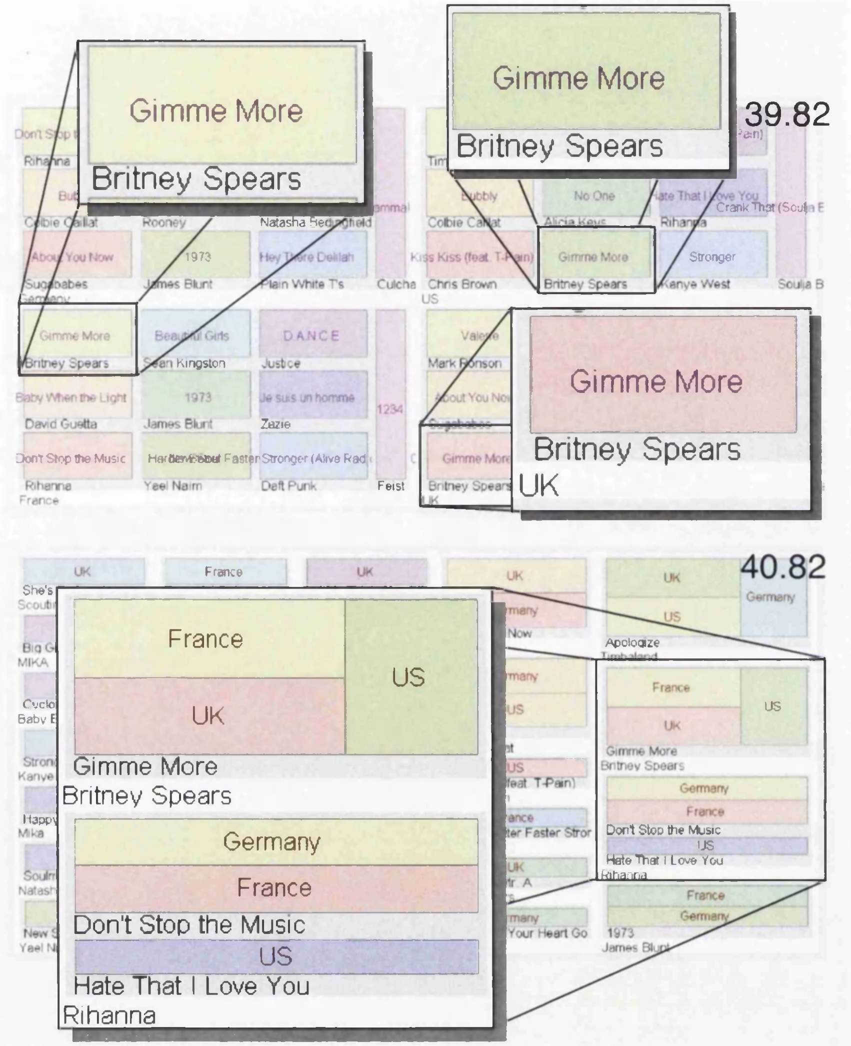


Figure 6.14: TreeMap visualization of the iTunes chart data set (4 countries only). Top: Schema-based semantics deduced from the Music Chart DO. Bottom: Instance-based semantics derived from the source data by the Data Analysis module.

In Figure 6.14 we see the effects of the two methods on the visualization of the iTunes music chart using a TreeMap (4 countries only). The top visualization uses the DO as defined (schema level). This shows a Country → Artist → Song hierarchy. The second uses an instance level analysis to augment and override the DO. As such, we have a Artist → Song → Country hierarchy. This is because the Data Analyser deduces that a Song "has" a Country, rather than a Country "has" an Artist. For the iTunes Store music chart, the second method produces a cognitively more valuable visualization compared to the first method which doesn't provide much more insight over the original web page's table of data. The top visualization is based on semantics deduced from the Music Chart Domain Ontology (schema-based). The bottom visualization uses semantics deduced from the source data (instance-based).

In this case, instance-based categorisation produces a cognitively more useful visualization since we can deduce 2 extra pieces of information more easily:

- the artists who have multiple songs in the charts.
- the songs which are in the charts in more than one country.

The question arises around which method should be used. If the source data set is limited (in terms of number of records), then very little extra semantic information can be gained from Instance-level categorisation via the data analysis module. In this case, we must utilise the semantics held within the Domain Ontology. However, a large dataset may contain a greater richness of semantics than exists in the Domain Ontology.

A TreeMap visualization for all 22 countries in the iTunes music chart is shown in figure 6.15.

6.9.2 Annotation - Comprehending Automatically Created Visualizations

At the top of figure 6.16, we show the highest scoring visualization for the BBC Top 40 webpage. Nearest to the origin, we can clearly see a cluster of shapes just below the X=Y line, representing those songs which have fallen least since the previous week. Shapes along the X-axis represent new song entries since they have no value (zero) for "Last Week".

Notice that the six visualizations with the highest scores have a diagonal line ($X = Y$) overlaid on the visualization. SemViz has automatically instructed ILOG Discovery to draw this line to assist the end-user with observing trends. A rule exists in the system which states that when a mapping permutation's concept pairs pertaining to the the X coordinate and the Y coordinate have a *complements* value greater than a certain threshold, then there is a benefit in drawing the user's attention to the placement of shapes relative to the $X = Y$ line. Therefore, for this particular permutation, an assistance line is drawn. The values related to this process are highlighted in figure 6.16. We only show the *complements_complements* sub-table here since it is the only one pertinent to this example. Each visualization permutation has two numbers highlighted (orange in the first six visualizations, yellow in the seventh and final). These numbers represent the degree to which the source entities mapped to X and Y have a similar *complements* value. The numbers range from 0.0 to 1.0. In the case of the top 6 visualizations, these values are 1, therefore indicating the maximum degree of complementary relationship. Each of the top 6 visualizations therefore has a $X = Y$ annotation line drawn on it. This situation has arisen because the top 6 visualizations all have the following mappings:

- BBC_TW mapped to 2Dgraph_X and BBC_LW mapped to 2Dgraph_Y or,
- BBC_LW mapped to 2Dgraph_X and BBC_TW mapped to 2Dgraph_Y.

The seventh (final) visualization has:

- BBC_Artist mapped to 2Dgraph_X and BBC_TW mapped to 2Dgraph_Y.

This results in the values highlighted in yellow (0.11) in figure 6.16 which represent a low degree of complementary relationship. These values are not high enough to be above the

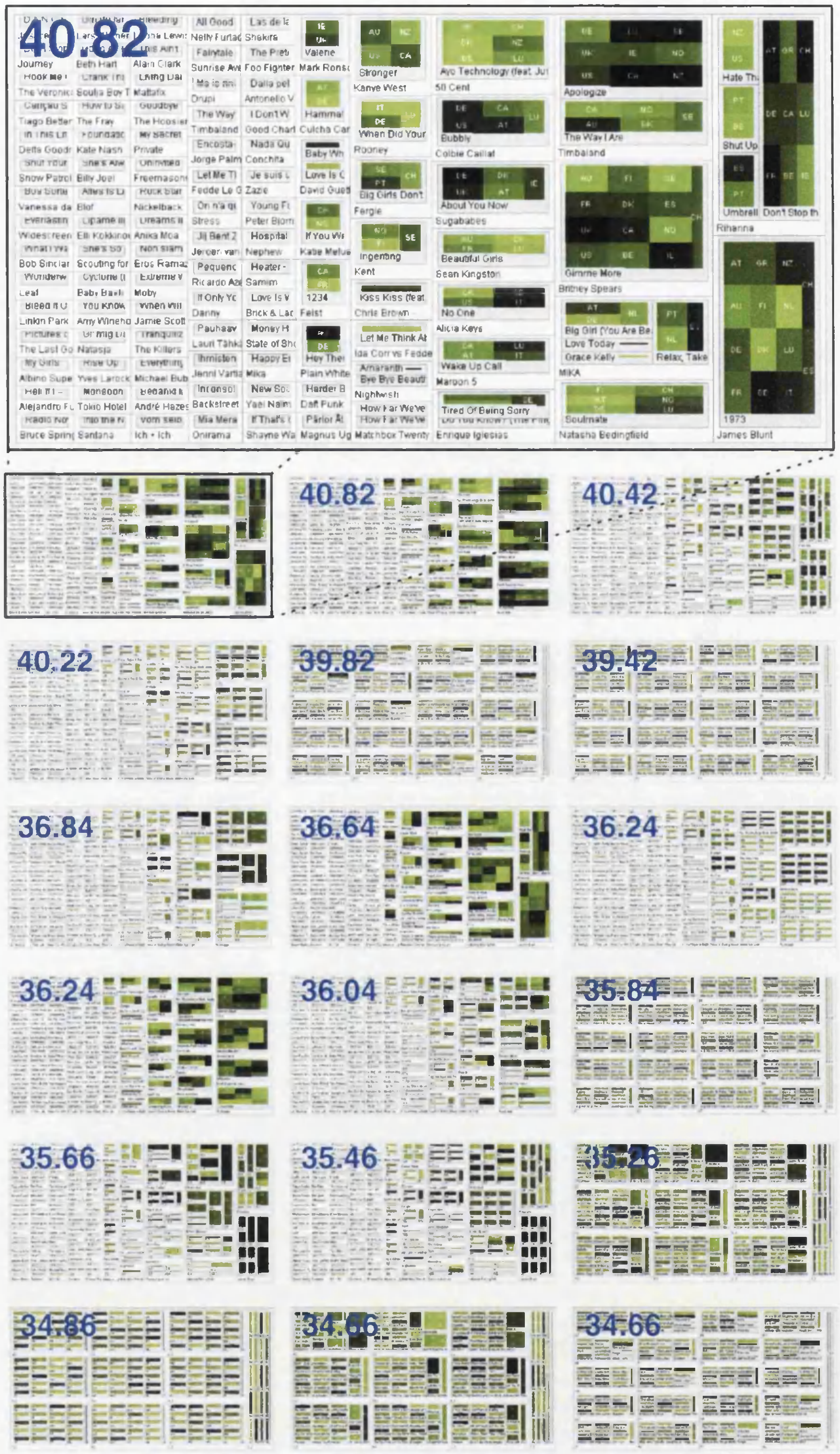
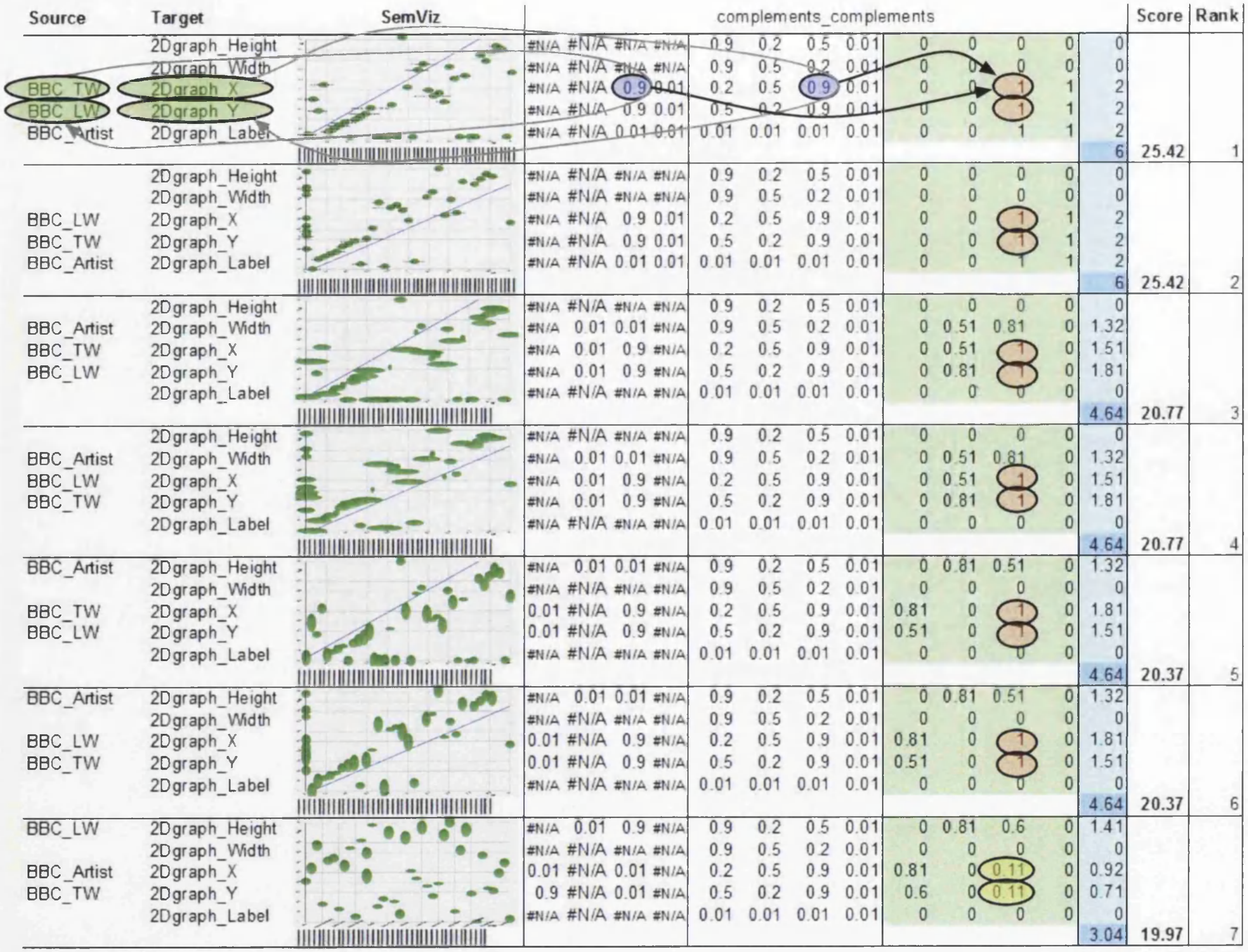


Figure 6.15: TreeMap visualization of the iTunes chart data set (all 22 countries).

threshold required to annotate the visualization with an $X = Y$ line. This can be seen in the visualization.



Source	Target	SemViz	complements_complements													Score	Rank
	2Dgraph_Height		#N/A	#N/A	#N/A	#N/A	0.9	0.2	0.5	0.01	0	0	0	0	0		
	2Dgraph_Width		#N/A	#N/A	#N/A	#N/A	0.9	0.5	0.2	0.01	0	0	0	0	0		
BBC_TW	2Dgraph_X		#N/A	#N/A	0.9	0.01	0.2	0.5	0.9	0.01	0	0	1	1	2		
BBC_LW	2Dgraph_Y		#N/A	#N/A	0.9	0.01	0.5	0.2	0.9	0.01	0	0	1	1	2		
BBC_Artist	2Dgraph_Label		#N/A	#N/A	0.01	0.01	0.01	0.01	0.01	0.01	0	0	1	1	2		
															6	25.42	1
	2Dgraph_Height		#N/A	#N/A	#N/A	#N/A	0.9	0.2	0.5	0.01	0	0	0	0	0		
	2Dgraph_Width		#N/A	#N/A	#N/A	#N/A	0.9	0.5	0.2	0.01	0	0	0	0	0		
BBC_LW	2Dgraph_X		#N/A	#N/A	0.9	0.01	0.2	0.5	0.9	0.01	0	0	1	1	2		
BBC_TW	2Dgraph_Y		#N/A	#N/A	0.9	0.01	0.5	0.2	0.9	0.01	0	0	1	1	2		
BBC_Artist	2Dgraph_Label		#N/A	#N/A	0.01	0.01	0.01	0.01	0.01	0.01	0	0	1	1	2		
															6	25.42	2
	2Dgraph_Height		#N/A	#N/A	#N/A	#N/A	0.9	0.2	0.5	0.01	0	0	0	0	0		
BBC_Artist	2Dgraph_Width		#N/A	0.01	0.01	#N/A	0.9	0.5	0.2	0.01	0	0.51	0.81	0	1.32		
BBC_TW	2Dgraph_X		#N/A	0.01	0.9	#N/A	0.2	0.5	0.9	0.01	0	0.51	1	0	1.51		
BBC_LW	2Dgraph_Y		#N/A	0.01	0.9	#N/A	0.5	0.2	0.9	0.01	0	0.81	1	0	1.81		
	2Dgraph_Label		#N/A	#N/A	#N/A	#N/A	0.01	0.01	0.01	0.01	0	0	0	0	0		
															4.64	20.77	3
	2Dgraph_Height		#N/A	#N/A	#N/A	#N/A	0.9	0.2	0.5	0.01	0	0	0	0	0		
BBC_Artist	2Dgraph_Width		#N/A	0.01	0.01	#N/A	0.9	0.5	0.2	0.01	0	0.51	0.81	0	1.32		
BBC_LW	2Dgraph_X		#N/A	0.01	0.9	#N/A	0.2	0.5	0.9	0.01	0	0.51	1	0	1.51		
BBC_TW	2Dgraph_Y		#N/A	0.01	0.9	#N/A	0.5	0.2	0.9	0.01	0	0.81	1	0	1.81		
	2Dgraph_Label		#N/A	#N/A	#N/A	#N/A	0.01	0.01	0.01	0.01	0	0	0	0	0		
															4.64	20.77	4
BBC_Artist	2Dgraph_Height		#N/A	0.01	0.01	#N/A	0.9	0.2	0.5	0.01	0	0.81	0.51	0	1.32		
	2Dgraph_Width		#N/A	#N/A	#N/A	#N/A	0.9	0.5	0.2	0.01	0	0	0	0	0		
BBC_TW	2Dgraph_X		0.01	#N/A	0.9	#N/A	0.2	0.5	0.9	0.01	0.81	0	1	0	1.81		
BBC_LW	2Dgraph_Y		0.01	#N/A	0.9	#N/A	0.5	0.2	0.9	0.01	0.51	0	1	0	1.51		
	2Dgraph_Label		#N/A	#N/A	#N/A	#N/A	0.01	0.01	0.01	0.01	0	0	0	0	0		
															4.64	20.37	5
BBC_Artist	2Dgraph_Height		#N/A	0.01	0.01	#N/A	0.9	0.2	0.5	0.01	0	0.81	0.51	0	1.32		
	2Dgraph_Width		#N/A	#N/A	#N/A	#N/A	0.9	0.5	0.2	0.01	0	0	0	0	0		
BBC_LW	2Dgraph_X		0.01	#N/A	0.9	#N/A	0.2	0.5	0.9	0.01	0.81	0	1	0	1.81		
BBC_TW	2Dgraph_Y		0.01	#N/A	0.9	#N/A	0.5	0.2	0.9	0.01	0.51	0	1	0	1.51		
	2Dgraph_Label		#N/A	#N/A	#N/A	#N/A	0.01	0.01	0.01	0.01	0	0	0	0	0		
															4.64	20.37	6
BBC_LW	2Dgraph_Height		#N/A	0.01	0.9	#N/A	0.9	0.2	0.5	0.01	0	0.81	0.6	0	1.41		
	2Dgraph_Width		#N/A	#N/A	#N/A	#N/A	0.9	0.5	0.2	0.01	0	0	0	0	0		
BBC_Artist	2Dgraph_X		0.01	#N/A	0.01	#N/A	0.2	0.5	0.9	0.01	0.81	0	0.11	0	0.92		
BBC_TW	2Dgraph_Y		0.9	#N/A	0.01	#N/A	0.5	0.2	0.9	0.01	0.6	0	0.11	0	0.71		
	2Dgraph_Label		#N/A	#N/A	#N/A	#N/A	0.01	0.01	0.01	0.01	0	0	0	0	0		
															3.04	19.97	7

Figure 6.16: The process of deciding whether visualizations should be annotated.

6.9.3 Prefuse Visualizations

Prefuse [HCL05] provides a visualization framework on which new visualization styles can be built. Many experimental visualization styles are included in the default package. Using

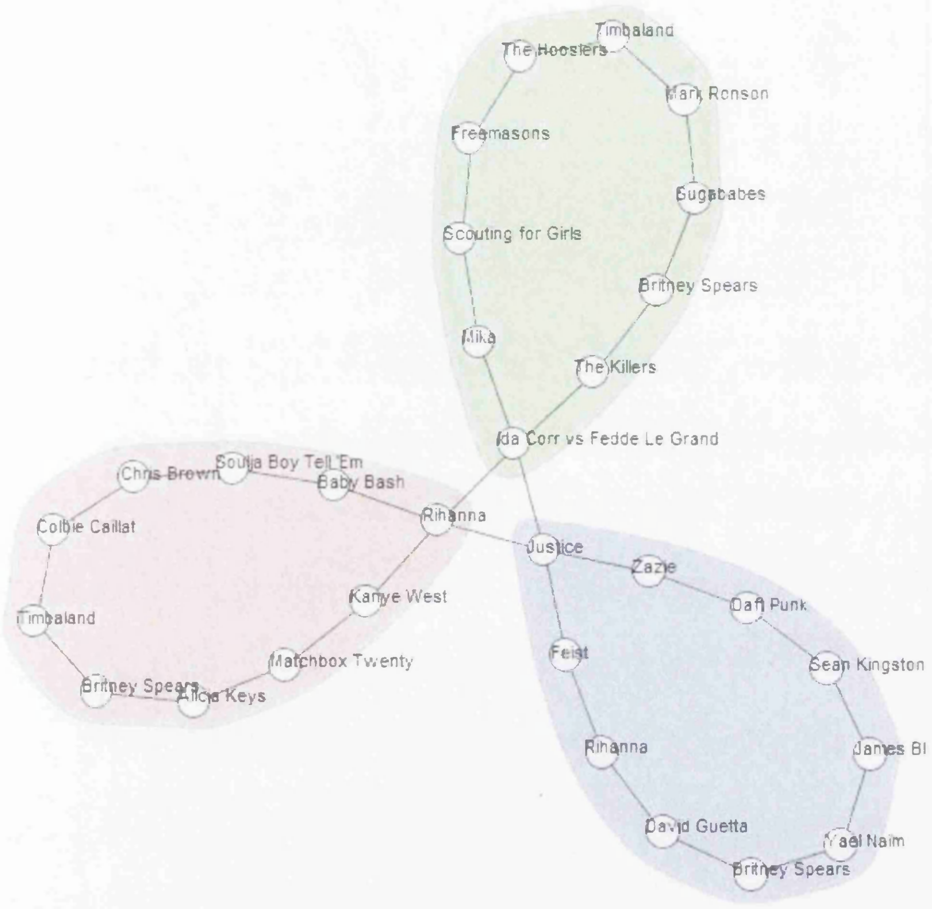


Figure 6.17: The iTunes music chart visualized using Prefuse's Aggregate Graph visualization style.

SemViz, we have produced visualization using 2 of these and the iTunes music chart data set:

Aggregate Graph (figure 6.17) The chart entries for 3 countries are shown. Only the artist name is give, but each chart entry is grouped by the country's chart it appears in.

Graph Network (figure 6.18) The chart entries for 4 countries are shown. Each song name node is connected to the country node in who's chart it appears. Each song name node is also connected to the artist name node of the artist who sung the song. In this way, there are no duplicate nodes, and we can see which artists have multiple songs in the 4 countries' charts. This provides a similar type of insight to that gained from the TreeMap shown in 6.14.

