Swansea University
Prifysgol Abertawe

SWANSEA UNIVERSITY

REPORT SERIES

VizThis : Rule-based Semantically Assisted Information Visualization

Owen Gilson, Nuno Silva, Phil W. Grant, Min Chen, and João Rocha

Report # CSR 16-2007

Computer Science
Gwyddor Cyfrifiadur

VizThis : Rule-based Semantically Assisted Information Visualization

Owen Gilson[1], Nuno Silva[2], Phil W. Grant[1], Min Chen[1], and João Rocha[2]

[1] Department of Computer Science, University of Wales Swansea, UK
{csowen, P.W.Grant, M.Chen}@swan.ac.uk
[2] ISEP Instituto Superior de Engenharia, Instituto Politécnico do Porto, Portugal
{Nuno.Silva, Joao.Rocha}@dei.isep.ipp.pt

Abstract. Information Visualization can be considered as the task of mapping data entities in a source file to representation artefacts in a target visualization format. As such, mapping plays a significant role in this process. The field of Ontology Mapping presents a wealth of work in the information mapping domain which we propose to exploit by applying it to Information Visualization for the Semantic Web. In this paper we describe a User Interface and a tool (VizThis) which demonstrates the proposed approach. We discuss the benefits which a mapping paradigm brings to Information Visualization, including automaticity and rule constraint. The system facilitates automatic mapping while still allowing users the ability to tweak the chosen mappings in order to improve the cognitive value of the visualizations. Additionally, the mapping choices available are constrained by rules governed by the characteristics of both the source and target format. A worked example is provided together with the results of a qualitative user evaluation of the resulting visualizations.

1 Introduction

Card et. al[13] state that: "Information visualization is the use of computer-supported, interactive, visual representations of abstract data to amplify cognition".

Information Visualization can be a labour intensive task because of the challenges of mapping source data entities to target representation artefacts. These challenges include syntactic, structural and semantic aspects. When we consider Information Visualization for the Semantic Web, the syntactic challenges are simplified due to the standardised formats of XML and RDF. Also, syntactic and structural aspects can be automated through standard mapping software such as Altova MapForce[1] and MAFRA toolkit[14][19]. However the semantic aspect is far more complex in that it requires the most human involvement.

When we consider Semantic Web Information Visualization in these terms, we begin to realise that there are parallels with the mapping community. Information and data mapping are very broad areas with a great deal of prior work.

However, one area which has a particular affinity with the Semantic Web is Ontology Mapping. It has progressed significantly over recent years and has drawn heavily from the work of other mapping areas. Ontology Mapping has some interesting characteristics which we can apply to Information Visualization:

Semantic Bridging The ability to form mappings between related concepts.
Automaticity The degree to which mappings can be created automatically.
Value Transformation The ability to assign a function or regular expression to source object so that its value is transformed before being passed to a target object.
Extensional Specification The ability to categorise the relationship between hierarchical or relational objects as first class entities.

The choice of semantic mappings between source data entities and target representation artefacts is a very subjective one. It also has a great bearing on the cognitive value of the final visualization. However, there are some elements of the semantic mapping process which can be automated. We believe that we can simplify the semantic mapping process by automatically predicting mappings and by constraining the mappings available to simplify the users's choices.

In this paper, we propose to define Information Visualization in terms of Ontology Mapping processes and terminology. We discuss related work in section 2. VizThis, a tool and user interface for performing Information Visualization of the Semantic Web is described in section 3, where we give a worked example of the visualization of sports fans information expressed in XML. We describe the techniques of rules based constraints and semantic assistance in section 4. We present the results of an informal, qualitative user evaluation of the visualization of a set of Top 40 music chart information in section 5. We describe how the visualization of graph based information (such as RDF) differs from tree-based information and how the VizThis tool can be adapted to address these differences in section 6. Finally, we give plans for future work and present our conclusions in section 7.

2 Related Work

The predominant format for information representation in the Semantic Web is RDF. Because RDF statements intrinsically represent labeled, directed pseudographs, much of the work on their visualization focusses on nodes and their relationships, thus giving an insight into the data structure. The resulting visualizations can answer queries such as: "which schema parts are present in an instance, which properties are specific to a given instance in an instance set, and how do schemas evolve in time"[7]. We believe that while structural visualization is very important for RDF information, the need to visualize attribute values should not be forgotten. The example visualizations in this paper focus on the visualization of attribute values.

An example of a tool used for RDF visualization is GViz [6], a general purpose graph visualization tool. This tool can browse and edit customisable visualizations. GViz separates the graph layout (*where* nodes and edges are drawn) from the graph mapping (*how* nodes and edges are drawn). GViz supports many graph layouts including spring embedder, the directed tree, the 3D stacked layout and the nested layout. The shape, size and colour of each node and edge can be customised by the user by writing Tcl scripting code. Additionally, the user can specify custom interactions, again using Tcl code, which allow actions to be taken when the user selects a node or edge with the mouse. This could for example be used for displaying additional attribute information. More details on GViz can be found in [11].