Both visualizations take the form of interactive Java applets where individual nodes can be dragged and moved around the display. This level of interaction is greater than that provided by ILOG Discovery.

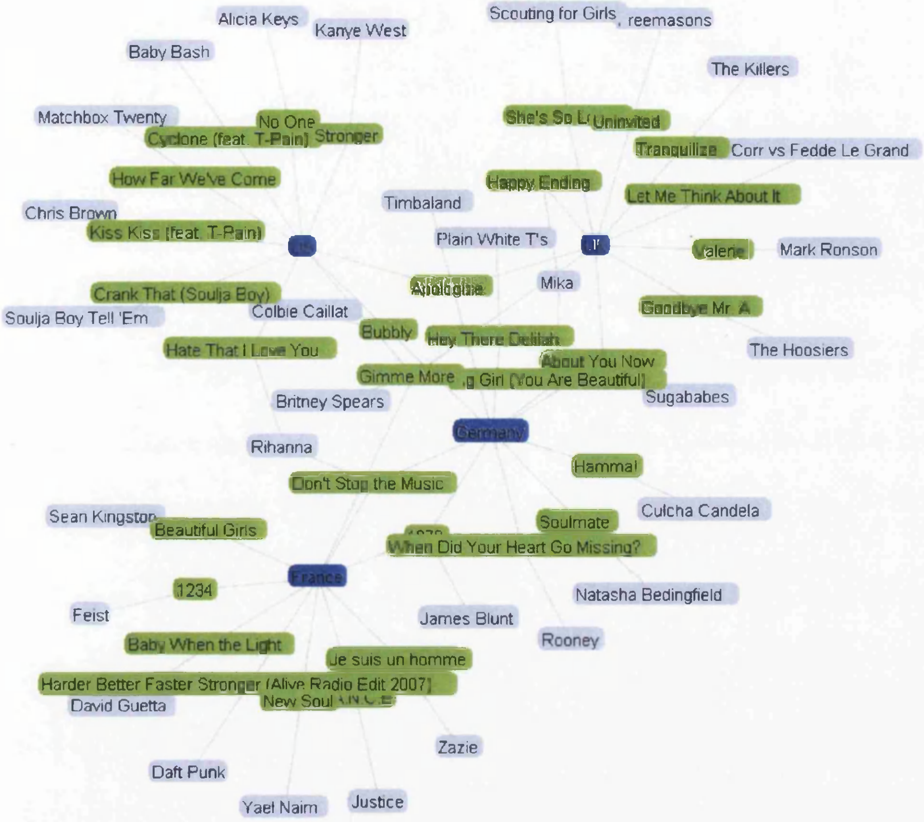


Figure 6.18: The iTunes music chart (4 countries only) visualized using Prefuse's Graph Network visualization style.

6.10 Summary

In this chapter, we have described a pragmatic method of producing automatic visualizations using domain knowledge captured in ontologies. The Domain Ontology (DO) captures knowledge about the source web pages' subject domain; the Visual Representation Ontologies (VRO) capture the semantics of popular visual representations/styles; and the Semantic Bridging Ontology (SBO) holds key knowledge about the relationships between data entities of the source subject domain and the visual artefacts of the target visualizations. We have rationalised the relationships between concepts in the DO and VRO into a core set of *semantic equivalences* which form the basis of the scoring algorithm.

We have adapted an existing ontology mapping algorithm to encompass Certainty Factors. This algorithm has a good trade-off between computational cost and ability to produce high quality automatic visualizations. We have implemented the visualization pipeline in a prototype, SemViz which functions end-to-end from source web page to target visualization. SemViz interfaces with two public-domain visualization frameworks.

We have shown demonstrable results by taking music chart web pages and using SemViz to interface with the ILOG Discovery and Prefuse visualization toolkits to produce examples in a variety of popular visualization styles. The visualization pipeline and supporting data-structures provide a good framework on which to extend and refine the current ontology mapping algorithm.

6.10.1 Advantages

Knowledge-based SemViz utilises knowledge stored in the form of ontologies in order to improve the visualization process. Most automatic visualization toolkits (including VizThis) rely on analysing the nature of the source data in order to create cognitively useful visualizations. The nature of the data includes type information (string, number, date etc.) and the variance of the data (number of unique values). As such, the ability of the toolkit to automatically produce useful visualizations is limited. In considering the additional semantics stored in a Domain Ontology, a toolkit has a far greater richness of semantics at its disposal. This ultimately will result in higher quality visualizations.

Toolkit Re-use The core competency of SemViz is mapping source data entities to target visualization artefacts. SemViz does not attempt to produce new graphical visualizations itself. As such, it utilises the vast body of work which the visualization community have put into creating visualization toolkits – specifically visualization toolkits which expose their semantics to other applications. Compare this approach with VizThis which attempts to be a multi-function tool and perform data analysis, source to target mapping, and finally visualization graphic production. As described in section 5.8, the results are of limited quality.

Gallery Selection In the SemViz approach to visualization, we make the judgement that it is better to show the user a variety of possible visualizations which are deemed to have high cognitive value, rather than attempting to show the one visualization which the system thinks has the highest cognitive value. The reason for this is that the human brain is very good at finding the best solution from a limited selection (say 10 options). The human brain is poor at finding the best solution from a large selection (50 or more options). Therefore, we can utilise computational power to reduce the users search space from hundreds or thousands of options, down to only 10 or so. By allowing a system the freedom to present the user with multiple options, we do not have to build a system which always gives the best answer first.

Focus on information SemViz assumes that the source data is of a high quality. This means that there is no need for data cleansing, and there are no complexities caused by the ambiguities of the schema design. In this way, again, SemViz concentrates on its core competency of mapping rather than data cleansing or cross-mapping features which VizThis has.

6.10.2 Disadvantages

Ontology Quality The quality of the visualizations produced by SemViz is dependent on the comprehensiveness of the Domain Ontology and the Visual Representation

Ontology. Visualization quality also depends on the applicability of the source web page subject area to the Domain Ontology subject area. For example, a web page about cricket would have limited success at being visualised if the only domain ontology available were about football. Finally, the visualization quality depends on the applicability of existing Semantic Bridges (stored in the SBO) to the source web page. In summary, the quality of the visualizations is very dependant on the quality and appropriateness of the ontologies. This currently poses a problem due to the limited number of ontologies available for all but the most specialised areas (such as bioinformatics or chemistry).

Computational Complexity The computational complexity of the ontology mapping algorithm used in SemViz is factorial. As such, the time to score all possible visualizations can become unreasonable when larger datasets are considered. The use of the Semantic Bridge Ontology helps reduce the complexity by "invalidating" some visualization possibilities based on the appropriateness score of the semantic bridges. However, again, the effectiveness of this depends on the quality of the ontologies, specifically the Semantic Bridge Ontology.

Openness of Toolkits SemViz is able to achieve improvements in automatic visualization by utilising existing visualization toolkits. However, in order to exploit the visualization capabilities, the toolkits must expose their functionality in some way. Ideally, they must programmatically allow the mappings between the source data entities and target visual artefacts to be specified by another system (in this case, SemViz). If this is not the case, but there is an open, non-binary file format (as with ILOG Discovery), then custom data integration code can be written. Not all visualization toolkits provide this openness, especially those with a more commercial origin. This could limit SemViz from utilising more modern visualization techniques if an implementation with open interfaces is not available.

Chapter 7

SemViz Case Study : Scalability Evaluation and User Testing

Contents

7.1 Introduction

In the previous chapter we introduced SemViz, a software tool and visualization pipeline for scoring and ranking visualizations according to knowledge stored in ontologies. The examples we gave focus on the subject domain of music charts with visualizations created from two different source web pages (iTunes music chart and BBC Top 40 music chart). The examples given, although effective, were relatively small. The Domain Ontology for music charts contains 8 concepts (Record Label, Artist, Song, Country, Genre, Current Chart Position, Last Week Chart Position, and Weeks in Chart). Typically, a subject domain which a user wishes to visualise might contain 10's or 100's of different concepts. Therefore, in this chapter, we set out to assess the scalability and validity of SemViz by investigating a subject domain with a larger number of concepts.

In addition, the previous chapter introducing SemViz does not provide any user evaluation of the visualization results obtained. Instead, subjective evaluations are made on the quality of the visualizations based on the author's own judgement. Clearly, in order for the SemViz approach to validated, external and independent evaluation of resulting visualizations must be conducted.

This chapter is therefore split up into two major sub-sections:

Scalability Evaluation The purpose of scalability evaluation is to judge whether the techniques of SemViz described in chapter 6 can be scaled to a larger number of domain concepts. We also need to measure the validity of the scores which the SemViz algorithm gives for various scenarios.

User Testing We perform user testing to ascertain whether the relative scores given to various visualizations are indeed a valid measure of their cognitive value. We do this by giving a selection of subject users various tasks. These vary from closed questioned, objective, timed tasks to open-ended, subjective, discussions. Combining the results of all these tasks together will give an indication as to the validity of the SemViz technique. It will also highlight areas where further work is needed. The final user testing task is to ask an expert user to inspect the visualizations created in order to comment on their cognitive value and to suggest improvements.

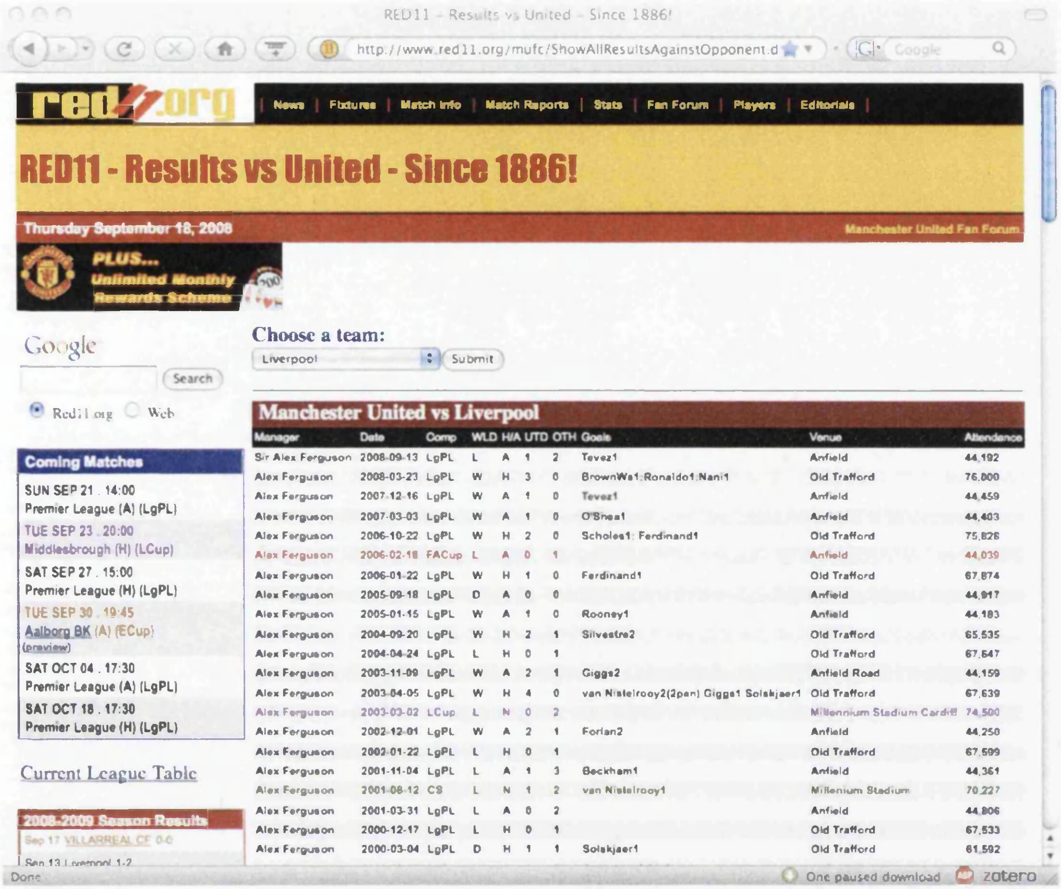
7.2 Subject Domain - Football

Football (or soccer) is one of the worlds most popular sports. Whether measured in terms of revenue generated, number of players (professional or amateur), or number of viewers (live or televised), there is no doubting the popularity of the sport. One aspect which fascinates fans the world-over is football related statistics. The long history of the game together with the sheer number of professional games played every year means that there is a vast quantity of statistical data available.

With the advent of the world-wide-web, these statistics have been published both by football clubs' official websites, but more often, by amateur fans who wish to share their love of the sport (and a particular team) with other fans. The statistics largely exist as raw information in the form of tables of results. Often these tables are organized into different categorisations (e.g., by team or year), and also by different sort order. However, visual presentation methods are seldom used. In particular information visualization techniques seem very rarely to be used. Therefore, this presents a good opportunity to test whether the SemViz approach (algorithm, methodology and user-interaction style) can be successfully applied to the areas of web-based football statistics.

7.2.1 Source Websites

We have identified 4 popular and detailed websites which record football statistics. We have not restricted ourselves to any particular team, country or competition. The websites we have used are shown in table 7.1. The use of such a varied sample of source websites will allow us to test out the approach of using a Domain Ontology in terms of its ability to cope with a wide-variety of domain concepts, all under the umbrella of web-based football statistics.


RED11 - Results vs United - Since 1886!
http://www.red11.org/mufc/ShowAllResultsAgainstOpponent.d
Google
red11.org
News | Fixtures | Match Info | Match Reports | Stats | Fan Forum | Players | Editorials
Manchester United Fan Forum
PLUS... Unlimited Monthly Rewards Scheme
Google
Search
Red11.org
Web
Current League Table
2008-2009 Season Results
Sep 17 VILLARREAL CF 0-0
Sep 13 Liverpool 1-2
Done
One paused download
zotero

RED11 - Results vs United - Since 1886!

Thursday September 18, 2008

Coming Matches

SUN SEP 21 . 14:00
Premier League (A) (LgPL)
TUE SEP 23 . 20:00
Middlesbrough (H) (LCup)
SAT SEP 27 . 15:00
Premier League (H) (LgPL)
TUE SEP 30 . 19:45
Aalborg BK (A) (ECup)
(preview)
SAT OCT 04 . 17:30
Premier League (A) (LgPL)
SAT OCT 18 . 17:30
Premier League (H) (LgPL)

Choose a team: Liverpool [Submit]

Manchester United vs Liverpool

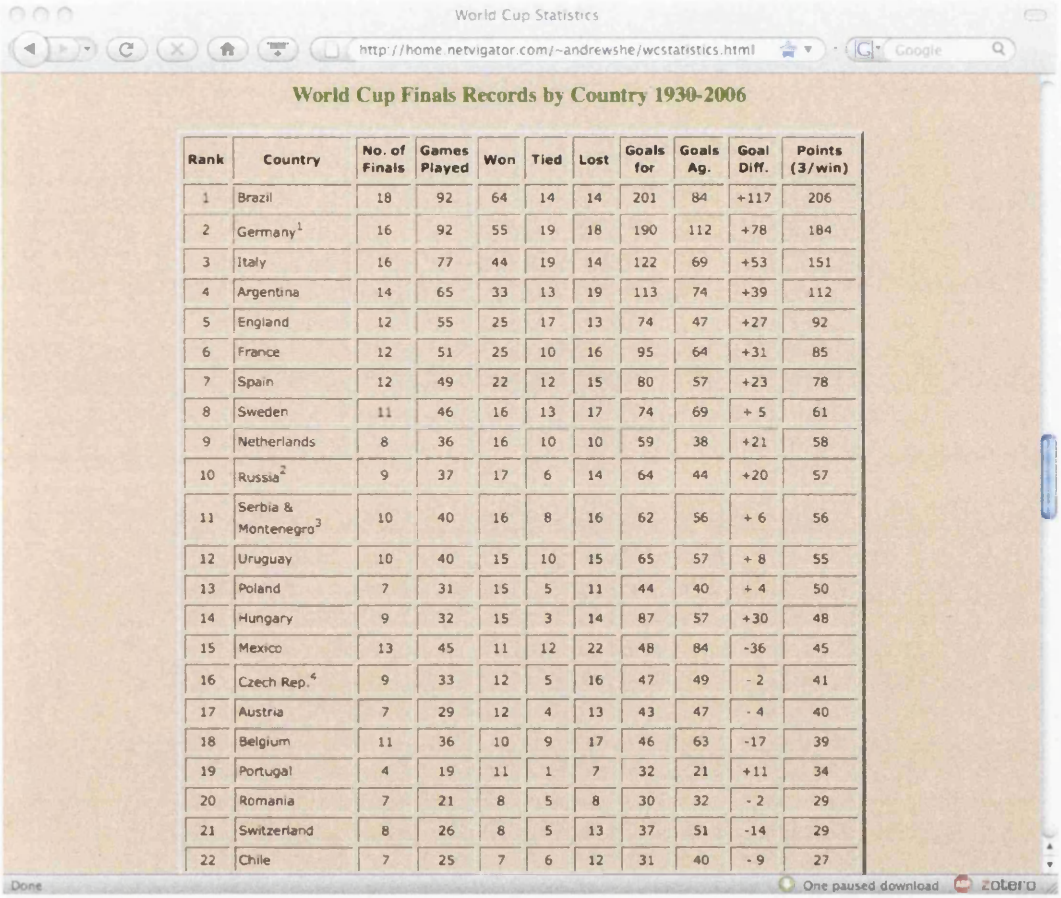
Manager	Date	Comp	WLD	H/A	UTD	OTH	Goals	Venue	Attendance
Sir Alex Ferguson	2008-09-13	LgPL	L	A	1	2	Tevez1	Anfield	44,192
Alex Ferguson	2008-03-23	LgPL	W	H	3	0	BrownWe1;Ronaldo1;Nani1	Old Trafford	76,000
Alex Ferguson	2007-12-16	LgPL	W	A	1	0	Tevez1	Anfield	44,459
Alex Ferguson	2007-03-03	LgPL	W	A	1	0	O'Shea1	Anfield	44,403
Alex Ferguson	2006-10-22	LgPL	W	H	2	0	Scholes1; Ferdinand1	Old Trafford	75,828
Alex Ferguson	2006-02-18	FACup	L	A	0	1		Anfield	44,039
Alex Ferguson	2006-01-22	LgPL	W	H	1	0	Ferdinand1	Old Trafford	67,874
Alex Ferguson	2005-09-18	LgPL	D	A	0	0		Anfield	44,917
Alex Ferguson	2005-01-15	LgPL	W	A	1	0	Rooney1	Anfield	44,183
Alex Ferguson	2004-09-20	LgPL	W	H	2	1	Silvestre2	Old Trafford	65,535
Alex Ferguson	2004-04-24	LgPL	L	H	0	1		Old Trafford	67,647
Alex Ferguson	2003-11-09	LgPL	W	A	2	1	Giggs2	Anfield Road	44,159
Alex Ferguson	2003-04-05	LgPL	W	H	4	0	van Nistelrooy2(2pen) Giggs1 Solskjaer1	Old Trafford	67,639
Alex Ferguson	2003-03-02	LCup	L	N	0	2		Millennium Stadium Cardiff	74,500
Alex Ferguson	2002-12-01	LgPL	W	A	2	1	Forlan2	Anfield	44,250
Alex Ferguson	2002-01-22	LgPL	L	H	0	1		Old Trafford	67,599
Alex Ferguson	2001-11-04	LgPL	L	A	1	3	Beckham1	Anfield	44,361
Alex Ferguson	2001-08-12	CS	L	N	1	2	van Nistelrooy1	Millenium Stadium	70,227
Alex Ferguson	2001-03-31	LgPL	L	A	0	2		Anfield	44,806
Alex Ferguson	2000-12-17	LgPL	L	H	0	1		Old Trafford	67,533
Alex Ferguson	2000-03-04	LgPL	D	H	1	1	Solskjaer1	Old Trafford	61,592

Figure 7.1: The Red11.org Manchester United Fan Website. Note: Comp - Competition; WLD - Win, Loose or Draw; H/A - Home or Away match; UTD - Number of goals scored by Manchester United; OTH - Number of goals scored by the Other team.

Website Name	Website Type	Website Subject Sub-Domain
Manutd.com	Official site	Manchester United stats
Red11.org	Fan site	Manchester United stats
Football.co.uk	Football Magazine	UK Premiership stats
Andrew's FIFA World Cup Stats	Fan site	World Cup stats

Table 7.1: The football subject domain source websites.

One way we wish to evaluate the scalability of SemViz is by how far it scales vertically with respect to Subject Sub-Domains. For example, we are modelling the domain of football. However, within this domain are sub-domains which specialise in specific areas of football. In the web sites we use as sources (see table 7.1), these are: Manchester United; UK Premiership; and FIFA World Cup. This provides additional challenges to the evaluation of SemViz. We illustrate the nature of this relationship in figure 7.5.



World Cup Finals Records by Country 1930-2006

Rank	Country	No. of Finals	Games Played	Won	Tied	Lost	Goals for	Goals Ag.	Goal Diff.	Points (3/win)
1	Brazil	18	92	64	14	14	201	84	+117	206
2	Germany[1]	16	92	55	19	18	190	112	+78	184
3	Italy	16	77	44	19	14	122	69	+53	151
4	Argentina	14	65	33	13	19	113	74	+39	112
5	England	12	55	25	17	13	74	47	+27	92
6	France	12	51	25	10	16	95	64	+31	85
7	Spain	12	49	22	12	15	80	57	+23	78
8	Sweden	11	46	16	13	17	74	69	+ 5	61
9	Netherlands	8	36	16	10	10	59	38	+21	58
10	Russia[2]	9	37	17	6	14	64	44	+20	57
11	Serbia & Montenegro[3]	10	40	16	8	16	62	56	+ 6	56
12	Uruguay	10	40	15	10	15	65	57	+ 8	55
13	Poland	7	31	15	5	11	44	40	+ 4	50
14	Hungary	9	32	15	3	14	87	57	+30	48
15	Mexico	13	45	11	12	22	48	84	-36	45
16	Czech Rep.[4]	9	33	12	5	16	47	49	- 2	41
17	Austria	7	29	12	4	13	43	47	- 4	40
18	Belgium	11	36	10	9	17	46	63	-17	39
19	Portugal	4	19	11	1	7	32	21	+11	34
20	Romania	7	21	8	5	8	30	32	- 2	29
21	Switzerland	8	26	8	5	13	37	51	-14	29
22	Chile	7	25	7	6	12	31	40	- 9	27

Figure 7.2: Andrew's FIFA World Cup Statistics Website – Web address: http://home.netvigator.com/˜andrewshe/.

7.3 Football Visualizations

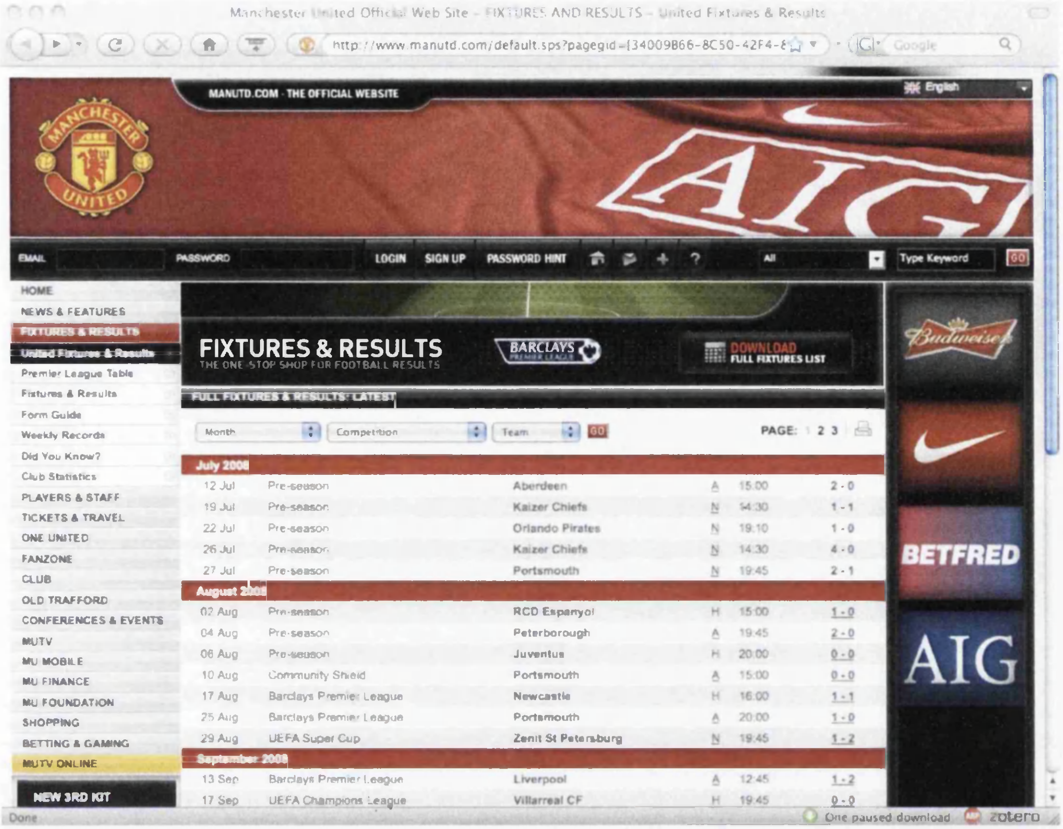
A summary of the visualizations produced as part of the Scalability Evaluation and User Testing tasks is shown in tables 7.2, 7.3, and 7.4. All visualizations referenced are shown from figures 7.6 to 7.15.

7.4 Scalability Evaluation

7.4.1 Approach

In order to evaluate the scalability and validity of the SemViz approach, we demonstrate the visualization of a series of example data-sets, all using the football domain ontology. Our approach is as follows:

1. Create the Domain Ontology from analysing the source data entities (i.e., DO concepts) from Red11.org and Andrew's FIFA World Cup Statistics. The Domain



July 2008					
12 Jul	Pre-season	Aberdeen	A	15:00	2 - 0
19 Jul	Pre-season	Kaizer Chiefs	N	14:30	1 - 1
22 Jul	Pre-season	Orlando Pirates	N	19:10	1 - 0
26 Jul	Pre-season	Kaizer Chiefs	N	14:30	4 - 0
27 Jul	Pre-season	Portsmouth	N	19:45	2 - 1
August 2008					
02 Aug	Pre-season	RCD Espanyol	H	15:00	1 - 0
04 Aug	Pre-season	Peterborough	A	19:45	2 - 0
06 Aug	Pre-season	Juventus	H	20:00	0 - 0
10 Aug	Community Shield	Portsmouth	A	15:00	0 - 0
17 Aug	Barclays Premier League	Newcastle	H	16:00	1 - 1
25 Aug	Barclays Premier League	Portsmouth	A	20:00	1 - 0
29 Aug	UEFA Super Cup	Zenit St Petersburg	N	19:45	1 - 2
September 2008					
13 Sep	Barclays Premier League	Liverpool	A	12:45	1 - 2
17 Sep	UEFA Champions League	Villarreal CF	H	19:45	0 - 0

Figure 7.3: The Official ManchesterUnited.com Football Club Website.

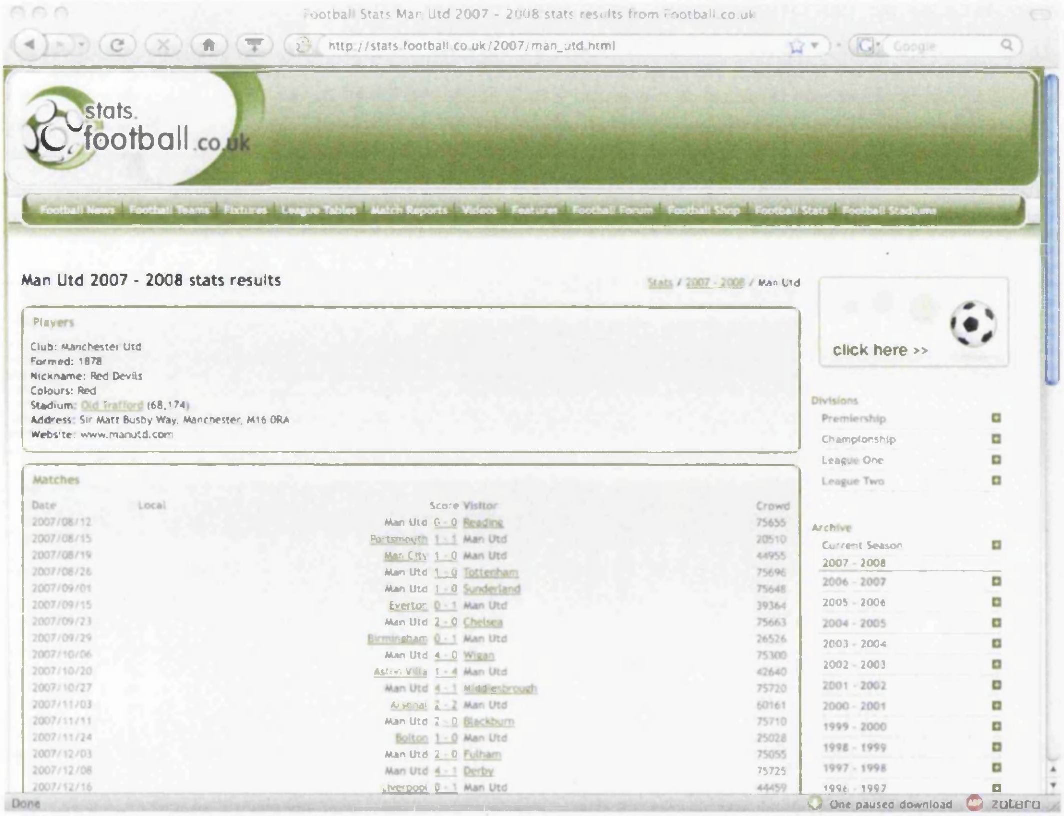
Ontology took 22 hours to create and add certainty factor values. Most of this time was spent deciding on the certainty factor weightings for each concept pair for each relationship type. There are 4,389 relationships (i.e., certainty factor values) which is too many to show here. Therefore, we just show the extracted concepts names in table 7.5. However, we show an extract of the Domain Ontology weightings in figure 7.16.

2. Create visualizations using instance data from Red11.org and Andrew's FIFA World Cup Statistics websites.

3. Create visualizations using instance data from ManUtd.com and Football.co.uk (but using the DO concepts created from Red11.org and Andrew's FIFA World Cup Statistics).

7.4.2 Tests and Results

In this section we detail each of the tests we applied to the system in order to assess the scalability and validity of the SemViz algorithm on a subject domain with a large Domain Ontology.



Man Utd 2007 - 2008 stats results

Stats / 2007 - 2008 / Man Utd

Players

Club: Manchester Utd
Formed: 1878
Nickname: Red Devils
Colours: Red
Stadium: Old Trafford (68,174)
Address: Sir Matt Busby Way, Manchester, M16 0RA
Website: www.manutd.com

Matches

Date	Local	Score	Visitor	Crowd
2007/08/12	Man Utd	0 - 0	Reading	75655
2007/08/15	Portsmouth	1 - 1	Man Utd	20510
2007/08/19	Man City	1 - 0	Man Utd	44955
2007/08/26	Man Utd	1 - 0	Tottenham	75696
2007/09/01	Man Utd	1 - 0	Sunderland	75648
2007/09/15	Everton	0 - 1	Man Utd	39364
2007/09/23	Man Utd	2 - 0	Chelsea	75663
2007/09/29	Birmingham	0 - 1	Man Utd	26526
2007/10/06	Man Utd	4 - 0	Wigan	75300
2007/10/20	Aston Villa	1 - 4	Man Utd	42640
2007/10/27	Man Utd	4 - 1	Middlesbrough	75720
2007/11/03	Arsenal	2 - 2	Man Utd	60161
2007/11/11	Man Utd	2 - 0	Blackburn	75710
2007/11/24	Bolton	1 - 0	Man Utd	25028
2007/12/03	Man Utd	2 - 0	Fulham	75055
2007/12/08	Man Utd	4 - 1	Derby	75725
2007/12/16	Liverpool	0 - 1	Man Utd	44459

Divisions

- Premiership ⊞
- Championship ⊞
- League One ⊞
- League Two ⊞

Archive

- Current Season ⊞
- 2007 - 2008
- 2006 - 2007 ⊞
- 2005 - 2006 ⊞
- 2004 - 2005 ⊞
- 2003 - 2004 ⊞
- 2002 - 2003 ⊞
- 2001 - 2002 ⊞
- 2000 - 2001 ⊞
- 1999 - 2000 ⊞
- 1998 - 1999 ⊞
- 1997 - 1998 ⊞
- 1996 - 1997 ⊞

Figure 7.4: The Football.co.uk Magazine Website. Note the difference in terminology used from that captured in the Football Domain Ontology: "Local" is used instead of "Home Team"; "Visitor" is used instead of "Away Team"; and "Crowd" is used instead of "Attendance".

7.4.2.1 Scalability of DO in terms of number of concepts

Test ID: s01

Test Description: Can the SemViz algorithm still produce visualizations with a high degree of cognitive value, even when the number of concepts is increased from the initially tested 8 (see the Music Charts Domain Ontology in Section 6) to 33 (as used by the Football Domain Ontology)?

Expected Result: The visualizations produced by SemViz will be ranked according to their cognitive value with the highest scoring visualization being the one which SemViz deems to have the highest cognitive value.

Visualizations Inspected: Figure 7.6 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website (2D graph).
Figure 7.13 - The highest scoring visualization for the "World Cup Finals Records by Country 1930-2006" page on the Andrew's FIFA World Cup Statistics website (Parallel Coordinates).

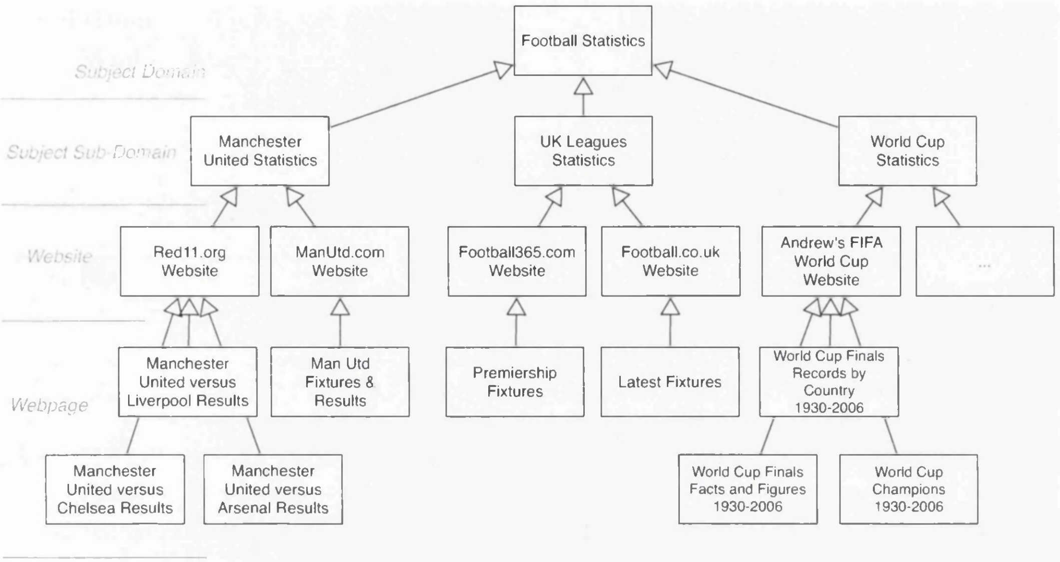

Football Statistics
Subject Domain
Subject Sub-Domain
Manchester United Statistics
UK Leagues Statistics
World Cup Statistics
Website
Red11.org Website
ManUtd.com Website
Football365.com Website
Football.co.uk Website
Andrew's FIFA World Cup Website
...
Webpage
Manchester United versus Liverpool Results
Man Utd Fixtures & Results
Premiership Fixtures
Latest Fixtures
World Cup Finals Records by Country 1930-2006
Manchester United versus Chelsea Results
Manchester United versus Arsenal Results
World Cup Finals Facts and Figures 1930-2006
World Cup Champions 1930-2006

Figure 7.5: The Subject Domain inheritance model.

Result: The two visualizations inspected all have a high cognitive value. The mappings between the source data entities and target representation artefacts for the 2D graph result in a good visualization. The most important quantitative values (Date and Attendance) are mapped to the X axis and Y axis respectively. The qualitative values (Manager and Venue) are mapped to the qualitative artefacts. Also, the Parallel Coordinates visualization has a good ordering of source data entities which respects priority (most important on the left) and also complementary entities are plotted next to each other.

Conclusion: *Pass*

7.4.2.2 Scalability of SemViz approach in terms of number of records.

Test ID: s02

Test Description: Can the SemViz algorithm still produce visualizations with a high degree of cognitive value, even when the number of records is increased from the initially tested 40 (from the BBC Top 40 Music Chart in Chapter 6) to 175 (From the Manchester United versus Liverpool Results Page on Red11.org)?

Expected Result: The visualizations will still be readable and we will gain an insight into the scalability issues which may result when there are an even greater number of records.

Visualizations Inspected: Figure 7.6 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website (2D graph).
Figure 7.10 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website (Parallel Coordinates).
Figure 7.11 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website (TreeMap).

Sub-Dom	Website	Page	Source	Target	Viz Tech	Score	Fig
ManU	Red11.org	MvsL	Date	x	2D graph	92.17	7.6
			Attendance	y		(highest)	
			OTH	w			
			UTD	h			
			Manager	label			
			Venue	color			
ManU	Red11.org	MvsL	Date	x	2D graph	83.67	7.7
			Attendance	y		(median)	
			OTH	w			
			Venue	h			
			UTD	label			
			Manager	color			
ManU	Red11.org	MvsL	Manager	x	2D graph	77.17	7.8
			Venue	y		(lowest)	
			Date	w			
			UTD	h			
			Attendance	label			
			OTH	color			
ManU	Red11.org	MvsL	Date	x	2D graph	90.99	7.9
			WLD	y		(highest)	
			Comp	w			
			Manager	h			
			HA	label			
			Venue	color			

Table 7.2: Figures 7.6, 7.7 and 7.8 represent the highest ranking, median ranking and lowest ranking of the visualizations for the same sub-set of the source data using a 2D graph. Figure 7.9 represents the highest ranking visualization for a different sub-set of the source data (note the difference in Source Entities used). **Key**: ManU - Manchester United Sub-Domain; MvsL - Manchester United versus Liverpool, all results since 1886.

Result:. For the 2D graph and Parallel Coordinates, there is not a large overhead in having a greater number of records. However, the TreeMap quickly becomes unusable when there are a large number of source data entities (i.e., levels) and records. Therefore, we repeat the same exercise, but reduce the number of source data entities from 6 down to 3. This is illustrated in figure 7.12 which is a more clear visualization with a higher cognitive value.

Conclusion: *Partial Pass - TreeMaps require consideration of number of records.*

7.4.2.3 Comparing scores between visualizations which use *the same* visualization technique

Test ID: s03

Test Description: Validity of the visualizations score for comparing between visualizations

Sub-Dom	Website	Page	Source	Target	Viz Tech	Score	Fig
ManU	Red11.org	MvsL	UTD	Ypoint0	Parallel Co	80.01	7.10
			OTH	Ypoint1		(highest)	
			Manager	Ypoint2			
			Date	Ypoint3			
			Attendance	Ypoint4			
			Venue	Ypoint5			
ManU	Red11.org	MvsL	Attendance	Color	TreeMap	78.23	7.11
			UTD	Level0		(highest)	
			OTH	Level1			
			Date	Level2			
			Venue	Level3			
			Manager	Level4			
ManU	Red11.org	MvsL	Competition	Color	TreeMap	23.31	7.12
			Manager	Level0		(highest)	
			WLD	Level1			

Table 7.3: Figures 7.10, and 7.11 use the same source data sub-set as figures 7.6, 7.7 and 7.8 but are visualized using Parallel Coordinates and a Tree Map (respectively). Figure 7.12 is a TreeMap with a reduced source data sub-set (3 source entities, instead of 6). **Key**: ManU - Manchester United Sub-Domain; MvsL - Manchester United versus Liverpool, all results since 1886.

which use the same visualization technique (e.g., all are 2D graph).

Expected Result: Visualizations with a higher score are cognitively more valuable than those with a lower score.

Visualizations Inspected: Figure 7.6 - The highest ranking visualization for the Manchester United versus Liverpool page on the Red11.org website (2D graph).
Figure 7.7 - A middle ranking visualization for the Manchester United versus Liverpool page on the Red11.org website (2D graph).
Figure 7.8 - The lowest ranking visualization for the Manchester United versus Liverpool page on the Red11.org website (2D graph).

Result:. The three visualizations inspected do have a lower degree of cognitive value from figure 7.6, to figure 7.7 and figure 7.8. This is based on the author's ability to find individual values and also to discern patterns in the information.

Conclusion: *Pass*

7.4.2.4 Comparing between visualizations which use *different* visualization techniques

Test ID: s04

Test Description: Validity of the visualizations score for comparing between visualizations which use different visualization techniques (e.g., one is a 2D graph, one is a Parallel

Sub-Dom	Website	Page	Source	Target	Viz Tech	Score	Fig
WorldC	Andrew	WC Finals	Country	Ypoint0	Parallel Co	82.36	7.13
			Rank	Ypoint1		(highest)	
			Points	Ypoint2			
			Won	Ypoint3			
			Tied	Ypoint4			
			Lost	Ypoint5			
UK Prem	FootB.co	MU Res	Date	x	2D graph	86.88	7.14
			Crowd	y		(highest)	
			Home	w			
			Away	h			
			Local	label			
			Visitor	color			
ManU	MU.com	ResFixs	Home/Away	Color	TreeMap	23.61	7.15
			Away Team	Level0		(highest)	
			Comp	Level1			

Table 7.4: Figures 7.13, 7.14 and 7.15 all use different source data sets from different websites which are from different sub-domains. Each visualization is the highest scoring one amongst the 3 possible visualization techniques (2D Graph, TreeMap and Parallel Coordinates). **Key**: WorldC - FIFA World Cup Sub-Domain; UK Prem - UK Premiership League Sub-Domain; ManU - Manchester United Sub-Domain; Andrew - Andrews FIFA World Cup Statistics Website; FootB.co - Football.co.uk Website; MU.com - ManchesterUnited.com; WC Finals - World Cup Finals Records by Country 1930-2006; MU Res - Football Stats Man Utd 2007 - 2008 Season; ResFixs - Manchester United Fixtures and Results.

Coordinates and one is a Tree Map).

Expected Result: The cognitive value of the three different visualization techniques will be reflected in the scores given.

Visualizations Inspected: Figure 7.6 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website (2D graph).
Figure 7.10 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website (Parallel Coordinates).
Figure 7.11 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website (TreeMap).

Result:. The author has judged that the visualizations have this order of cognitive value (highest first): 2D graph; Parallel Coordinates; and TreeMap. This is also reflected in the scores given. Note that the authors judgement will be compared with the judgement of independent user testers in section 7.4.

Conclusion: *Pass*

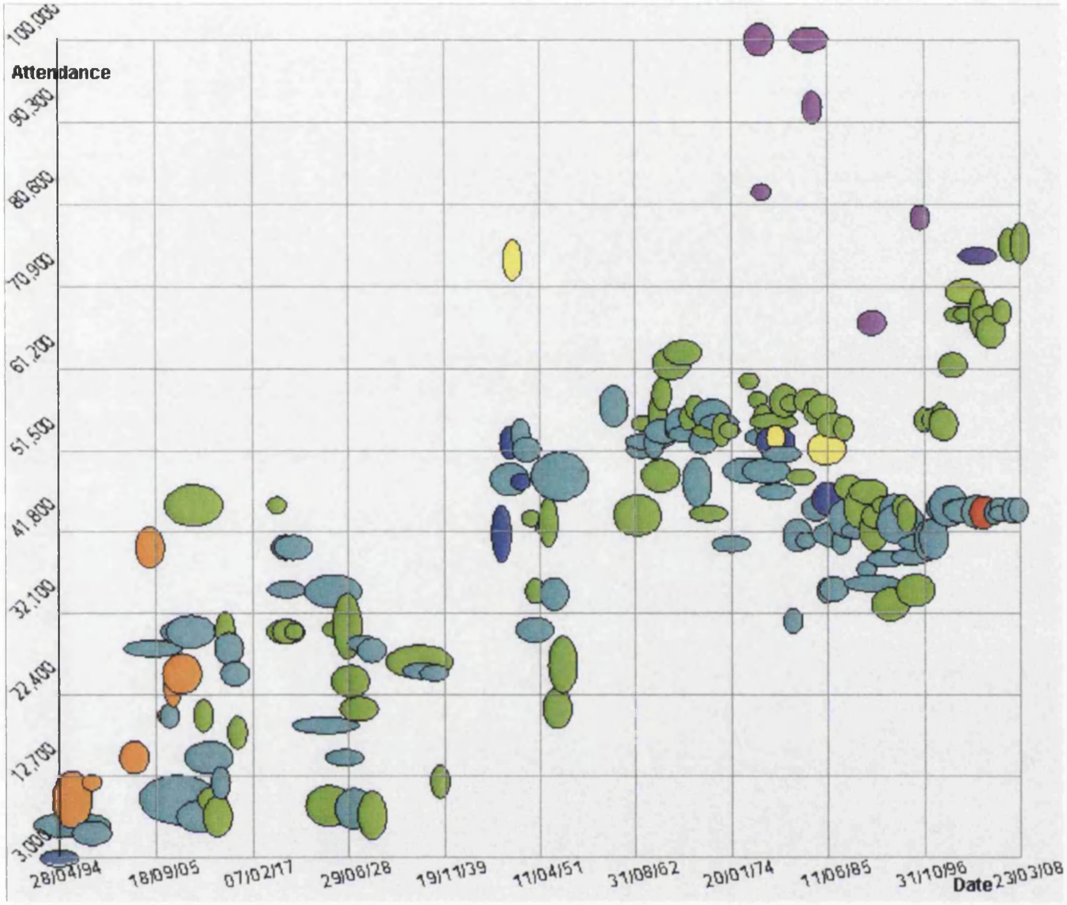


Figure 7.6: 2D graph visualization of Manchester United subject sub-domain, Manchester United versus Liverpool matches from Red11.org website with mappings: x to Date; y to Attendance; width to Away Score; height to Home Score; label to Manager; color to Venue. **Score = 92.17** (Highest ranking score)

7.4.2.5 Comparing the visualization of *different sub-sets* of the source data attributes using the same visualization technique

Test ID: s05

Test Description: Validity of the visualizations score for comparing the visualization of different sub-sets of the source data attributes using the same visualization technique (e.g., one visualization shows a 2D graph of: Home Score, Away Score, Attendance; another shows Home Score, Date, Manager; another shows Venue, Competition, Date).