As can be seen, GViz provides a great deal of flexibility and supports many different visualization styles, layouts and interactions. However, the details of the target format are encapsulated within GViz and are not exposed to the user. The advantage of this approach is that the user deals with visual artefacts at a high conceptual level and can therefore concentrate on the best way to visualize the source information. The disadvantage of this approach is that the user looses the flexibility which comes with being able to manipulate the visual artefacts in the target format directly. Additionally, the GViz approach is limited to one target format. We categorise the matching and target generation phase of GViz as being *closely coupled*. This approach is also taken by other tools such as SpotFire[5].

The paradigm proposed and described in this paper we call, "Visualization as Mapping" and is based on a process which is *loosely coupled* end-to-end. We consider the whole Information Visualization process stack as a mapping problem. The source information (to be mapped from) and the target format (to be mapped to) are considered in the same way as any other information format (e.g. an ontology). The source information is made up of *source data entities* and the target format is made up of *target representation artefacts*. Mappings are created between the entities and the artefacts. During the mapping process entity values can be transformed. Given this architecture, the process stack is made up of one very flexible end-to-end mapping layer.

We propose to define the features of this layer in terms of the techniques and terminology of Ontology Mapping. The anticipated gains of this approach are discussed in section 7. An area of research which we would like to address with the Visualization as Mapping paradigm is "Tailorable Information Visualization"[8] which argues for a new class of tools. It states that, "tools should combine user-defined displays with automatically generated displays and be integrated with search mechanisms capable of accessing data in a variety of information bases". We believe that the Visualization as Mapping approach has the potential to address all of these aspects.

3 A Description of VizThis

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

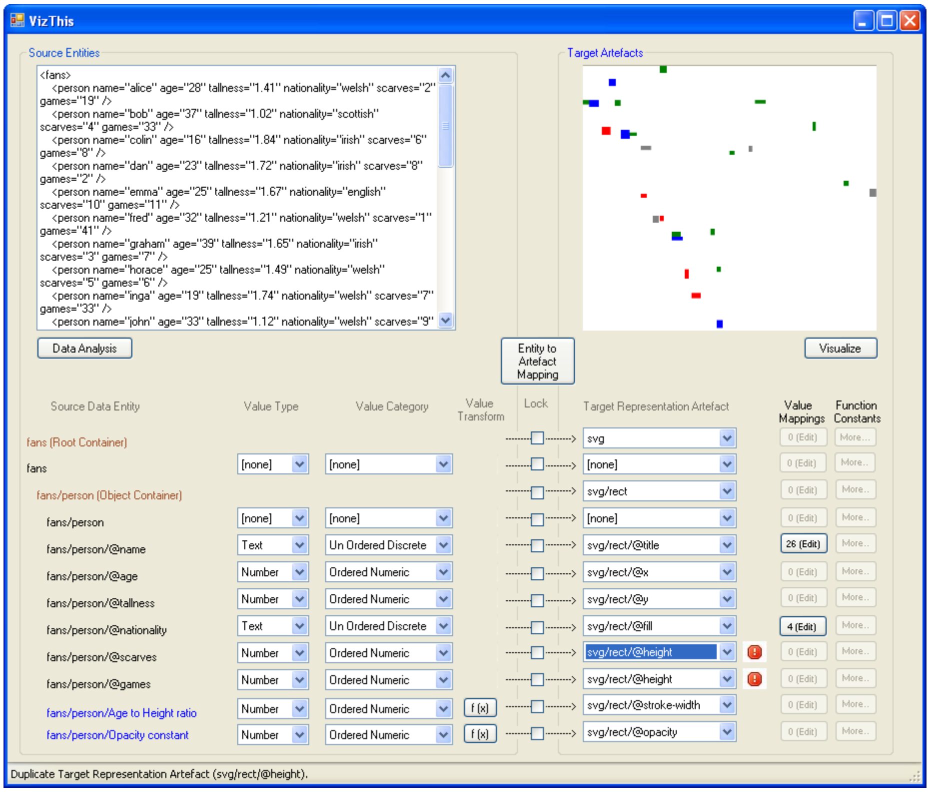


Fig. 1. VizThis : Visualizing the Sports fans data set in SVG

VizThis tool. We make extensive reference to the screenshot shown in figure 1. The screenshot shows the visualization of the sports fans information, expressed in XML