Expected Result: The visualization which uses the sub-set of source attribute data which can give the most cognitive value when visualised has the highest score.

Visualizations Inspected: Figure 7.6 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website when the sub-set of source data entities is Date, Attendance, UTD, OTH, Manager, Venue (2D graph).

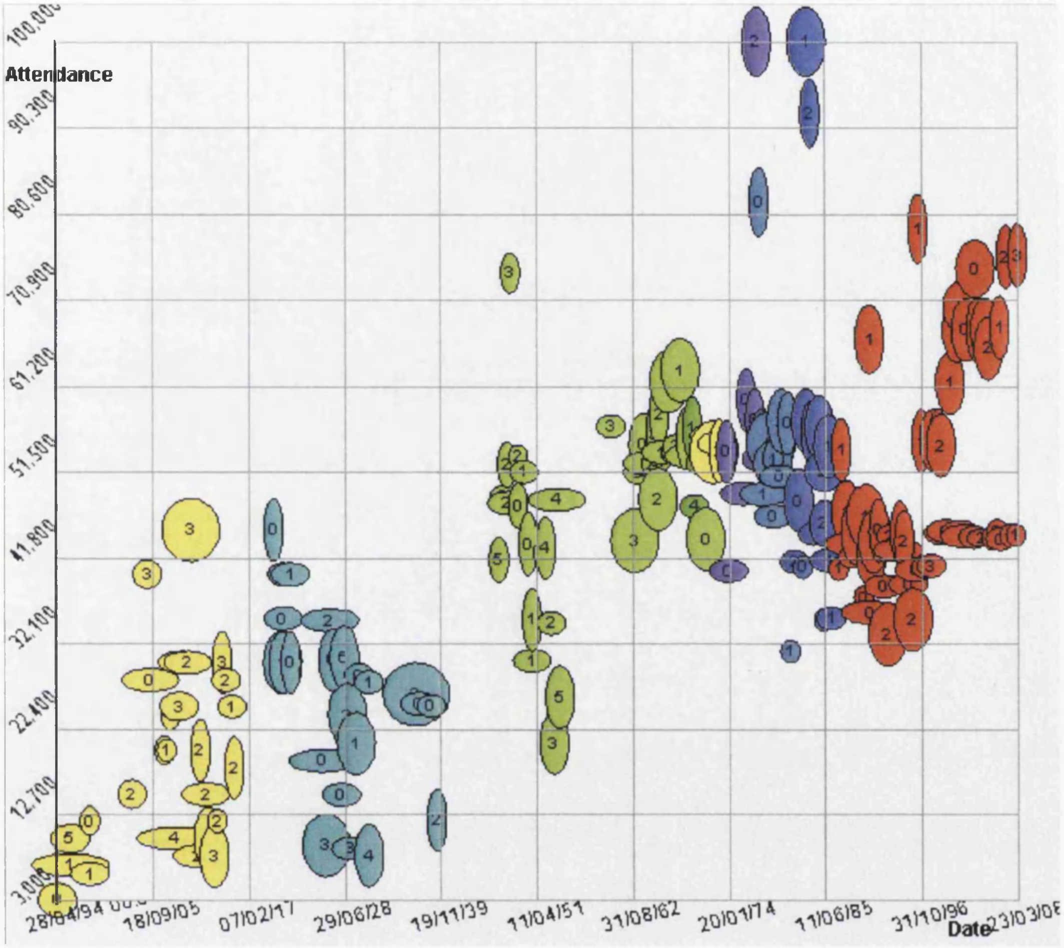


Figure 7.7: 2D graph visualization of Manchester United subject sub-domain, Manchester United versus Liverpool matches from Red11.org website with mappings: x to Date; y to Attendance; width to OTH; height to Venue; label to UTD; color to Manager. **Score = 83.67** (Middle ranking score)

Figure 7.9 - The highest scoring visualization for the Manchester United versus Liverpool page on the Red11.org website when the sub-set of source data entities is Date, WLD, Competition, Manager, HA, Venue (2D graph).

Result:. The visualization in Figure 7.6 has the most cognitive value when compared to the visualization in Figure 7.9 (as judged by the author). This corresponds to the score given by the SemViz scoring algorithm.

Conclusion: *Pass*

7.4.2.6 Same sub-domain, different websites

Test ID: s06

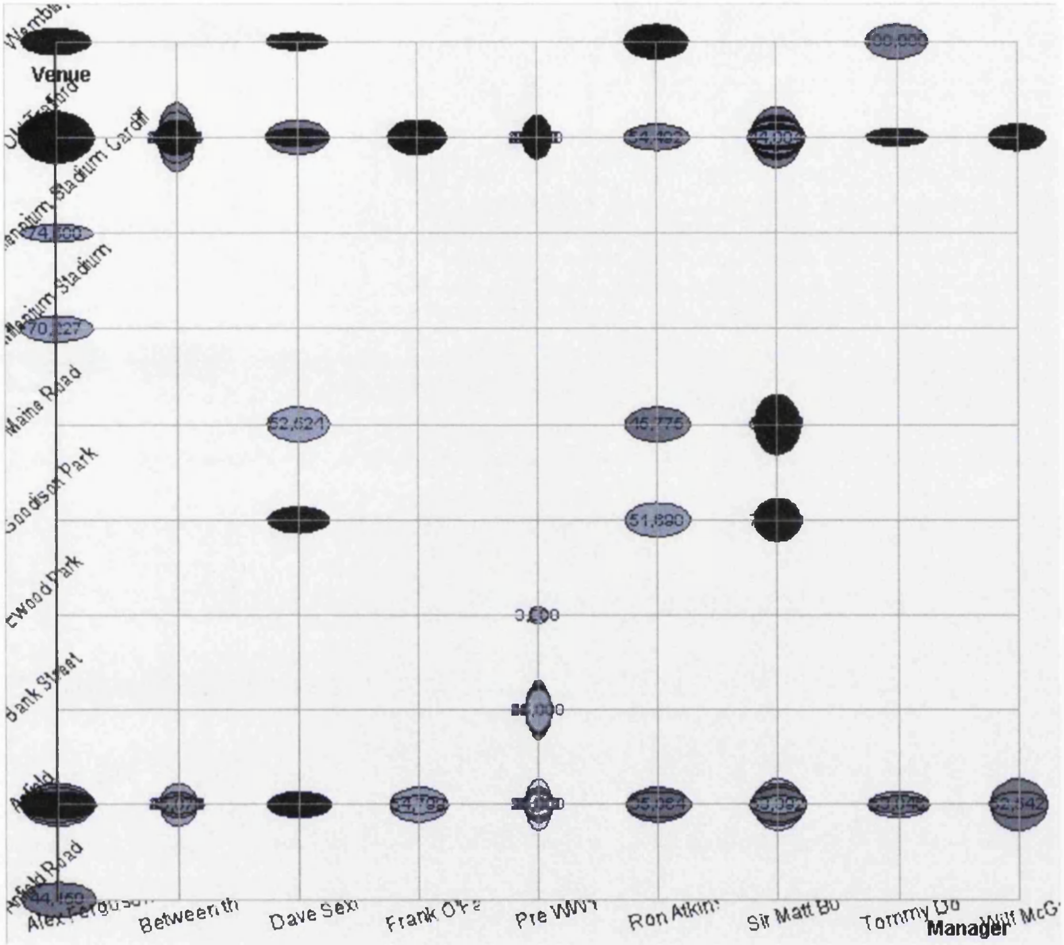


Figure 7.8: 2D graph visualization of Manchester United subject sub-domain, Manchester United versus Liverpool matches from Red11.org website with mappings: x to Manager; y to Venue; width to Date; height to UTD; label to Attendance; color to OTH. **Score = 77.17** (Lowest ranking score)

Test Description: We need to test the scalability of the Domain Ontology for a wide-breadth of subject area concepts within the overall umbrella domain (i.e., 4 different websites, 3 different sub-domains (Manchester United, UK Premiership, FIFA World Cup), 1 domain ontology (Football)). In this test, we consider the same sub-domain (e.g., Manchester United stats), different websites (e.g., ManUtd.com when DO is created from Red11.org)

Expected Result: Cognitively valuable visualization are still produced (and scored highly) for websites which were not considered in the modelling of the original Domain Ontology, but which have a sub-domain which was considered.

Visualizations Inspected: The original DO was modelled on the Red11.org website (Manchester United statistics) and Andrew's FIFA World Cup Statistics website (FIFA World Cup statistics). We will inspect the visualizations produced and scored highly from the ManUtd.com website. Figure 7.15 - The highest scoring visualization for the Results

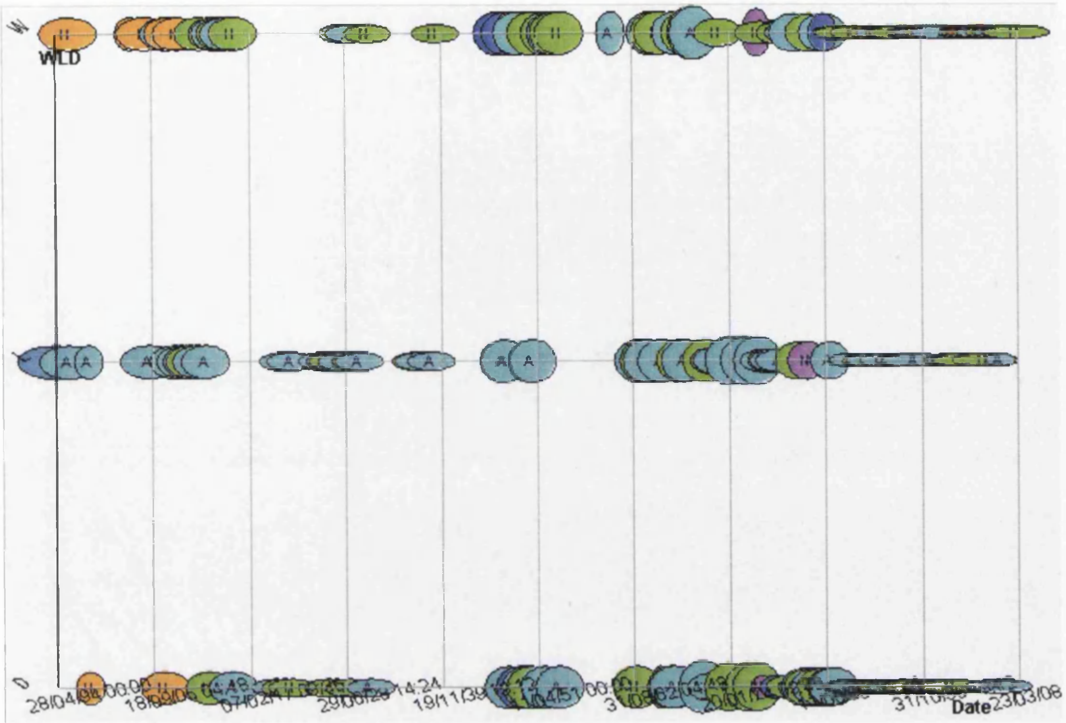


Figure 7.9: 2D graph visualization of Manchester United subject sub-domain, Manchester United versus Liverpool matches from Red11.org website with mappings: x to Date; y to Won/Lost/Drawn; width to Competition Name; height to Manager; label to Home/Away; color to Venue. **Score 90.99** (Highest ranking score but with a different subset of source data entities from figure 7.6. Notice that figure 7.6 has a higher cognitive value and indeed a higher score. See section 7.4.2.5 for further discussion.)

and Fixtures page on the ManUtd.com website.

Result:. The highest scoring visualization according to SemViz has a high degree of cognitive value as judged by the author.

Conclusion: *Pass*

7.4.2.7 Different sub-domain, different websites

Test ID: s07

Test Description: In this test, we consider a different sub-domain (e.g., A website from the UK Premiership sub-domain when DO is modeled from Manchester United stats and World Cup stats)

Expected Result: Cognitively valuable visualization are still produced (and scored highly) for websites with sub-domains which were not considered in the modelling of the original Domain Ontology.

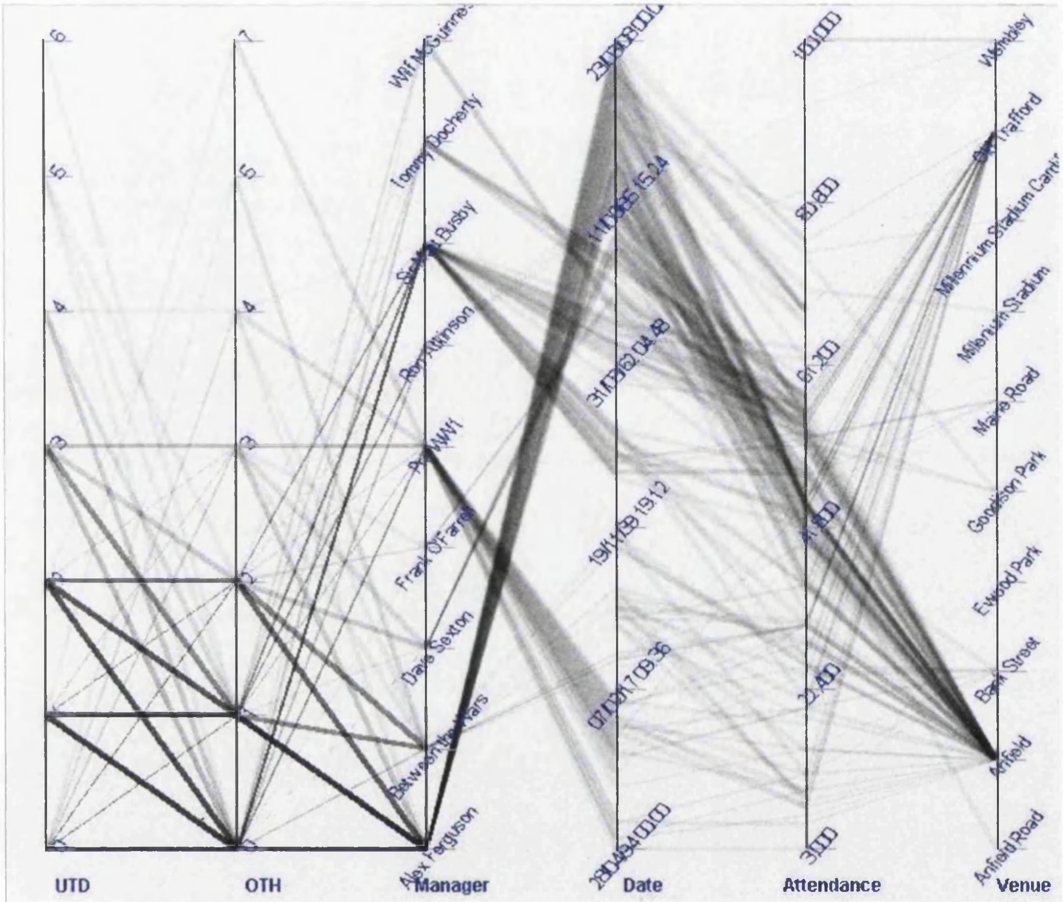


Figure 7.10: Parallel coordinates visualization of Manchester United subject sub-domain, Manchester United versus Liverpool matches from Red11.org website with mappings: Ypoint0 to UTD; Ypoint1 to OTH; Ypoint2 to Manager; Ypoint3 to Date; Ypoint4 to Attendance; Ypoint5 to Venue. **Score 80.01** (Highest ranking score for the same subset of source data entities as shown in figure 7.6. Notice that figure 7.10 has a lower cognitive value and indeed a lower score than figure 7.6. See sections 7.4.2.2 and 7.4.2.4 and for further discussion.)

Visualizations Inspected: The original DO was modelled on Manchester United statistics (the Red11.org website) and FIFA world cup statistics (Andrew's FIFA World Cup Statistics website). We will inspect the visualizations produced and scored highly from data in the UK Premiership sub-domain (football.co.uk website). Figure 7.14 - The highest scoring visualization for the Football Stats Man Utd 2007 - 2008 Season page on the Football.co.uk website.

Result:. The highest scoring visualization according to SemViz has a high degree of cognitive value as judged by the author.

Conclusion: *Pass*

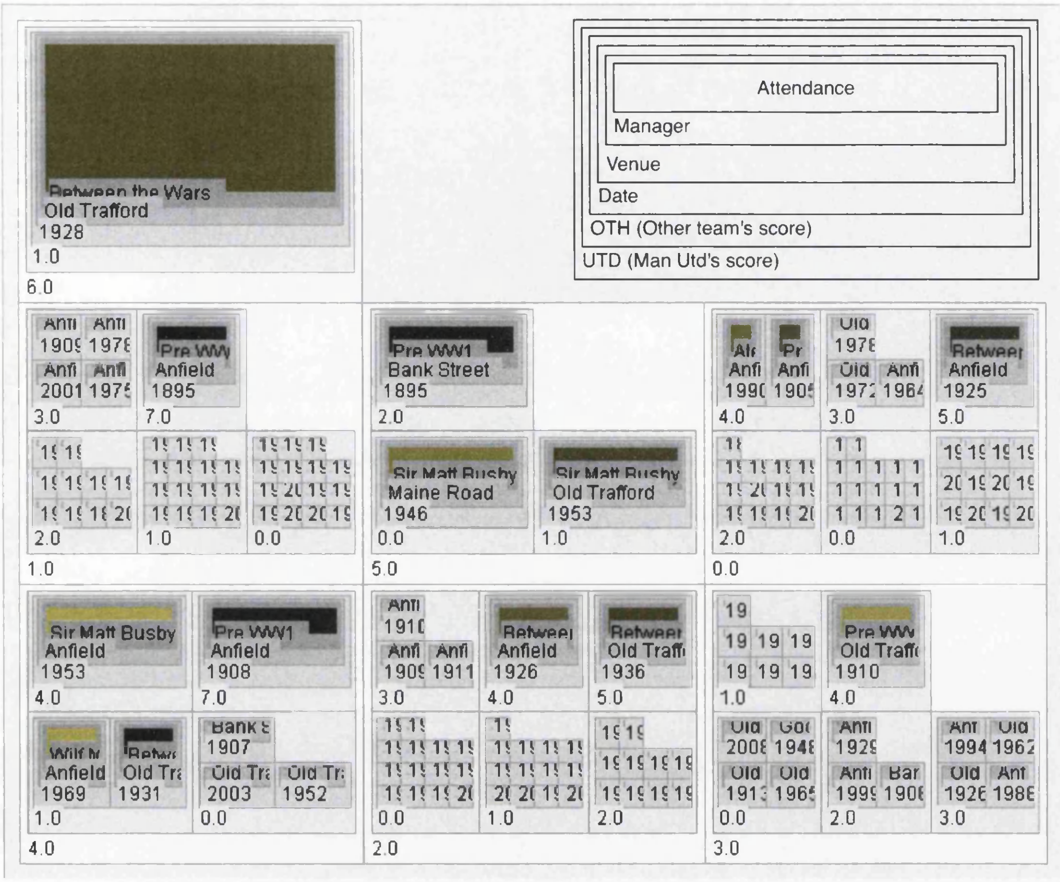


Figure 7.11: TreeMap visualization of Manchester United subject sub-domain, Manchester United versus Liverpool matches from Red11.org website with mappings: Level0 to UTD; Level1 to OTH; Level2 to Date; Level3 to Venue; Level4 to Manager; Color to Attendance. **Score 77.83** (Highest ranking score for the same subset of source data entities as shown in figure 7.6. Notice that figure 7.11 has a lower cognitive value and indeed score than figure 7.6. See sections 7.4.2.2, 7.4.2.4 and 7.7.1.3 for further discussion.)

7.5 User Testing : Approach

The purpose of the user testing is to mimic as closely as possible the scenario whereby a user wishes to perform ad-hoc visualization of a dataset on a webpage. An advantage which users of general purpose visualization packages have is that they usually have a specific idea of the visualization they have in mind. This leads them to being "experts" at reading their own visualization. With the SemViz interaction technique, a more ad-hoc approach is taken to providing the visualization (c.f. the Design Galleries approach as discussed in section 2.5.1.2). However, with Design Galleries, the user is iteratively led to the visualization. In contrast, SemViz gives the user a visualization immediately with no training or lead-in. It is important to assess the impact of this and ask if ad-hoc visualization still allows the user to gain cognitive insights into the data.

The user tests were conducted according to the principles of discount usability engineering

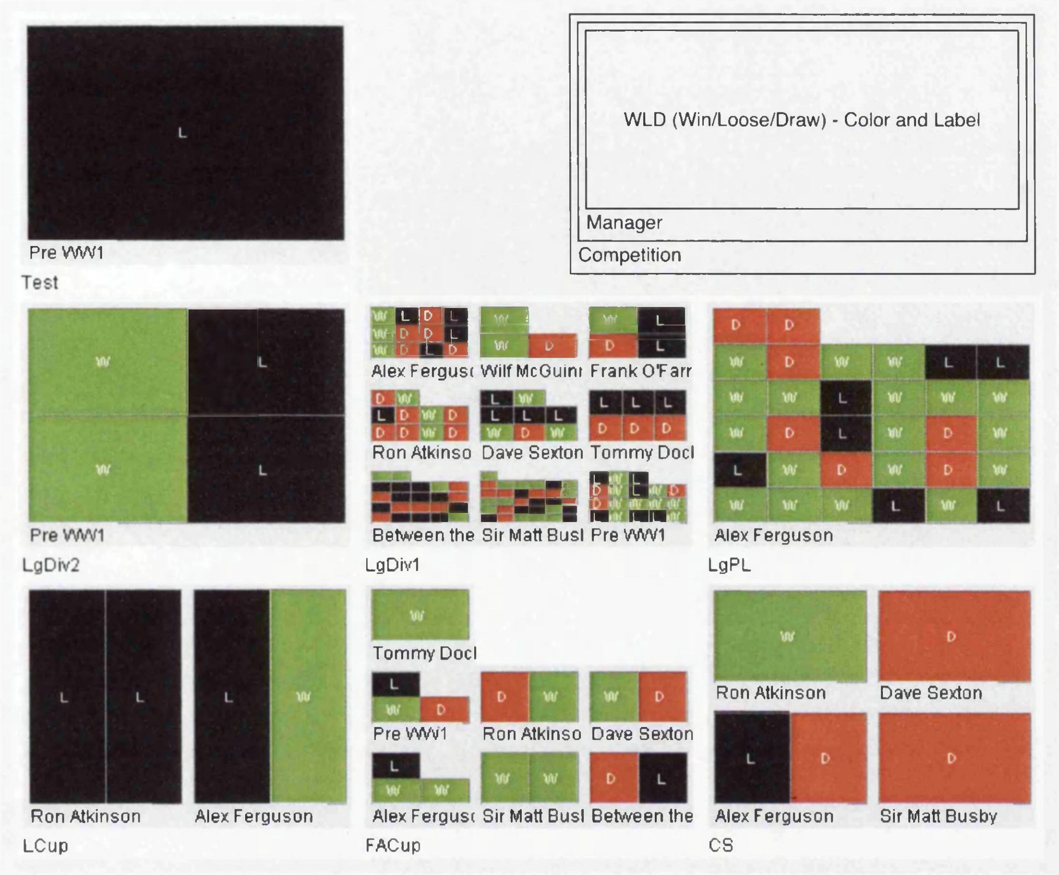


Figure 7.12: TreeMap visualization of Manchester United subject sub-domain, Manchester United versus Liverpool matches from Red11.org website with mappings: Level0 to Competition; Level1 to Manager; Color to Win/Loose/Draw; Label to Win/Loose/Draw. **Score 23.31** (Highest ranking score for a reduced subset of source data entities compared with figure 7.11. Notice that this visualization has a higher cognitive value, yet its score is lower than figure 7.11. The reasons for this are discussed in sections 7.4.22 and 7.7.1.3.)

[Nie95]. We aim to get the maximum insight into the validity of the SemViz approach with the minimal of cost (in terms of number of participants and the task length). The aim of this is to gain early feedback which can validate the SemViz approach and also inform further work.

7.5.1 Subject Preparation

Each user testing participant is given a brief description of the test environment, including the subject domain (football) and the reasons for conducting the tests. The following are also described:

1. Brief description of ILOG Discovery's interaction. User is told that they can move pointer over a visualization to get record specific information and to see all field

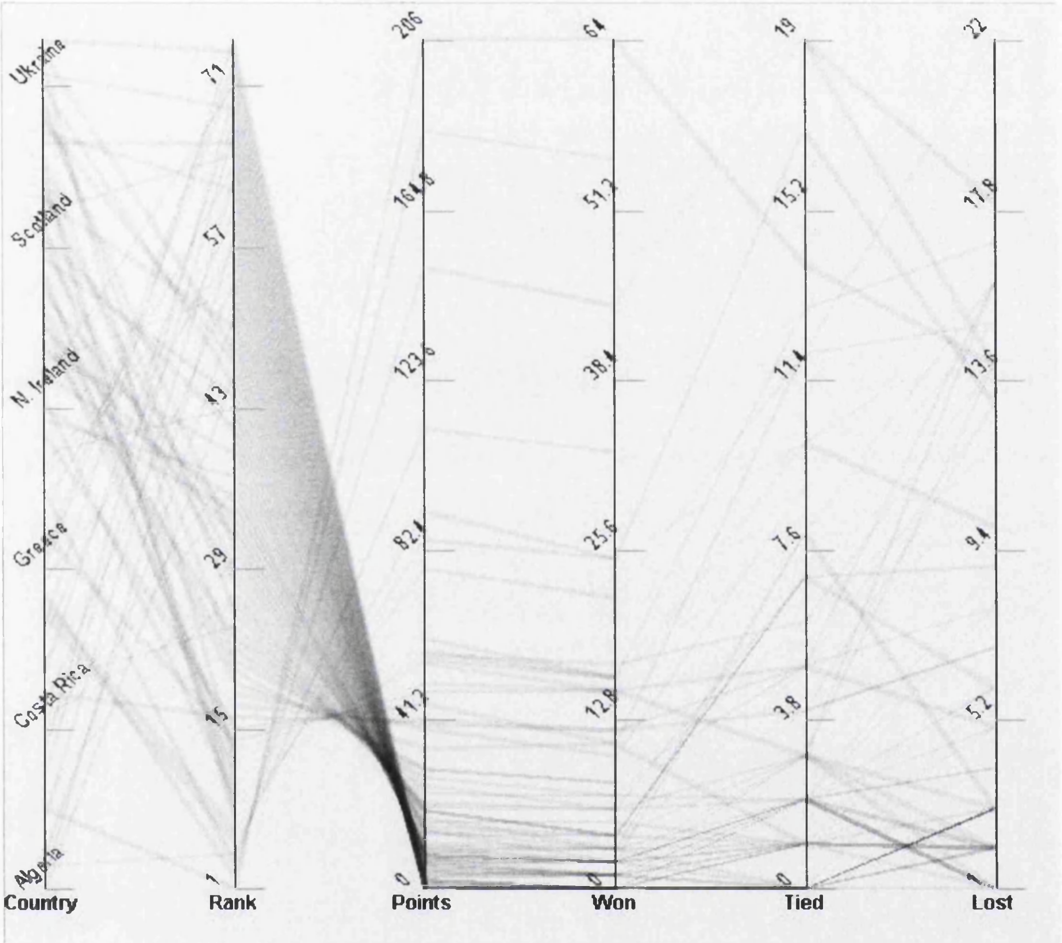


Figure 7.13: Parallel coordinates visualization of World Cup Statistics subject sub-domain, World Cup Finals Records by Country 1930-2006 from Andrew's FIFA World Cup Website with mappings: Ypoint0 to Country; Ypoint1 to Rank; Ypoint2 to Points; Ypoint3 to Won; Ypoint4 to Tied; Ypoint5 to Lost. **Score 82.36** (Highest ranking score for a Parallel Coordinates visualization of a different dataset compared to the previous visualizations. Note that this visualization scored higher than the equivalent highest scoring visualization using a 2D graph or TreeMap technique. See sections 7.4.2.1 and 7.7.1.2 for further discussion.)

values.

2. Description of 2D graph (what each parameter means).
3. Description of TreeMap (what each parameter means).
4. Description of Parallel Coordinates (what each parameter means).

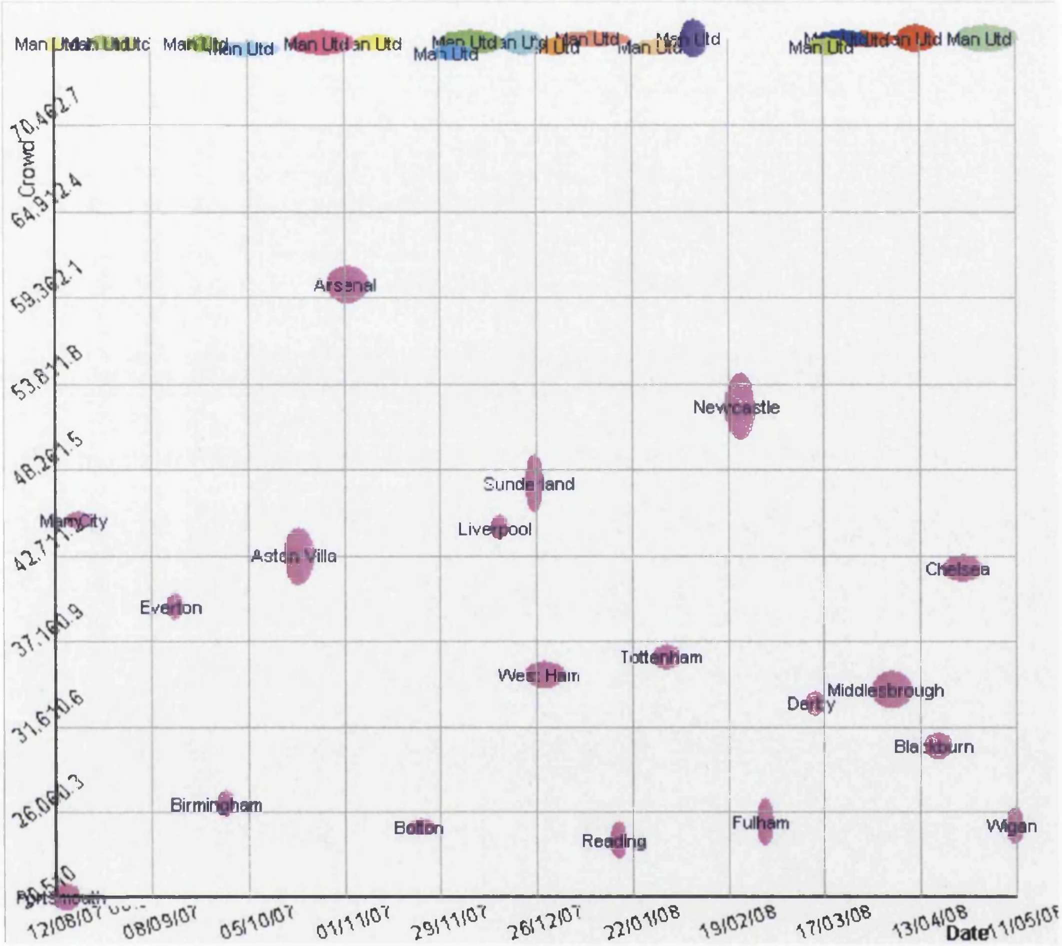


Figure 7.14: 2D Graph visualization of UK Premiership subject sub-domain, Results of Manchester United Matches 2007/2008 season from Football.co.uk Website with mappings: x to Date; y to Crowd; width to Home; height to Away; label to Local; color to Visitor. **Score 86.88** (Highest ranking score for a 2D graph visualization of a different dataset compared to the previous visualizations. Note that this visualization scored higher than the equivalent highest scoring visualization using a Parallel Coordinates or TreeMap technique. See section 7.4.2.7 for further discussion.)

7.5.2 Apparatus

1. Apple MacBook running Windows XP under the Parallels Virtual Machine.
2. ILOG Discovery version 20050321.
3. iSight recorder for capturing subject.
4. Screen recording software for recording subjects' actions.
5. Canon HF100 camcorder for recording whole experiment (subject and instructor).

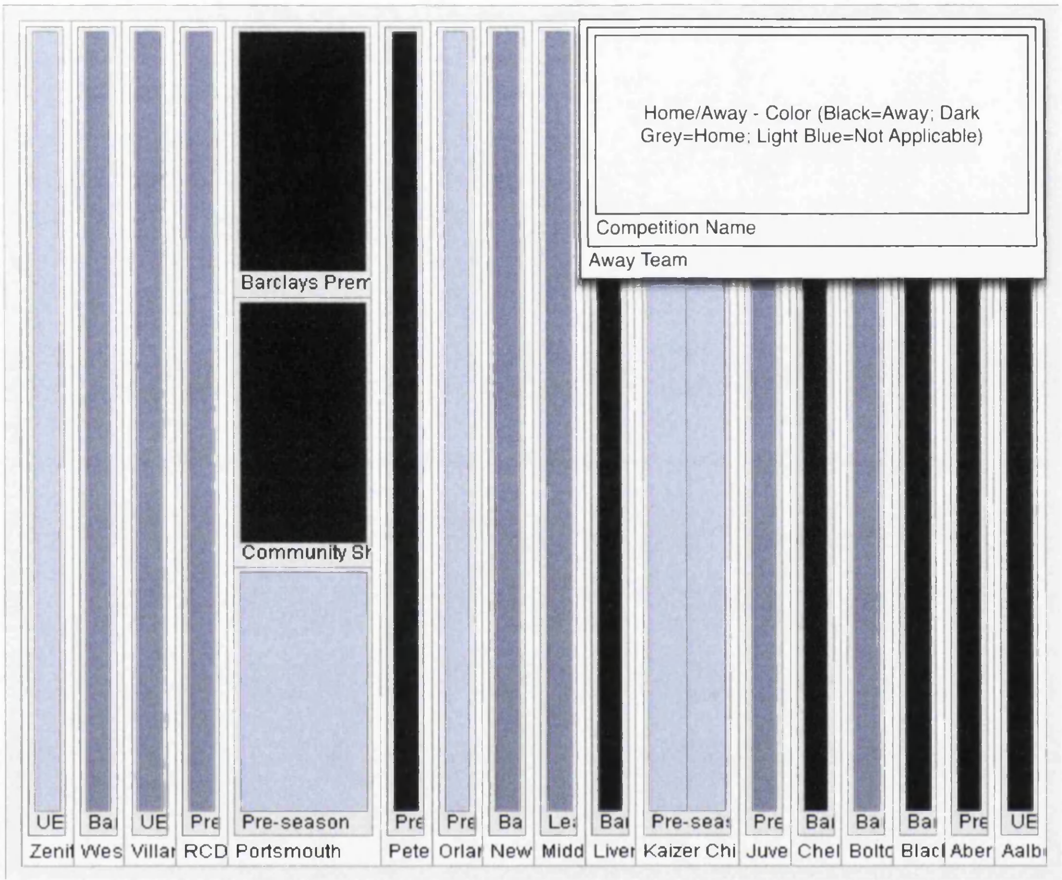


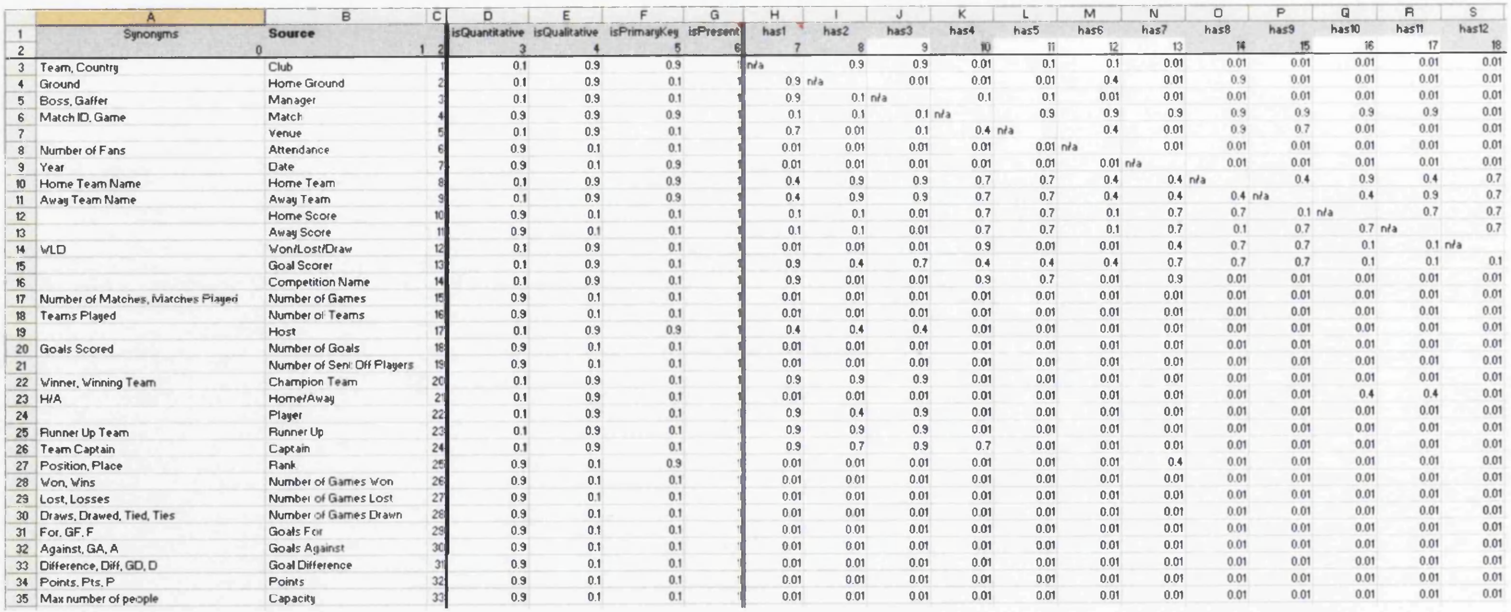
Figure 7.15: TreeMap visualization of Manchester United subject sub-domain, Manchester United Results and Fixtures from ManUtd.com website with mappings: Level0 to Away Team; Level1 to Competition Name; Color to Home/Away. **Score 23.61** (Highest ranking score for a TreeMap visualization of a different dataset compared to the previous visualizations. Note that this visualization scored higher than the equivalent highest scoring visualization using a 2D graph or Parallel Coordinates technique. See section 7.4.2.6 for further discussion.)

7.5.3 Subjects

The subjects chosen were all from a non-Computer Science background and certainly had no knowledge of Information Visualization techniques. This profile was deliberately chosen so as to mimic the target audience as described in section 1. We wish to validate SemViz as a tool to allow

general purpose information visualization for non-experts

The subject profiles are listed in table 7.6.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Synonyms	Source		isQuantitative	isQualitative	isPrimaryKey	isPresent	has1	has2	has3	has4	has5	has6	has7	has8	has9	has10	has11	has12
2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
3	Team, Country	Club	1	0.1	0.9	0.9	1	n/a	0.9	0.9	0.01	0.1	0.1	0.01	0.01	0.01	0.01	0.01	0.01
4	Ground	Home Ground	2	0.1	0.9	0.1	1	0.9	n/a	0.01	0.01	0.01	0.4	0.01	0.9	0.01	0.01	0.01	0.01
5	Boss, Gaffer	Manager	3	0.1	0.9	0.1	1	0.9	0.1	n/a	0.1	0.1	0.01	0.01	0.01	0.01	0.01	0.01	0.01
6	Match ID, Game	Match	4	0.9	0.9	0.9	1	0.1	0.1	0.1	n/a	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.01
7		Venue	5	0.1	0.9	0.1	1	0.7	0.01	0.1	0.4	n/a	0.4	0.01	0.9	0.7	0.01	0.01	0.01
8	Number of Fans	Attendance	6	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	n/a	0.01	0.01	0.01	0.01	0.01	0.01
9	Year	Date	7	0.9	0.1	0.9	1	0.01	0.01	0.01	0.01	0.01	0.01	n/a	0.01	0.01	0.01	0.01	0.01
10	Home Team Name	Home Team	8	0.1	0.9	0.9	1	0.4	0.9	0.9	0.7	0.7	0.4	0.4	n/a	0.4	0.9	0.4	0.7
11	Away Team Name	Away Team	9	0.1	0.9	0.9	1	0.4	0.9	0.9	0.7	0.7	0.4	0.4	0.4	n/a	0.4	0.9	0.7
12		Home Score	10	0.9	0.1	0.1	1	0.1	0.1	0.01	0.7	0.7	0.1	0.7	0.7	0.1	n/a	0.7	0.7
13		Away Score	11	0.9	0.1	0.1	1	0.1	0.1	0.01	0.7	0.7	0.1	0.7	0.1	0.7	0.7	n/a	0.7
14	WLD	Won/Lost/Draw	12	0.1	0.9	0.1	1	0.01	0.01	0.01	0.9	0.01	0.01	0.4	0.7	0.7	0.1	0.1	n/a
15		Goal Scorer	13	0.1	0.9	0.1	1	0.9	0.4	0.7	0.4	0.4	0.4	0.7	0.7	0.7	0.1	0.1	0.1
16		Competition Name	14	0.1	0.9	0.1	1	0.9	0.01	0.01	0.9	0.7	0.01	0.9	0.01	0.01	0.01	0.01	0.01
17	Number of Matches, Matches Played	Number of Games	15	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
18	Teams Played	Number of Teams	16	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
19		Host	17	0.1	0.9	0.9	1	0.4	0.4	0.4	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
20	Goals Scored	Number of Goals	18	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
21		Number of Sent Off Players	19	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
22	Winner, Winning Team	Champion Team	20	0.1	0.9	0.1	1	0.9	0.9	0.9	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
23	H/A	Home/Away	21	0.1	0.9	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.4	0.4	0.01
24		Player	22	0.1	0.9	0.1	1	0.9	0.4	0.9	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
25	Runner Up Team	Runner Up	23	0.1	0.9	0.1	1	0.9	0.9	0.9	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
26	Team Captain	Captain	24	0.1	0.9	0.1	1	0.9	0.7	0.9	0.7	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
27	Position, Place	Rank	25	0.9	0.1	0.9	1	0.01	0.01	0.01	0.01	0.01	0.01	0.4	0.01	0.01	0.01	0.01	0.01
28	Won, Wins	Number of Games Won	26	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
29	Lost, Losses	Number of Games Lost	27	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
30	Draws, Drawed, Tied, Ties	Number of Games Drawn	28	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
31	For, GF, F	Goals For	29	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
32	Against, GA, A	Goals Against	30	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
33	Difference, Diff, GD, D	Goal Difference	31	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
34	Points, Pts, P	Points	32	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
35	Max number of people	Capacity	33	0.9	0.1	0.1	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Figure 7.16: An extract from the Football DO. All thirty three concepts are shown with their relationships for "is" attributes and and the first twelve has relationships.

ID	Source Concept
1	Club
2	Home Ground
3	Manager
4	Match
5	Venue
6	Attendance
7	Date
8	Home Team
9	Away Team
10	Home Score
11	Away Score
12	Won/Lost/Draw
13	Goal Scorer
14	Competition Name
15	Number of Games
16	Number of Teams
17	Host
18	Number of Goals
19	Number of Sent Off Players
20	Champion Team
21	Home/Away
22	Player
23	Runner Up
24	Captain
25	Rank
26	Number of Games Won
27	Number of Games Lost
28	Number of Games Drawn
29	Goals For
30	Goals Against
31	Goal Difference
32	Points
33	Capacity

Table 7.5: The Football Domain Ontology Concept Names.

7.6 User Testing : Closed Tasks

It is difficult to evaluate any system without gaining solid metrics on its performance. With user testing, it is necessary to set tasks which gain objective results. In this set of tasks we measure the time taken for users to perform a set of well-defined, closed tasks.

Task ID c1

Overview Give subjects specific tasks with a 2D graph and time how long it takes them to complete.

ID	Gender	Age	Occupation	English proficiency
1	Female	28	Teacher	Advanced
2	Female	32	Accountant	Upper Intermediate
3	Female	38	Secretary	Lower Intermediate
4	Female	46	Secretary	Lower Intermediate
5	Male	32	Engineer	Upper Intermediate
6	Male	31	Banker	Native
7	Female	30	Student	Native
8	Male	60	Retired	Native
9	Female	28	Student	Native

Table 7.6: Profiles of each of the User Testing subjects.

Purpose of task Find out how well the SemViz scoring algorithm and ad-hoc visualization works for 2D graphs.

Procedure

1. Show the subject the source web page - Manchester United versus Liverpool results from Red11.org (see figure 7.1).
2. Give subjects an opportunity to ask questions about the webpage. (To help clarify field names, and allow subjects to familiarize themselves with the number of records and number of fields).
3. Subject is shown the visualization (see figure 7.6).
4. Subject is given the questions c1.1 to c1.5 (see below).
5. Record whether the subject gives the correct answer and how long it takes.
6. Subject is given three attempts to find the correct answer.

Questions

- **c1.1** What is the trend of the attendance against time?
- **c1.2** On which date was the highest scoring draw?
- **c1.3** Give me the dates of 5 matches when a high number of goals were scored by both teams.
- **c1.4** Give me the dates of 5 matches when one team scored many more goals than the other team.
- **c1.5** Can you see any anomalies (strange exceptions in the data)? On which date did they occur? Why do you think they occurred?

Task ID c2

Overview Give subjects specific tasks with a Parallel Coordinates visualization and time how long it takes them to complete.

Purpose of task Find out how well the SemViz scoring algorithm and ad-hoc visualization works for Parallel Coordinates.

Procedure 1. Tell the subject that the next visualization is based on the same dataset (from the web page shown in figure 7.1).

2. Subject is shown the visualization (see figure 7.10).

3. Subject is given the questions c2.1 to c2.4 (see below).

4. Record whether the subject gives the correct answer and how long it takes.

5. Subject is given three attempts to find the correct answer.

Questions

- **c2.1** For what duration has Alex Ferguson been manager?
- **c2.2** Which score draws are most frequent?
- **c2.3** Can you see any periods in time when no matches were played? When were these periods?
- **c2.4** Can you see any patterns with the number of goals scored?

Task ID c3

Overview Give subjects specific tasks with a TreeMap and time how long it takes them to complete.

Purpose of task Find out how well the SemViz scoring algorithm and ad-hoc visualization works for TreeMap.

Procedure 1. Tell the subject that the next visualization is again based on the same dataset (from the web page shown in figure 7.1).

2. User is shown the visualization (see figure 7.11).

3. Subject is given the questions c3.1 to c3.4 (see below).

4. Record whether the subject gives the correct answer and how long it takes.

5. Subject is given three attempts to find the correct answer.

Questions

- **c3.1** How many managers have there been while Manchester United have been in the Premier League (LgPL)? What is/are their name(s)?
- **c3.2** How many wins, losses and draws have Manchester United had against Liverpool in the Premier League? Would you say Manchester United have done well, ok or badly against Liverpool in the Premier League?
- **c3.3** How many wins, losses and draws have Manchester United had against Liverpool in the FA Cup? Would you say Manchester United have done well, ok or badly against Liverpool in the FA Cup?
- **c3.4** Under which Manager and in which Competition have Manchester United not done very well?

7.6.1 Expected Answers

The task id and expected answers are shown below:

c1.1 Attendance has gone up over time.

c1.2 22-08-58.

c1.3 Any five dates from 02-11-1895, 25-03-1908, 19-02-1910, 22-08-1953, 04-04-1988, 10-11-1962.

c1.4 Any five dates from 19-09-1925, 01-11,1913, 05-05-1928, 19-12-1953, 12-04-1952, 13-12-1969, 26-12-1978, 05-04-2003.

c1.5 The red ellipse representing Anfield Road should be Anfield (probably a data entry error in the original website). The two ellipses with unique colours representing Millennium Stadium and Millennium Stadium Cardiff (should have the same name). The match on 24-01-1948 with a very high attendance of 74,000 for its time (maybe a data entry error).

c2.1 From 26-12-1986 to present.

c2.2 0-0, 1-1, 2-2.

c2.3 Any three dates from 02-04-1915 to 01-01-20, 24-12-1921 to 10-03-1926, 03-04-1931 to 21-11-1936, 06-05-1939 to 11-09-1946, 19-12-1953 to 30-01-1960.

c2.4 Manchester United seem to have had more wins (because the lines always point down from UTD to OTH). Also, when there is a really high score, it seems Liverpool are the scorers (and winners).

c3.1 One - Alex Ferguson.

c3.2 18 wins, 7 losses, 7 draws.

c3.3 8 wins, 3 losses, 4 draws.

c3.4 Ron Atkinson in the League Cup, Alex Ferguson in the Charity Shield.

7.6.2 Results and Remarks

The time taken for completing each of the tasks are shown in table 7.7. This is shown visually in figure 7.17 and figure 7.18. Where a task was not completed, or the correct answer was not given after three attempts, the value is empty.

In most visualization tasks, users have a set goal. They know with a high degree of certainty the information they are looking for. With SemViz, users are presented with a visualization and may not have a set goal. Their goal may be as open as getting a summary or even finding out "something interesting". For this reason, each subject was given the opportunity to comment on each of the three visualizations. These comments provide addition information on which to assess the effectiveness of SemViz.

Subject ID	1	2	3	4	5	6	7	8	9
c1.1	240	21	115	60	12	2	2	6	4
c1.2	5	104	105		11	5	6	14	11
c1.3	73	65	135		30	23	19	44	43
c1.4	24	41	102		29	22	18	28	30
c1.5	3	58	85		59	49	21	31	12
c2.1	12	20	122		27	8	9	15	18
c2.2	64	77	103		35	5	9	4	38
c2.3	11	10	130		48	32	39	47	24
c2.4	44				67	49	20	25	59
c3.1	2	9	7		1	1	2	4	3
c3.2	13	23	34		9	11	9	24	15
c3.3	18	45	20		8	9	10	23	18
c3.4	18	66	80		12	8	15	19	8

Table 7.7: The number of seconds it took each subject to perform each task.

Below, we summarise how each subject performed in the tasks and compare this to their profile (age, occupation, gender and English-language proficiency). We also list any comments the subjects made about the visualizations and the tasks themselves.

Subject 1 This subject, although her native language was not English, had a high degree of proficiency. Therefore, the language of the questions and tasks did not seem to hinder her progress. The timing of her tasks is very similar to that of the native speakers (subjects 6 to 9) except for task c1.1 where the subject took the longest time of all subjects. From observing the subject performing this task it seems that she overestimated the complexity of the task and was trying to give a more indepth

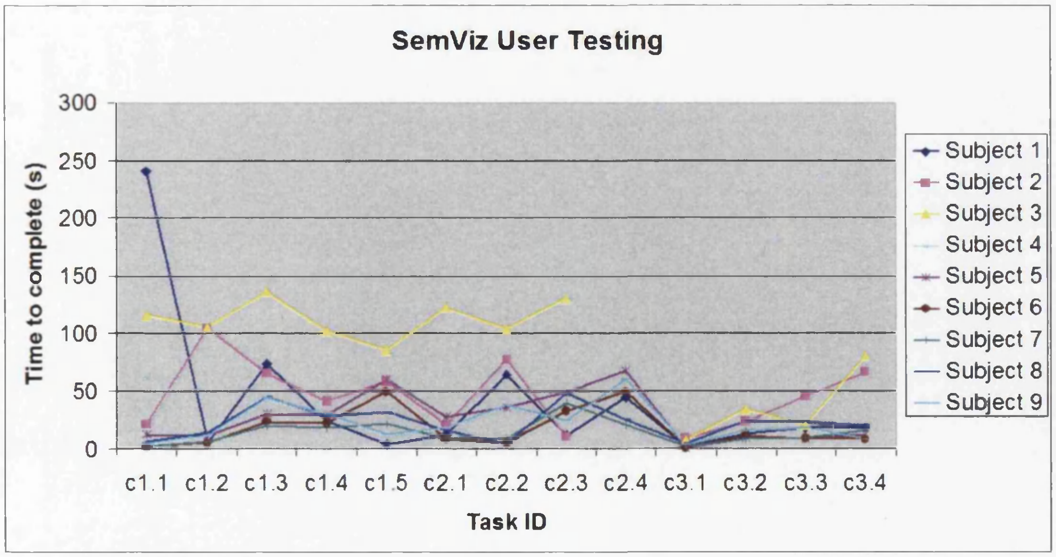


Figure 7.17: Results of SemViz User Testing with 9 subjects and 13 tasks.

answer than what was required.

Subject 2 Stated that one reason she had trouble interpreting the visualizations was because she had very little interest in the area of football, in particular the competition names (Premier League, FA Cup etc.) were alien to her. Also stated that Parallel Coordinates were complicated and less understandable than 2D graphs or TreeMaps. However, the subject completed the tasks in a reasonable time except for task c2.4 where the subject was asked to spot any patterns in the number of goals scored. The subject did not understand the question and when it was explained, she did not give any answer.

Subject 3 This subject had a low degree of English language proficiency when compared to the other non-native speakers. However, the subject completed all of the tasks (except c2.4 again), but the duration was longer. Preferred 2D graphs and Parallel Coordinates over TreeMaps. Stated that she liked the idea of Parallel Coordinates and could see how they might be helpful in her work. Task time was slowed due to unfamiliarity with the track-pad on the laptop used.

Subject 4 In the 2D graph (figure 7.6) the subject couldn't equate the size of the shape with the number of goals scored. The subject was having problem with understanding the questions also. The only task where the correct answer was given was task c1.1. After attempting all other tasks up to c1.4, the test was abandoned as it was felt unlikely that any further success would be gained. Task time was slowed due to unfamiliarity with the track-pad on the laptop used.

Subject 5 This subject had a good English language proficiency level and stated that he had an interest in football. He completed all the tasks in a good time except task c2.4 where he took longer than average. This seemed to be because the subject was trying to find anomalies which weren't obvious (e.g., patterns in the number of goals scored compared to attendance).

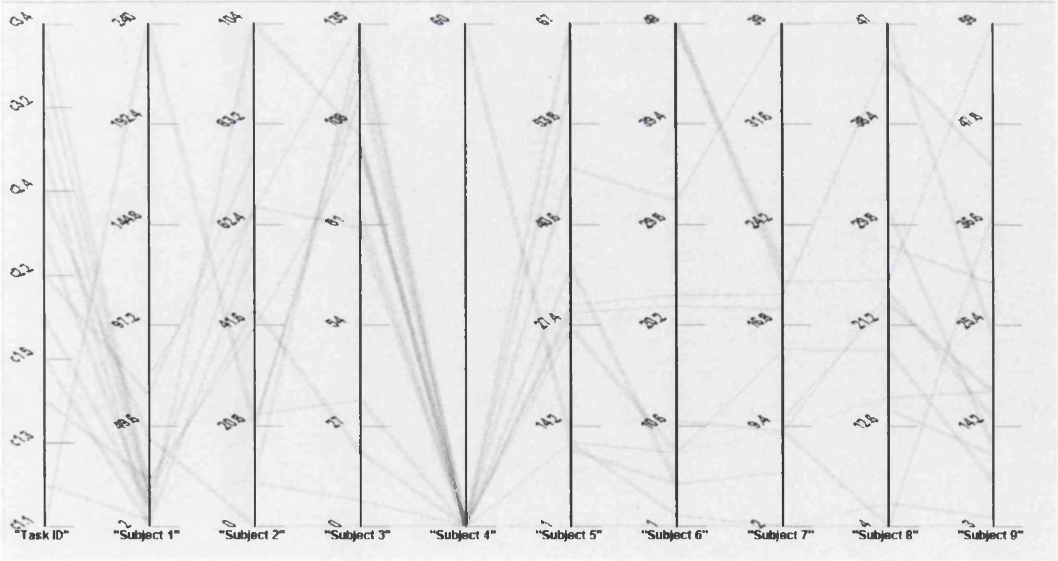


Figure 7.18: Results of SemViz User Testing - Parallel Coordinates.

Subject 6 This subject completed many of the tasks very quickly. The subject was a native speaker, from a numerate discipline and had an interest in football. Again task c2.4 seemed to take longer than average to complete and it seems that the reason for this was the same - the subject was trying to find in-depth anomalies which did not exist (or were not one of the expected results).

Subject 7 This subject stated that she had no real interest in football. However, the tasks were all completed in a short time. The subject was very focussed on the tasks and she did not try to find any complicated or in-depth answers. This is reflected in the low time for task c2.4.

Subject 8 This subject was the oldest of all subjects tested. He had an interest in football and had a numerate background. However, some of the tasks took longer than average (particularly the Parallel Coordinates) due to the subject's unfamiliarity with the track-pad and poor eye-sight.

Subject 9 This subject stated that she had no interest in football. As such, some of the terms (e.g., "score draw") had to be explained to the subject. Again, task c2.4 took a long time to complete. However, this seemed to be due to the subject's inability to identify anomalies rather than trying too hard to identify complex anomalies which weren't present (like subjects 5 and 6).

After analysing all results we rationalise the findings into 6 key points:

English proficiency For subjects who were native speakers, Advanced, or Upper-Intermediate level, there was not a significant difference in task completion time. However, the two subjects who were at a Lower-Intermediate level had significantly different results. Subject 3 completed the majority of tasks, but at slower pace. Subject 4 was only able to complete one task of the 13 tasks set. Therefore, we can conclude that performing user testing on non-native speakers is a valid and useful exercise as long as their English-language proficiency is Upper-Intermediate or above.

Domain Knowledge Adhering to a common stereotype, the male subjects said they had an interest in the subject domain (football), while all female subjects said that they weren't particularly interested. This seems to have had an effect on the time taken to complete tasks. For most tasks, the time seems to have been reduced. Interestingly, for task c2.4 (finding anomalies) the male subjects took more time. This seemed to be related to "trying too hard" to find anomalies which weren't present.

Parallel Coordinates Usability Three of the subjects had problems selecting individual lines (records) when using the Parallel Coordinates. This added time to some tasks which wasn't related to the subjects' cognition speed.

Track-Pad Usability Three of the subjects had not used a laptop's Track-Pad before participating in the user testing. Again, this resulted in increased task time which was not related to cognition speed.

Design of Task c2.4 This task asked subjects to find anomalies in the data. For some subjects, the task proved to be very straightforward. For others, it took a long time to complete. This was mainly due to different interpretations of the task by different subjects. This points to the task being poorly designed. If the user testing exercise were to be conducted again, the author would make this question more specific in order to avoid misinterpretation.

Validity of SemViz Except for subject 3, all subjects gained cognitive insight into the data on the original webpage. The three visualizations used were the highest scoring ones as produced by SemViz. Our user testing therefore shows that SemViz has benefit for non-expert visualization users.

7.7 User Testing : Expert Visual Inspection

In this series of tests, we asked an experienced Computer Scientist with expert knowledge of both information visualization and the subject domain of football to inspect the highest ranking visualizations of different input parameters and comment on their cognitive value. We asked the expert user to evaluate the best visualization based on:

1. Mapping permutation between source and target entities.
2. Sub-set selection of all the source entities.
3. Highest scoring visualization technique for that dataset and source entity sub-set.

The evaluation gives purely subjective results. However, it gives an indication of how the Domain Ontology and Visual Representation ontology could be adjusted to gain better results.

7.7.1 Evaluation and Results

7.7.1.1 2D graph

The expert user was asked to evaluate the following 2D graph visualizations:

Figure 7.6 A 2D Graph from the Manchester United versus Liverpool webpage from the Red11.org website. The sub-set of the source entities is: Date; Attendance; Away Score; Home Score; Manager; Venue. (Score 92.17).

Figure 7.9 As above, but the sub-set of the source entities is: Date; Won/Lost/Drawn; Competition Name; Manager; Home/Away; Venue.(Score 90.99).

The expert user stated that the first visualization was more cognitively useful based on the sub-set of the source entities used. It was also stated that it would be better to model Venue against Competition to see any patterns or anomalies. For instance, Wembley Stadium hosts FA Cup, League Cup and Charity Shield matches, but never hosts any Premier League games. This piece of knowledge would be represented in the Domain Ontology (DO) as a higher value for the *complements* relationship between **Venue** and **Competition**. Alternatively, the Semantic Bridge Ontology (SBO) could be employed to record a Semantic Bridge between the **Venue** and **Competition** concepts with a high *appropriateness* value (greater than 100).

7.7.1.2 Parallel Coordinates

The expert user was asked to evaluate the following Parallel Coordinates visualizations:

Figure 7.10 A Parallel Coordinates visualization from the Manchester United versus Liverpool webpage from the Red11.org website. (Score 80.01).

Figure 7.13 A Parallel Coordinates visualization from the World Cup Finals Records by Country 1930-2006 webpage from the Andrew's FIFA World Cup website. (Score 82.36).

The expert user stated that the first Parallel Coordinates visualization had a high degree of cognitive value and would be unlikely to benefit from further adjustment. However, it was noted that the second Parallel Coordinates visualization has an inverse relationship between **Rank** and **Points** (i.e., the lower the value of the Rank, the higher the value of the Points). Therefore, it was suggested that the Parallel Coordinates scale of either Rank or Points be inverted. This is a feature available for each data entity in ILOG Discovery. This piece of semantic knowledge could be captured within the Domain Ontology as a *reverse complements* relationship, instead of a *complements* relationship. This relationship type does not currently exist, but it could be added along with other semantic equivalences.

7.7.1.3 TreeMap

The expert user was asked to evaluate the following TreeMap visualizations:

Figure 7.11 A Parallel Coordinates visualization from the Manchester United versus Liverpool webpage from the Red11.org website. (Score 77.83)..

Figure 7.12 A Parallel Coordinates visualization from the World Cup Finals Records by Country 1930-2006 webpage from the Andrew's FIFA World Cup website. (Score 23.31).

The expert user reiterated the fact that TreeMap visualizations are most effective when a low number of parameters are used (around 2 to 4 levels, a colour, and a label). This is clear when comparing the two visualizations.

The expert user also commented that it is often useful to have multiple methods of visualising a particular source data entity. For example, it is beneficial to add text labels to rectangular areas to re-emphasise what the colour of each rectangle means. Therefore, the colour allows the user to see the visualization at overview level while the label serves as a useful reminder at an individual record level.

7.8 Summary

At the beginning of this chapter, we set out to evaluate the value of the SemViz pipeline, algorithm and interaction methodology using two methods: Scalability Evaluation; and User Testing. The purpose of scalability evaluation was to judge the effectiveness of SemViz by applying it to a case study which was larger than the examples given in the initial description of the system in chapter 6. The results of the scalability evaluation show that the approach can be applied to a larger example.

The major findings of the Scalability and Validity Evaluation were:

Domain Ontology Creation The football domain ontology has 33 source concepts. This results in 4,389 relationships between the concepts of the domain ontology's fully-connected graph. It took the author 3 days to enter valid numbers for this domain ontology. Clearly this is a large amount of time which would be unacceptable in most scenarios. Therefore, this presents the need for automated tools and more rapid methodologies for the creation of domain ontologies.

Pipeline Stages The majority of this work has focussed on the pipeline stage when the Domain Ontology (DO) is mapped to the Visual Representation Ontology (VRO). This is done using the SemViz scoring and ranking algorithm. The previous pipeline stage concerns the mapping between the Web Page (WP) and the Domain Ontology (DO). Since this is an area of existing Computer Science research, we have not focussed our attention on this. However, from the case study and evaluation we have conducted in this chapter, we can see that further work is needed in integrating current research techniques into this stage of the SemViz pipeline.

Automaticity - Pipeline While the objective of this work has been to achieve an automatic knowledge-driven visualization pipeline, there are some aspects of the process which are not automatic. As mentioned in section 6.2, the process of presenting SemViz with the data from the source web page must be performed using a CSV formatted file. The user must use a screen-scraper as most web pages do not provide their data in this format. Depending on the tool and the exact nature of the source webpage, the amount of manual interaction involved will vary.

Automaticity - Ontologies The ontologies involved in the SemViz pipeline have varying degrees of automaticity. There needs to be a DO for every subject domain which the SemViz tool is capable of handling. The process of creating the DO is a manual one and as such it is time consuming (see section 7.4.1). In section 8.3 we discuss further work which would address these shortcomings through automatic ontology generation. The VRO must also be manually created. However, this is a far less expensive task because only one VRO is required for each visualization technique. This is a one-off task and as such it can be considered as part of the development of the SemViz tool. When additional visualization techniques are added to the SemViz tool a new VRO would need to be created. Finally, the SBO is an ontology whose creation is totally automated. The SBO is created, and updated as the SemViz tool is used.

The purpose of the User Testing was to judge the effectiveness of the visualizations produced from SemViz by an independent group of third-party users who fit the profile of non-expert visualization users. The results of the user testing gave generally positive results, showing the effectiveness of the SemViz approach. It also provides some interesting findings which set some questions for future research.

The major findings of the User Testing were:

Reducing Visualization Complexity Our SemViz algorithm assumes that the user wishes to visualise all source entities which can be mapped to available target artefacts. This assumption provides good results for 2D graphs and Parallel Coordinates where the number of records does not significantly change the readability of the visualization (see figures 7.6 and 7.10). However, for TreeMaps, where the number of records significantly alters the expressability of the visualization, there needs to be a trade-off between the number of levels used (Level0, Level1, ... , LevelN) and the number of records. SemViz does not currently take this into account which means that some overly complicated visualizations are scored highly (see figure 7.11). This visualization gave poor results for test subjects. Manual intervention here resulted in a similar, more readable visualization (see figure 7.12) which usability test subjects found easier to read and less confusing.

Proportional Scoring Currently, the SemViz algorithm scores visualizations based on the number of source data entities and target representation artefacts which are mapped, together with all the relationship types which are considered. This Absolute Scoring approach results in visualizations with a higher number of visual artefacts scoring higher, even though their cognitive value may be reduced due to over complexity. If we change the SemViz algorithm so that the score is calculated as a proportion of the maximum score, then it is valid to compare scores between visualizations which consider a different number of source data entity to target representation artefact mappings. This is pertinent to the TreeMap example given above.

Refining Visualization Techniques A useful technique for aiding the readability of both 2D graphs and TreeMaps is to repeat information given by colour encoding with a text label. For example, in the TreeMap in figure 7.12 we show the Win/Lost/Draw source entity using the rectangle colour (red being a loose, black being a draw and green being a win). In order for a user to know what the colour encoding is, they either need to use ILOG Discovery interactively where they can hover their mouse pointer over shapes in order to find out the values of specific fields, or they need to be told the mapping in advance with a lookup table. After usability inspection by an advance user, it was stated that the text label attribute was unmapped. If used to duplicate the Win/Lost/Draw concept, it would help the user to map the area colour to its meaning, therefore negating the need for a lookup table or having to use ILOG Discovery interactively. This is a piece of visualization knowledge which should be added to the SemViz system. However, it requires mappings to be created between one source data entity and multiple target visual artefacts (i.e., a one-to-many mapping). Currently, the SemViz algorithm does not support this feature, but would clearly benefit from its inclusion.

Task-based Visualization versus Cognitive Presentation The majority of user testing of visualization techniques is focussed on specific tasks (Task-based Visualization). As such, it is relatively easy to quantify, compare and evaluate the effectiveness of techniques. This approach is acceptable when the user is familiar with a data set's subject domain because they have a solid idea of their goal. However, for users who are not so familiar with the data or the subject domain, their task may be less defined and as such, their goal would be to attempt to gain some additional insight into the data (Cognitive Presentation). The main objective of this research has been to allow *general purpose information visualization for non-experts.* As such, the main aim of the tool is to allow Cognitive Presentation in order to give users insight into the data. Due to this change of focus, the techniques of user testing are subtly different in that they must combine quantitative and qualitative user testing in order to give an effective evaluation of the technique. This is discussed further in section 8.3.

Chapter 8

Conclusions

Contents

During this thesis, we have aimed to meet our objective of facilitating *general purpose information visualization for non-experts.* The motivations for creating this objective are described in Chapter 1, but can be summarised as exploiting the three decades of visualization research to provide cognitive insights into data for people outside of academic circles. We proposed to do this by utilising expert domain knowledge to create cognitively valuable visualizations automatically.

This work has achieved its objectives as set out in Chapter 1. The objectives addressed are indicated in brackets. We have:

- Surveyed the area of Information Visualization to ascertain how suitable current visualization toolkits are for automatic visualization. We also investigated the frameworks and models which have been formalised in order to understand and generalise the process of information visualization. *(Objective 1)*

- Conducted an investigation into the use of Ontologies and Ontology Mapping for driving a knowledge-based, automatic visualization pipeline. *(Objective 1)*

- Proposed a general model for realising/perceptualising data into different perceptual formats such as visual, audial and textual output. This model utilises knowledge stores for capturing source and target semantics and was validated using a proof of concept system. *(Objectives 1 and 2)*

- Created a tree-based mapping toolkit which allows the semi-automatic creation of visualizations. The tool allows the creation and editing of alignments between source entities and target artefacts using a mapping paradigm. *(Objective 2)*

- Created a graph-based, ontology driven visualization pipeline which automatically created scored and ranked visualizations from a source data set. The system uses three ontologies to capture: subject domain semantics; visualization technique semantics; and any available expert domain knowledge. *(Objectives 2 and 3)*

- Tested the graph-based, ontology driven visualization pipeline using a number of datasets from different subject domains and sources using a variety of different visualization techniques from different visualization toolkits. The results have been validated through a user study on non-expert users and also on a visualization expert. *(Objective 4)*

Within section 8.1, we elaborate on how and to what nature the aforementioned objectives were met. In section 8.2, we describe the benefits of an ontological approach to information visualization. Finally, in section 8.3 we discuss the future work that can be conducted within the context of this thesis.

8.1 Summary of Contributions

To create an automatic visualization pipeline which utilised knowledge captured in ontologies, a survey of two distinct fields was conducted. Chapter 2 provides a description of the history and formalisation of the field of information visualization. We discuss various domain specific and general purpose visualization toolkits available including state of the art toolkits which use type-constraints to provide automatic visualizations. We also discuss the advent of web-based, collaborative visualization toolkits and compare their feature set with more traditional toolkits. The main findings of this chapter were that while many sophisticated and powerful visualization techniques exist, they are confined to expert users unless they are presented in a domain specific visualization system. General purpose visualization toolkits provide a degree of automaticity which can help non-expert users. However, while helping novice visualization users, they fail at providing high-quality automatic visualizations.

In Chapter 3, we surveyed the areas of ontologies and ontology mapping. We discussed the purpose of ontologies and looked at various associated technologies for creation, editing and inferencing. We discussed the application of ontologies with particular reference to the semantic web and the tools which are emerging in this field. We then continued with a survey of ontology mapping, including basic techniques and the systems which use these techniques. Additionally, we provide formal definitions of ontologies and ontology mapping processes. Our main conclusion from this chapter is that the area of ontologies and ontology mapping is still developing with no clear, general purpose solution emerging. Ontologies must be created by expert ontology engineers and ontology mapping systems require operation by similarly expert users who ideally need to know the subject domains too. This level of complexity does not lend itself well to assisting in the information visualization process.

In Chapter 4, we proposed a general purpose approach for creating perceptualisations (visual, audial or textual output) from any source data set. This approach uses ontology descriptors to capture target format semantics. From this approach, we successfully built a proof of concept demonstration. Additionally, we investigated tree-based and graph-based mapping approaches. We tested these approaches by creating two simple translators between common graphics formats. These tests were performed using a commercial package (tree-based) and an academic toolkit (graph-based). While informative, these tests did not give conclusive results, so we decided to investigate both approaches further.

In Chapter 5, we built a tree-based XML-centric mapping toolkit. The toolkit uses ontology descriptors and is based on principles of mapping such as schema generation, mapping lines (alignments), mapping locks and automatic alignment creation. The toolkit pragmatically addresses source data idiosyncrasies by providing data cleansing and schema creation facilities where needed. For automatic generation of alignments, it uses a type constrained system which only generates one possible solution during each automatic mapping exercise. The user can adjust mapping parameters at any point and in this way, the system is semi-automatic. While providing a good demonstration of the use of ontology descriptors to capture target format semantics, it does not consider any source semantics (aside from type information). In this way, the visualizations produced do not consider all semantics which are available.

In Chapter 6, we defined and built a system which considers both type and other semantics. The result is a fully automated pipelines (in fact no user interaction is possible during the mapping process) which produces scored and ranked visualizations. The system uses a Domain Ontology (DO) to capture the semantics of the subject domain (e.g., music). A Visual Representation Ontology (VRO) is used to capture semantics about each visualization techniques (e.g., 2D graph, TreeMap). Finally, a Semantic Bridge Ontology (SBO) is used to capture available expert domain knowledge about individual mappings between source entities and target artefacts.

In Chapter 7, we validated the SemViz approach using a further subject domain (football) with source data from four different websites. We have produced visualizations using 4 different techniques (2D graph, TreeMap, Parallel Coordinates and Graph Networks) using two different visualization toolkits (ILOG Discovery and Prefuse). Finally, we validated the results by conducting user studies on 6 non-expert subjects and one visualization expert.

8.2 Benefits of an Ontological Approach to Information Visualization

Knowledge capture An ontology can be a shared and consensual terminology because it is used for information sharing and exchange. Information Visualization is a domain which is ripe for having its knowledge captured more formally. There has been 30 years of development of information visualization techniques. All this knowledge needs to be captured so that firstly, it is not lost, and secondly so that it can be utilised by non-visualization-expert users so that they too can take advantage of these techniques.

Maturity of Visualization Visualization techniques are maturing. Public toolkits such as ManyEyes, Swivel.com, Data360 provide collaborative visualization tools. However, all rely on hand-crafted mapping between source data and target visualization. This is acceptable when the dataset is "cherished" (i.e., high-value data set which has been refined). However, when a dataset is large, unproven or when any visualization task is inherently speculative, there still exists a high barrier to producing a visualization. Therefore, automatic visualization becomes of great importance. We therefore need to capture the years of combined wisdom which the visualization community have created. This wisdom can be expressed in the semantics which can be contained in ontologies.

Certainty Factors Many ontology driven knowledge-based system are based on rules and statements - they only record hard facts. They then reason using description logics. Our system reasons based on a softer notion of knowledge. The advantage of this is that there are no perfectly right or wrong answers. Instead, the system outputs a set of the best alternative visualizations. A second advantage is that system based on certainty factors can be fine-tuned according to user feedback. In general, a Domain Ontology is created by a domain expert who "primes" the ontology with appropriate values for the relationship strengths. These values are indicative and can alter as the system is used.

Scoring Visualization Styles As well as scoring individual visualizations, we can also score the visualization style (e.g., 2D graph versus TreeMap) which is most appropriate for the source data. This follows the same principle as scoring individual visualizations. Because each visualization style has its own ontology instance, the scored mapping permutations can be compared inter- as well as intra-visualization style.

8.3 Further Work

The work in this thesis, while creating some useful and encouraging results, has also generated a number of ideas for future work. We describe these below.

Semantic loss The principle of "semantic loss" is that when mapping between any two domains, there is rarely a complete one-to-one mapping between concepts which have exactly the same meaning. As such, there is always a loss or misinterpretation of a proportion of the meaning. A successful mapping system can perform concept mapping with high accuracy and also minimise the impact of poorly mapped concepts. There are two stages in the SemViz pipeline where there is potential for significant semantic loss: the mapping between the web page (WP) and the Domain Ontology; and the mapping between the Domain Ontology and Visual Representation Ontology. However, the semantic loss which takes place at each stage are subtly different:

1. **Conceptual mis-match (WP to DO)** This is when a mapping between two concepts is incorrect or partially correct. For example, "Composer" mapped to "Artist" is partially correct, whereas "This Week's Position" mapped to "Song Name" is incorrect.

2. **Representational mis-match (DO to VRO)** This is when a concept in the domain

ontology is poorly represented by an artefact in the visual representation. Again, there are varying degrees of inappropriateness. For example, mapping "Artist Name" to "X position" on a 2D Graph is not highly appropriate (especially if there are better alternatives such as "This Week's Position"). However, mapping "This Week's Position" to hierarchy "Level 1" in a TreeMap is highly inappropriate.

In future work, we would aim to measure the degree of this semantic loss and assess how it impacts the visualizations end users see. We would additionally improve the system to minimise the most disruptive semantic loss.

Usability of imperfect visualizations In future work, we would aim to test further the effectiveness of automatic visualizations and how well users cope with "non polished" visualizations. In particular there are areas where it would be useful to capture further semantics about the subject domain. For example, in figure 6.11, we see that the axis of the best visualization have non integer values (both the x and y axis). This is because ILOG Discovery has decided to create non-integer graduations for the scale based on the range of the values. However, the values which are represented on the x and y axis are always integer. In fact, they would never be non-integer because they represent chart positions. The semantics of this last "fact" are not encoded in the source web page, or the domain ontology. Therefore, ILOG Discovery does not utilise this fact to produce a better visualization. Clearly there are many examples of lost semantics which could be further utilised in future.

Multiple perceptions As described in chapter 4 (Concepts of Visualization as Mapping), Information Realisation is not limited to visual representations. We can also consider audial, textual and haptic outputs. The work presented in chapter 6 (SemViz) is a visualization pipeline based on ontologies. However, the pipeline stages which are specific to visualization (the VRO and the interface to visualization toolkits) could be replaced with stages which relate to other representations. A particularly good candidate for this would be an audial representation using sonification. Sonification is a research field which lags visualization in terms if its progress. However, many general purpose sonification toolkits exist in the public domain which could be interfaced to a pipeline based on SemViz. Of course, new ontologies would have to be developed to replace the VRO's and capture the semantics of sonification.

Ontology Generation One aspect of the SemViz work which remains challenging is that of ontology creation. The effort required to create a significant Domain Ontology (DO) is high. Although tools exist to assist in the creation of ontologies, it is still a specialist task where the user must have knowledge of both ontology creation and the subject domain being modelled. One possible solution is that of "reverse engineering" ontologies. A tool could be created which presents domain experts with multiple visualizations and they pick which visualization best represents the source data and thus the subject domain. The tool would then create (or update) an ontology based on these answers. In this way, the subject domain expert would not need to also be an expert in ontology engineering. With a sufficient amount of time, a domain expert could create a domain ontology of sufficient quality that it could be employed in creating visualizations for users who are not subject domain experts.

Bibliography

[ACK$^+$07] E.W. Anderson, S.P. Callahan, D.A. Koop, E. Santos, C.E. Scheidegger, H.T. Vo, J. Freire, and C.T. Silva. VisTrails: Using provenance to streamline data exploration. In *Poster Proceedings of the International Workshop on Data Integration in the Life Sciences (DILS) 2007*, 2007. Poster presentation.

[Ado08] AdobeSVG. Adobe SVG viewer. URL: `www.adobe.com/svg/viewer/install/` [ACCESSED: 24 April 2009], 2008.

[Ahl96] Christopher Ahlberg. Spotfire: an information exploration environment. *SIGMOD Rec.*, 25(4):25–29, 1996.

[AKtKvH06] Zharko Aleksovski, Michel Klein, Warner ten Kate, and Frank van Harmelen. Matching unstructured vocabularies using a background ontology. In *Proc. 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, volume 4248 of *Lecture notes in computer science*, pages 182–197, Praha (CZ), 2006.

[Alt08a] Altova. Mapforce XML mapping toolkit. URL: `www.altova.com` [ACCESSED: 24 April 2009], 2008.

[Alt08b] Altova. SemanticWorks semantic web toolkit. URL: `www.altova.com` [ACCESSED: 24 April 2009], 2008.

[AS94] Christopher Ahlberg and Ben Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 313–317, New York, NY, USA, 1994. ACM Press.

[AW95] Christopher Ahlberg and Erik Wistr. IVEE: An environment for automatic creation of dynamic queries applications. In *In Proceedings of CHI '95*, pages 15–16. ACM, 1995.

[Bau02] Thomas Baudel. Canonical representation of data-linear visualization algorithms and its applications. URL: `www2.ilog.com/preview/Discovery/technology/DiscoveryResearchPaper.pdf` [ACCESSED: 24 April 2009], 2002.

[Bau04] Thomas Baudel. Browsing through an information visualization design space. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 765–766, New York, NY, USA, 2004. ACM Press.

[Bau06] Thomas Baudel. From information visualization to direct manipulation: extending a generic visualization framework for the interactive editing of large datasets. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 67–76, New York, NY, USA, 2006. ACM Press.

[BBC08] BBC. BBC - radio 1 - top 40 music chart. `http://www.bbc.co.uk/radio1/chart/singles.shtml` [ACCESSED: 24 April 2009], 2008.

[BBG01] Massimo Benerecetti, Paolo Bouquet, and Chiara Ghidini. On the dimensions of context dependence: partiality, approximation, and perspective. In *Proc. 3rd International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT)*, volume 2116 of *Lecture notes in computer science*, pages 59–72, Dundee (UK), 2001.

[BDD04] Ken Brodlie, David A. Duce, and David J. Duke. Visualization ontologies : Report of a workshop held at the national e-science centre. Technical report, National e-Science Centre, UK, April 2004.

[BDK05] Thanh-Le Bach and Rose Dieng-Kuntz. Measuring similarity of elements in OWL ontologies. In *Proc. AAAI Workshop on Contexts and Ontologies*, pages 96–99, Pittsburgh (PA US), 2005.

[BDKG04] Than-Le Bach, Rose Dieng-Kuntz, and Fabien Gandon. On ontology matching problems (for building a corporate semantic web in a multi-communities organization). In *Proc. 6th International Conference on Enterprise Information Systems (ICEIS)*, pages 236–243, Porto (PT), 2004.

[BDSW04] Ken Brodlie, David Duce, Musbah Sagar, and Jason Wood. gViz: Visualization and computational steering on the grid. *Proceedings of the UK e-Science All Hands Conference 2004*, pages 54 – 61, 2004.

[BEE$^+$04] Paolo Bouquet, Marc Ehrig, Jerome Euzenat, Enrico Franconi, Pascal Hitzler, Markus Krotzsch, Luciano Serafini, Giorgos Stamou, York Sure, and Sergio Tessaris. Specification of a common framework for characterizing alignment. URL: `ce.sharif.edu/courses/84-85/2/ce694/resources/root/Materials/Ontology Mapping/kweb-221.pdf`,2004.

[Ber83] J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, Madison, WI, 1983. (trans. W. Berg).

[BGM04] Benjamin B. Bederson, Jesse Grosjean, and Jon Meyer. Toolkit design for interactive structured graphics. *IEEE Trans. Softw. Eng.*, 30(8):535–546, 2004.

[BH06] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[BHS03] Thomas Baudel, Bruno Haible, and Georg Sander. Visual data mining with ilog discovery. In Giuseppe Liotta, editor, *Graph Drawing*, volume 2912 of *Lecture Notes in Computer Science*, pages 502–503. Springer, 2003.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web: Scientific american. *Scientific American*, May 2001.

[BMSZ03] Paolo Bouquet, Bernardo Magnini, Luciano Serafini, and Stefano Zanobini. A sat-based algorithm for context matching. In *Proc. 4th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT)*, volume 2680 of *Lecture notes in computer science*, pages 66–79, Stanford (CA US), 2003.

[Bri92] Eric Brill. A simple rule-based part of speech tagger. In *Proc. 3rd Conference on Applied Natural Language Processing (ANLC)*, pages 152–155, Trento (IT), 1992.

[Bro93] K.W. Brodlie. A classification scheme for scientific visualization. In R.A Earnshaw and D. Watson, editors, *Animation and scientific visualization*, pages 125–140. Academic Press, 1993.

[BS84] Bruce G. Buchanan and Edward H. Shortliffe. *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley series in artificial intelligence)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[BS03] Paolo Bouquet and Luciano Serafini. On the difference between bridge rules and lifting axioms. In *Proc. 4th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT)*, volume 2680 of *Lecture notes in computer science*, pages 80–93, Stanford (CA US), 2003.

[BSZ03] Paolo Bouquet, Luciano Serafini, and Stefano Zanobini. Semantic coordination: A new approach and an application. In *Proc. 2nd International Semantic Web Conference (ISWC)*, volume 2870 of *Lecture notes in computer science*, pages 130–145, Sanibel Island (FL US), 2003.

[BSZS06] Paolo Bouquet, Luciano Serafini, Stefano Zanobini, and Simone Sceffer. Bootstrapping semantics on the web: meaning elicitation from schemas. In *Proc. 15th International World Wide Web Conference (WWW)*, pages 505–512, Edinburgh (UK), 2006.

[Car03] M. S. T. Carpendale. Considering visual variables as a basis for information visualisation. Technical report, University of Calgary, Calgary, AB, 2003.

[Car07] Jorge Cardoso. The semantic web vision: Where are we? *IEEE Intelligent Systems*, 22(5):84–88, 2007.

[Cas98] Stephen M. Casner. *A task-analytic approach to the automated design of graphic presentations.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[CCMS96] Lois Mai Chan, John Comaromi, Joan Mitchell, and Mohinder Satija. *Dewey decimal classifcation: a practical guide.* OCLC Forest Press, Dublin (OH US), 1996.

[CEH$^+$09] Min Chen, David Ebert, Hans Hagen, Robert S. Laramee, Robert van Liere, Kwan-Liu Ma, William Ribarsky, Gerik Scheuermann, and Deborah Silver. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl.*, 29(1):12–19, 2009.

[Cha04] Robert J. Chassell. Black, white, and gray. URL: `http://www.rattlesnake.com/notions/black-white-gray.html` [ACCESSED: 24 April 2009], May 2004.

[Cha08] Robert J. Chassell. Certainty factors. URL: `www.rattlesnake.com/notions/certainty-factors.html` [ACCESSED: 24 April 2009], June 2008.

[Chi02] E. H. Chi. *A Framework for Visualizing Information (Human-Computer Interaction Series).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[Cor04] Oscar Corcho. *A declarative approach to ontology translation with knowledge preservation.* PhD thesis, Universidad Politécnica de Madrid, 2004.

[CRF03] William Cohen, Pradeep Ravikumar, and Stephen Fienberg. A comparison of string metrics for matching names and records. In *Proc. KDD Workshop on Data Cleaning and Object Consolidation*, pages 73–78, Washington (DC US), 2003.

[Cro06] Douglas Crockford. The application/json media type for javascript object notation (JSON). RFC 4627 (Informational), July 2006.

[CS08] Silvia Calegari and Elie Sanchez. Object-fuzzy concept network: An enrichment of ontologies in semantic information retrieval. *Journal of the American Society for Information Science and Technology*, 59(13):2171 – 2185, 2008.

[Dat07a] Data360. Data360. URL: `www. data360.org` [ACCESSED: 24 April 2009], 2007.

[Dat07b] Dataplace. Dataplace. URL: `www.dataplace.org` [ACCESSED: 24 April 2009], 2007.

[DBD04] D. J. Duke, K. W. Brodlie, and D. A. Duce. Building an ontology of visualization. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 597 – 598, Washington, DC, USA, 2004. IEEE Computer Society.

[DBDH05] David J. Duke, Ken W. Brodlie, David A. Duce, and Ivan Herman. Do you see what I mean? *IEEE Computer Graphics and Applications*, 25(3):6–9, 2005.

[DDH01] An-Hai Doan, Pedro Domingos, and Alon Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proc. 20th International Conference on Management of Data (SIGMOD)*, pages 509–520, Santa Barbara (CA US), 2001.

[De04] Mike Dean and Guus Schreiber (eds.). OWL web ontology language reference. Recommendation, February 2004.

[del08] delicious. Delicious. URL: `www.delicious.com` [ACCESSED: 24 April 2009], 2008.

[Der08] Derlien. Disk Inventory X. URL: `www.derlien.com` [ACCESSED: 24 April 2009], 2008.

[DGC$^+$98] David A. Duce, Daniela Giorgetti, Christopher S. Cooper, Julian R. Gallop, Ian J. Johnson, E. Robinson, and C. D. Seelig. Reference models for distributed cooperative visualization. *Comput. Graph. Forum*, 17(4):219–233, 1998.

[DS05] D.A. Duce and M. Sagar. skML: A markup language for distributed collaborative visualization. In Louise Lever and Mary McDerby, editors, *Proceedings of Theory and Practice of Computer Graphics 2005*, pages 171–178. Eurographics Association, 2005.

[Duk04] D. J. Duke. Linking representation with meaning. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 595 – 598, Washington, DC, USA, 2004. IEEE Computer Society.

[ES03] Jérôme Euzenat and Heiner Stuckenschmidt. The family of languages approach to semantic interoperability. pages 49–63. IOS press, Amsterdam (NL), 2003.

[ES07] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[Euz01] Jérôme Euzenat. Towards a principled approach to semantic interoperability. In *Proc. IJCAI Workshop on Ontologies and Information Sharing*, pages 19–25, Seattle (WA US), 2001.

[Euz04] Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd International Semantic Web Conference (ISWC)*, volume 3298 of *Lecture notes in computer science*, pages 698–712, Hiroshima (JP), 2004.

[Fei85] Steven Feiner. Apex: An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics and Applications*, 5(11):29–37, 1985.

[Fek04] Jean-Daniel Fekete. The infovis toolkit. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 167–174, Washington, DC, USA, 2004. IEEE Computer Society.

[Fen01] Dieter Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Heidelberg, Germany, 2001.

[FL03] Ferreira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):378–394, 2003.

[Fli08] Flickr. Flickr photo sharing. URL: `www.flickr.com` [ACCESSED: 24 April 2009], 2008.

[Flu08] Flux. Flux software. URL: `en.wikipedia.org/wiki/Flux_(software)` [ACCESSED: 24 April 2009], 2008.

[Fou95] David Foulser. IRIS Explorer: a framework for investigation. *SIGGRAPH Comput. Graph.*, 29(2):13–16, 1995.

[Fou02] OBO Foundry. Obo foundry. URL: `www.obofoundry.org` [ACCESSED: 24 April 2009], 2002.

[Fra08] FractalEdge. Fractal:edge. URL: `www.fractaledge.com` [ACCESSED: 24 April 2009], 2008.

[FTIN97] Issei Fujishiro, Yuriko Takeshima, Yoshihiko Ichikawa, and Kyoko Nakamura. Gadget: Goal-oriented application design guidance for modular visualization environments. In *Proceedings of the IEEE Conference on Visualization 1997 (Vis '97*, pages 245–252. Society Press, 1997.

[FV04] Jean Favre and Mario Valle. AVS and AVS/Express. In Chuck Hansen and Chris Johnson, editors, *The Visualization Handbook*, pages 655–672. Academic Press, December 2004.

[FW04] Oliver Fodor and Hannes Werthner. Harmonise: A step toward an interoperable e-tourism marketplace. *Int. J. Electron. Commerce*, 9(2):11–39, 2004.

[Gan04] Aldo Gangemi. Restructuring semi-structured terminologies for ontology building: a realistic case study in fishery information systems. Deliverable D16, 2004.

[GATTM05] Avigdor Gal, Ateret Anaby-Tavor, Alberto Trombetta, and Danilo Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *The VLDB Journal The International Journal on Very Large Data Bases*, 14(1):50–67, 2005.

[GDGL07] Emilio Di Giacomo, Walter Didimo, Luca Grilli, and Giuseppe Liotta. Graph visualization techniques for web clustering engines. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):294–304, 2007.

[GG04] Chiara Ghidini and Fausto Giunchiglia. A semantics for abstraction. In *Proc. 16th European Conference on Artificial Intelligence (ECAI)*, pages 343–347, Valencia (ES), 2004.

[GGMO03] Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari. Sweetening wordnet with dolce. *AI Magazine*, 24(3):13–24, 2003.

[GJ79] Michael Garey and David Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman & Co., New York (NY US), 1979.

[GRKM94] Jade Goldstein, Steven F. Roth, John Kolojejchick, and Joe Mattis. A framework for knowledge-based interactive data exploration. In *In Journal of Visual Languages and Computing*, pages 339–363, 1994.

[Gro04] LTD Numerical Algorithms Group. IRIS Explorer reference. URL: `www.nag.co.uk` [ACCESSED: 24 April 2009], 2004.

[Gru93] Thomas R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In Nicola Guarino and Roberto Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, volume 43, pages 907–928, Deventer, Netherlands, 1993. Kluwer Academic publishers.

[GS03] Fausto Giunchiglia and Pavel Shvaiko. Semantic matching. *The Knowledge Engineering Review*, 18(3):265–280, 2003.

[GSG+06] Owen Gilson, Nuno Silva, Phil W. Grant, Min Chen, and João Rocha. Information realisation: Textual, graphical and audial representations of the semantic web. In *I-KNOW '06 - 6th International Conference on Knowledge Management*, pages 465 – 472, 2006.

[GSG+07] Owen Gilson, Nuno Silva, Phil W. Grant, Min Chen, and João Rocha. VizThis : Rule-based semantically assisted information visualization. Technical Report CSR-16, Department of Computer Science, Swansea Univeristy, UK, 2007.

[GSGC08] Owen Gilson, Nuno Silva, Phil W. Grant, and Min Chen. From web data to visualization via ontology mapping. *Computer Graphics Forum*, 27(3):959–966, 2008.

[GSY04] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-Match: an algorithm and an implementation of semantic matching. In *Proc. 1st European Semantic Web Symposium (ESWS)*, volume 3053 of *Lecture notes in computer science*, pages 61–75, Hersounisous (GR), May 2004. ex 2004a.

[GSY06] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Discovering missing background knowledge in ontology matching. In *Proc. 17th*

European Conference on Artificial Intelligence (ECAI), pages 382–386, Riva del Garda (IT), 2006.

[GW99] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Verlag, Berlin (DE), 1999.

[GYG05] Fausto Giunchiglia, Mikalai Yatskevich, and Enrico Giunchiglia. Efficient semantic matching. In *ESWC*, pages 272–289, 2005.

[HCL05] Jeffrey Heer, Stuart K. Card, and James A. Landay. prefuse: a toolkit for interactive information visualization. In *CHI '05*, pages 421–430. ACM Press, 2005.

[HG02] Patrick E. Hoffman and Georges G. Grinstein. A survey of visualizations for high-dimensional data mining. *Information visualization in data mining and knowledge discovery*, pages 47–82, 2002.

[HKM07] David Huynh, David Karger, and Rob Miller. Exhibit: Lightweight structured data publishing. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International World Wide Web Conference*, pages 737–746, Banff, Alberta, May 2007. ACM Press.

[HM90] R.B. Haber and D. A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In *Visualization in Scientific Computing*. 1990.

[HMK07] David Huynh, Stefano Mazzocchi, and David Karger. Piggy bank: Experience the semantic web inside your web browser. *Web Semant.*, 5(1):16–27, 2007.

[HMK08] D. Huynh, R. Miller, and D. Karger. Potluck: Data mash-up tool for casual users. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):274–282, November 2008.

[Hof06] Robert D. Hof. Web 2.0: The new guy at work. *Business week*, (3989):58–59, June/JanuarySeptember/ 2006.

[Hor98] Ian Horrocks. The fact system. In *Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux'98)*, volume 1397 of *LNCS*, pages 307–312, Oisterwijk, Netherlands, May 1998. Springer-Verlag.

[HPS04] Adil Hameed, Alun Preece, and Derek Sleeman. Ontology reconciliation. pages 231–250. Springer Verlag, Berlin (DE), 2004.

[HPSB$^+$04] Ian Horrocks, Peter Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. *SWRL: a semantic web rule language combining OWL and RuleML*. 2004. http://www.w3.org/Submission/SWRL/.

[HSMM00] Ivan Herman, Ieee Cs Society, Guy Melançon, and M. Scott Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6:24–43, 2000.

[Huy08] David François Huynh. Timeline. URL: http://simile.mit.edu/timeline [ACCESSED: 24 April 2009], 2008.

[HVW07] Jeffrey Heer, Fernanda B. Viégas, and Martin Wattenberg. Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. *ACM Human Factors in Computing Systems (CHI), 2007*, 2007.

[IBM04] IBM. Opendx: Open visualization data explorer. URL: www.opendx.org [ACCESSED: 24 April 2009], 2004.

[IHT04] Ryutaro Ichise, Masahiro Hamasaki, and Hideaki Takeda. Discovering relationships among catalogs. In *Proc. 7th International Conference on Discovery Science*, volume 3245 of *Lecture notes in computer science*, pages 371–379, Padova (IT), 2004.

[ITH03] Ryutaro Ichise, Hideaki Takeda, and Shinichi Honiden. Integrating multiple internet directories by instance-based learning. In *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 22–30, Acapulco (MX), 2003.

[Jen95] Chris Jenks. *Visual Culture*. Routledge, 1995.

[JS05] Eui-Chul Jung and Keiichi Sato. A framework of context-sensitive visualization for user-centered interactive systems. *User Modeling 2005*, pages 423–427, 2005.

[JZY08] Ye Jun, Li Zhishu, and Ma Yanyan. JSON based decentralized SSO security architecture in e-commerce. *Electronic Commerce and Security, International Symposium*, 0:471–475, 2008.

[Kei02] Daniel A. Keim. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, 2002.

[KHG03] Robert Kosara, Helwig Hauser, and Donna L. Gresh. An interaction view on information visualization. In *State-of-the-Art Proceedings of EUROGRAPHICS 2003 (EG 2003)*, pages 123–137, 2003.

[KJH$^+$05] Jaehong Kim, Minsu Jang, Young-Guk Ha, Joo-Chan Sohn, and Sang-Jo Lee. MoA: OWL ontology merging and alignment tool for the semantic web. In *Proc. 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)*, volume 3533 of *Lecture notes in computer science*, pages 722–731, Bari (IT), 2005.

[KK94] Peter R. Keller and Mary M. Keller. *Visual Cues: Practical Data Visualization*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1994.

[KKV06] Teuvo Kohonen, Konstantinos Kotis, and George Vouros. *HCONE approach to Ontology Merging*, volume 3053 of *Lecture notes in computer science*. Springer, Hersounisous (GR), 2006.

[Kle01] Michel Klein. Combining and relating ontologies: an analysis of problems and solutions. In *Proc. IJCAI Workshop on Ontologies and Information Sharing*, Seattle (WA US), 2001.

[KN03] Jaewoo Kang and Jeffrey Naughton. On schema matching with opaque column names and data values. In *Proc. 22nd International Conference on Management of Data (SIGMOD)*, pages 205–216, San Diego (CA US), 2003.

[KR94] K. Konstantinides and J. R. Rasure. The Khoros software development environment for image and signal processing. *Image Processing, IEEE Transactions on*, 3(3):243–252, 1994.

[KSC$^+$08] D. Koop, C. E. Scheidegger, S. P. Callahan, J. Freire, and C. T. Silva. Viscomplete: Automating suggestions for visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1691–1698, 2008.

[LAB$^+$79] B. Liskov, R. R. Atkinson, T. Bloom, E. B. Moss, R. Schaffert, and A. Snyder. Clu reference manual. Technical report, Cambridge, MA, USA, 1979.

[LC94] Wen-Syan Li and Chris Clifton. Semantic integration in heterogeneous databases using neural networks. In *Proc. 20th International Conference on Very Large Data Bases (VLDB)*, pages 1–12, Santiago (CL), 1994.

[LCM$^+$02] Claudia Leacock, Martin Chodorow, George Miller, Mong Li Lee, Liang Huai Yang, Wynne Hsu, and Xia Yang. Xclust: clustering XML schemas for effective integration. *Proc. 11th International Conference on Information and Knowledge Management (CIKM)*, 24(1):292–299, 2002.

[Lev65] V Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady akademii nauk SSSR*, 4(163):845 – 848, 1965.

[LG90] Douglas Lenat and Ramanathan Guha. *Building large knowledge-based systems*. Addison Wesley, Reading (MA US), 1990.

[LMC98] Claudia Leacock, George A. Miller, and Martin Chodorow. Using corpus statistics and wordnet relations for sense identification. *Comput. Linguist.*, 24(1):147–165, March 1998.

[LNE89] James Larson, Shamkant Navathe, and Ramez Elmasri. A theory of attributed equivalence in databases with application to schema integration. *IEEE Transactions on Software Engineering*, 15(4):449–463, 1989.

[LRB$^+$97] M. Livny, R. Ramakrishnan, K. Beyer, G. Chen, D. Donjerkovic, S. Law, J. Myllymaki, and K. Wenger. Devise: Integrated querying and visual exploration of large datasets. In *In Proceedings of ACM SIGMOD*, pages 301–312, 1997.

[LS98] Ora Lassila and Ralph Swick. Resource description framework (RDF) model and syntax specification. Technical report, 1998.

[LSPR93] Ee-Peng Lim, Jaideep Srivastava, Satya Prabhakar, and James Richardson. Entity identification in database integration. In *Proc. 9th International*

Conference on Data Engineering (ICDE), pages 294–301, Vienna (AT), 1993.

[LV03] Peter Lyman and Hal R. Varian. How much information. Retrieved from http://www.sims.berkeley.edu/how-much-info-2003. [ACCESSED: 24 April 2009], 2003.

[MA01] Diana Maynard and Sophia Ananiadou. Term extraction using a similarity-based approach. pages 261–278. John Benjamins, Amsterdam (NL), 2001.

[MAB$^+$97] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *SIGGRAPH '97*, pages 389–400, NY, 1997.

[Mac86] Jock D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, 1986.

[Maz08] Stefano Mazzocchi. Timeplot. URL: `http://simile.mit.edu/timeplot/` [ACCESSED: 24 April 2009], 2008.

[Mee99] Robert Meersman. The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems. In *CODAS*, pages 1–14, 1999.

[MHS07] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. Show Me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.

[Mil95] George Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[MKFR03] I. Mierswa, R. Klinkenberg, S. Fischer, and O. Ritthoff. A flexible platform for knowledge discovery experiments: Yale – yet another learning environment. Technical Report 531, University of Dortmund, 2003.

[MM94] J. Jeffrey Mahoney and Raymond J. Mooney. Comparing methods for refining certainty-factor rule-bases. In *International Conference on Machine Learning*, pages 173–180, 1994.

[MMSV02] Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. MAFRA - A MApping FRAmework for Distributed Ontologies. volume 2473 of *LNCS*, pages 235–250, Sigenza, Spain, September 2002. Sigenza, Spain, Springer.

[MNJ05] Prasenjit Mitra, Natalya Noy, and Anuj Jaiswal. Ontology mapping discovery with uncertainty. In *Proceedings of the 4th International Semantic Web Conference (ISWC)*, volume 3729 of *LNCS*, pages 537–547, Galway (IE), 2005.

[msK06] mc schraefel and David Karger. The pathetic fallacy of RDF. Technical report, ECS, University of Southampton, Southampton UK, 2006.

[NCB02] NCBO BioPortal. Ncbo bioportal. URL: `bioportal.bioontology.org` [ACCESSED: 24 April 2009], 2002.

[Nie95] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.

[NM01] Natalya F. Noy and Mark A. Musen. Anchor-prompt: Using non-local context for semantic matching. In *IJCA '01: IIn Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence*, pages 63–70, Seattle (WA), USA, Aug 2001. Seattle (WA), USA.

[NNHY06] Fumitaka Nakaizumi, Haruo Noma, Kenichi Hosaka, and Yasuyuki Yanagida. Spotscents: A novel method of natural scent delivery using multiple scent projectors. In *VR '06: Proceedings of the IEEE conference on Virtual Reality*, pages 207–214, Washington, DC, USA, 2006. IEEE Computer Society.

[NS05] Henrik Nottelmann and Umberto Straccia. splmap: A probabilistic approach to schema matching. In *Proc. 27th European Conference on Information Retrieval Research (ECIR)*, pages 81–95, Santiago de Compostela (ES), 2005.

[NS06] Henrik Nottelmann and Umberto Straccia. A probabilistic, logic-based framework for automated web directory alignment. volume 204 of *Studies in fuzziness and soft computing*, pages 47–77. Springer Verlag, 2006.

[NSD$^+$01] Natalya F. Noy, Michael Sintek, Stefan Decker, Monica Crubezy, Ray W. Fergerson, and Mark A. Musen. Creating semantic web contents with protege-2000. *IEEE Intelligent Systems*, 2(16):60–71, 2001.

[NWN$^+$01] Saul Needleman, Christian Wunsch, Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjorn Naeve, Mikael Nilsson, Matthias Palmr, Tore Risch, Ian Niles, and Adam Pease. Towards a standard upper ontology. *Proc. 2nd International Conference on Formal Ontology in Information Systems (FOIS)*, 48(3):2–9, 2001.

[Oct08] Octaga. Octaga. URL: `www.octaga.com` [ACCESSED: 24 April 2009], 2008.

[OD06] Eyal Oren and Renaud Delbru. A prototype for faceted browsing of RDF data. URL: `www.semanticscripting.org/SFSW2006/challenge/FacetedBrowsing.pdf` [ACCESSED: 24 April 2009], May 2006.

[OL05] Miguel A. Otaduy and Ming C. Lin. Introduction to haptic rendering. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 3, New York, NY, USA, 2005. ACM.

[O'R05] Tim O'Reilly. O'reilly network: What is web 2.0. URL: `www.oreillynet.com/lpt/a/6228` [ACCESSED: 24 April 2009], September 2005.

[Par04] David Parry. A fuzzy ontology for medical document retrieval. In *ACSW Frontiers '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 121–126, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[PDYP05] Rong Pan, Zhongli Ding, Yang Yu, and Yun Peng. A bayesian network approach to ontology mapping. In *Proceedings of the Fourth International Semantic Web Conference*, pages 563 – 577, November 2005.

[PLB$^+$01] Hanspeter Pfister, Bill Lorensen, Chandrajit Bajaj, Gordon Kindlmann, Will Schroeder, Lisa Sobierajski Avila, Ken Martin, Raghu Machiraju, and Jinho Lee. The transfer function bake-off. *IEEE Computer Graphics and Applications*, 21(3):16–22, 2001.

[PS00] Christine Parent and Stefano Spaccapietra. Database integration: the key to data interoperability. pages 221–253. The MIT Press, Cambridge (MA US), 2000.

[Rac08] RacerSystems. Racer Pro. URL: `www.racer-systems.com` [ACCESSED: 24 April 2009], 2008.

[RB01] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[RKMG94] Steven F. Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. Interactive graphic design using automatic presentation knowledge. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 112–117, New York, NY, USA, 1994. ACM Press.

[RKR06] Philippa Rhodes, Eileen Kraemer, and Bina Reed. VisIOn: an interactive visualization ontology. In *ACM-SE 44: Proceedings of the 44th annual Southeast regional conference*, pages 405–410, New York, NY, USA, 2006. ACM Press.

[RMC91] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: animated 3d visualizations of hierarchical information. In *ACM CHI '91 Conference on Human Factors in Computing Systems*, pages 189–194, New Orleans, LA, USA, 1991. ACM Press.

[RR06] Hans Rosling and Ola Rosling. *New software brings statistics beyond the eye*, pages 522–530. Organisation for Economic Co-operation and Development, 2006.

[SAR08] Gao Shu, Nick J. Avis, and Omer F. Rana. Bringing semantics to visualization services. *Adv. Eng. Softw.*, 39(6):514–520, 2008.

[SC98] Y. Shahar and C. Cheng. Ontology-driven visualization of temporal abstractions. In *Proceedings of the Eleventh Workshop on Knowledge Acquisition Modeling and Management*, pages 1 – 20, Banff, Alberta, Canada, 1998.

[Sch99] Randall Schuh. *Biological systematics: principles and applications.* Cornell University Press, Ithaca (NY US), 1999.

[SdM06] Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Using the semantic web as background knowledge for ontology mapping. In *Proc. 1st ISWC International Workshop on Ontology Matching (OM)*, pages 1–12, Athens (GA US), 2006.

[SH02] Chris Stolte and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8:52–65, 2002.

[Shi05] Clay Shirky. Ontology is overrated: Categories, links, and tags. URL: `www.shirky.com/writings/ontology_overrated.html` [ACCESSED: 24 April 2009], 2005.

[Shn96] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages, 1996.*, pages 336–343, 1996.

[Shv04] Pavel Shvaiko. Iterative schema-based semantic matching. Technical Report DIT-04-020, 2004.

[SI94] Hikmet Senay and Eve Ignatius. A knowledge-based system for visualization design. *IEEE Comput. Graph. Appl.*, 14(6):36–47, 1994.

[Sim08] SimPack. SimPack. http://www.ifi.unizh.ch/ddis/simpack.html [ACCESSED: 24 April 2009], 2008.

[SKSP07] Markus Schedl, Peter Knees, Klaus Seyerlehner, and Tim Pohle. The CoMIRVA toolkit for visualizing music-related data. In *Proc. 9th Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'07)*, Norrkoping, Sweden, May 2007.

[SKW05] Markus Schedl, Peter Knees, and Gerhard Widmer. Interactive poster: Using CoMIRVA for visualizing similarities between music artists. In *Proceedings of the 16th IEEE Visualization 2005 Conference (Vis'05)*, Minneapolis, Minnesota, October 2005.

[SLCN88] Amit Sheth, James Larson, Aloysius Cornelio, and Shamkant Navathe. A tool for integrating conceptual schemas and user views. In *Proc. 4th International Conference on Data Engineering (ICDE)*, pages 176–183, Los Angeles (CA US), 1988.

[Sma08] SmartMoney. MarketMap - a view of the whole finanical market. URL: `www.smartmoney.com/map-of-the-market/` [ACCESSED: 24 April 2009], 2008.

[Spe00] Robert Spence. *Information Visualization.* Addison Wesley, December 2000.

[SPG$^+$07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Web Semant.*, 5(2):51–53, 2007.

[SR05] Nuno Silva and Joao Rocha. Multi-Dimensional Service-Oriented Ontology Mapping. *International Journal of Web Engineering and Technology*, 2(1):50–80, 2005.

[SVC$^+$05] Anastasiya Sotnykova, Christelle Vangenot, Nadine Cullot, Nacéra Bennacer, and Marie-Aude Aufaure. Semantic mappings in description logics for spatio-temporal database schema integration. 3534:143–167, 2005.

[SVG08] SVG. Scalable Vector Graphics (SVG) specification. URL: `http://www.w3.org/Graphics/SVG/` [ACCESSED: 24 April 2009], 2008.

[SVK$^+$08] C.E. Scheidegger, H.T. Vo, D. Koop, J. Freire, and C.T. Silva. Querying and re-using workflows with VisTrails. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008.

[SW01] Ben Shneiderman and Martin Wattenberg. Ordered treemap layouts. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 73, Washington, DC, USA, 2001. IEEE Computer Society.

[Swa02] Aaron Swartz. The semantic web in breadth. URL: `logicerror.com/semanticWeb-long` [ACCESSED: 24 April 2009], 2002.

[SWe04] Mike Smith, Christopher Welty, and Deborah McGuinness (eds.). OWL web ontology language guide. Recommendation, February 2004.

[Swi08] Swivel. Swivel. URL: `www.swivel.com` [ACCESSED: 24 April 2009], August 2008.

[TBF01] D. Thompson, J. Braun, and R. Ford. *OpenDX: Paths to Visualization*. Visualization and Imagery Solutions, Inc., 2001.

[Tho01] Henry Thompson. Normal form conventions for XML representations of structured data. In *XML Conference & Exposition*, Victoria (B.C.), Canada, 2001. Alexandria (VA), USA, IDEAlliance.

[Tib08] Tibco. Tibco : Enterprise software solutions. URL: `www. tibco.com` [ACCESSED: 24 April 2009], 2008.

[TM04] Melanie Tory and Torsten Möller. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84, January/February 2004.

[Val99] Petko Valtchev. *Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets*. Thèse d'informatique, Université Grenoble 1, 1999.

[VE97] Petko Valtchev and Jerome Euzenat. Dissimilarity measure for collections of objects and values. In *Proc. 2nd Symposium on Intelligent Data Analysis (IDA)*, volume 1280 of *Lecture notes in computer science*, pages 259–272, London (UK), 1997.

[VWvH$^+$07] Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. ManyEyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

[War04] Colin Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[WBP07] Gunther Weber, Peer-Timo Bremer, and Valerio Pascucci. Topological landscapes: A terrain metaphor for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1416–1423, 2007.

[Web08a] Web3D. Web3d consortium. URL: `www.web3d.org` [ACCESSED: 24 April 2009], 2008.

[Web08b] Webcoding. BBC - Radio 1 - top 40 music chart - XML feeds. `http://viper.bsg.webcoding.co.uk/top40.xml` [ACCESSED: 24 April 2009], 2008.

[Wil05] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[WL90] Stephen Wehrend and Clayton Lewis. A problem-oriented classification of visualization techniques. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 139–143. IEEE Computer Society Press, 1990.

[WP94] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

[WSFB07] Daniel Wigdor, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. Perception of elementary graphical elements in tabletop and multi-surface environments. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 473–482, New York, NY, USA, 2007. ACM.

[WWB97] Jason Wood, Helen Wright, and Ken Brodlie. Collaborative visualization. In *VIS '97: Proceedings of the 8th conference on Visualization '97*, pages 253–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.

[XZS06] Lijun Xie, Yao Zheng, and Bin Shen. Ontology construction for scientific visualization. *Computer and Computational Sciences, International Multi-Symposiums on*, 1:778 – 784, 2006.

[YaKSJ07] Ji Soo Yi, Youn ah Kang, John Stasko, and Julie Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

[YMC05] J. Younesy, T. Moller, and H. Carr. Visualization of time-varying volumetric data using differential time-histogram table. *Fourth International Workshop on Volume Graphics, 2005*, pages 21–224, 2005.

[Zho99] Michelle Xue Zhou. *Automated generation of visual discourse*. PhD thesis, New York, NY, USA, 1999. Adviser-Steven K. Feiner.

List of Figures

List of Tables