3.1 Source Data File

The source file is shown in the top left hand corner of figure 1. An extract from the source is shown 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>
```

This source information is analysed by the tool and from this, the source data entities are populated (see bottom left quarter of figure 1).

3.2 Source Data Entities

A source data entity is an element or attribute (when dealing with XML); or a resource node or property node (when dealing with RDF). 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`. The value type and value category of an entity are automatically populated, based on an analysis of the values of each of the source data entities. If a data schema is present (XSD in XML, RDFS for RDF), then the tool will take into account any relevant information provided. The value type is either explicit if provided by the schema, or inferred from the analysis of the data. It can be integer, real, string, date or URL etc.

Value Category is a categorisation of the values in terms of the degree and form of the variation. A value category is one of 3 values:

1. **orderedNumeric (ON)** Numeric values which by their very nature are ordered. For example, *age* (0 to 120) expressed as an integer.
2. **orderedDiscrete (OD)** Non-numeric values which have discrete values with an implied ordering. For example, *t-shirt-size* (S, M, L, XL, XXL).
3. **unorderedDiscrete (UOD)** Non-numeric values which have discrete values but with no implied ordering. For example, *name-of-person* (Bob, Mary, John).

Additionally, the user can add *virtual entities* (shown in blue). These are entities which are not present in the source data, but are created to map to specific artefacts in the target. They are used for such things as creating new entities, possibly based on the value of other (concrete) entities. An example of a virtual entity is shown in figure 1 as the blue text, `fans/person/Age to Height ratio` (see Section 3.6 for more information). Each source data entity can be mapped to a target representation artefact.

3.3 Target Representation 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 artefact which each entity is mapped to is shown on the right hand side. These are initially predicted by the system, thus demonstrating *mapping automaticity*. The technique of automatic mapping is also known as *semantic assistance* and is discussed in Section 4.3. The mappings can also be tweaked by the user within the constraints of a pre-defined set of rules. For example, in figure 1, we see that the system has detected an invalid mapping which is indicated by the red warning icons. This is because two different artefacts are mapped to the same entity. The rules, which are specific to the target representation format, indicate that this is not allowed. The rules system is described in detail in Section 4.

While the user is tweaking the mapping choices, the user can tell the system to re-generate the automatic mappings. If the user has decided that there are some mappings which they are happy with and do not wish to alter, they can lock them by selecting the Lock checkbox. In this case, the rule-system would consider such locks as definitive mappings and take them as anchors[18] for the rest of the mapping process.

Sometimes, there is also the need for the values of entities to be mapped to specific artefact values. For example, in the sports fans example, the `fans/person/@nationality` entity has four discrete values (Welsh, Scottish, Irish, English). This entity is mapped to the `svg/rect/@fill` artefact which represents the fill colour of the rectangle. Each entity value needs to be mapped to discrete colour values which the artefact has available (in this case, conventionally: red; blue; green; and white respectively). This can be specified using the Value Mappings button.

Some artefacts have constants associated with them. For example, the `svg/rect/@x` and `svg/rect/@y` artefacts have ranges associated with them which depend on the screen size available for the target user's display. These constants can be updated through the Function Constants button.

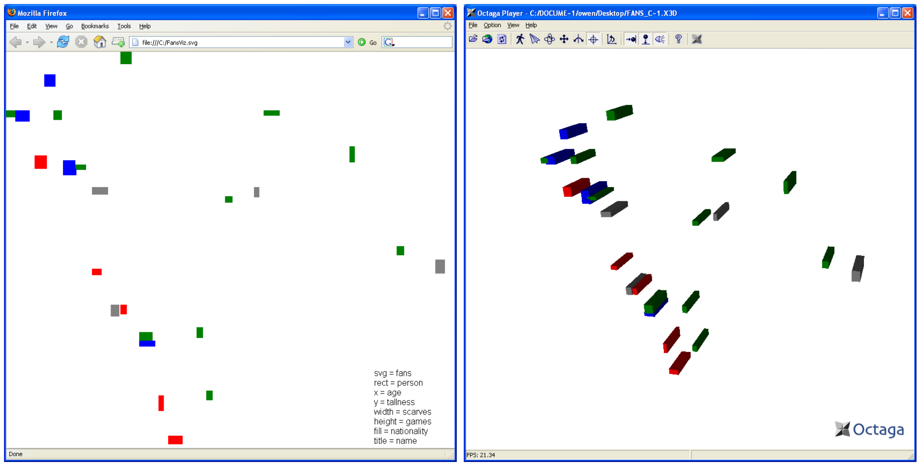
3.4 Target Visualization

Whenever any aspect of the mappings are changed, provided they are valid, the visualization in the top right of the window is updated to reflect these changes. The resulting SVG visualization is shown in figure 2a. For comparison, the same information set is shown visualized in X3D in figure 2b. The visualization produced is currently quite simplistic. Each `person` in the source is represented as a `rectangle` in the target, with `x` and `y` coordinates representing `age` and `tallness` (respectively), `width` and `height` representing number of `scarves` owned and `games` attended (respectively) and each rectangle's `colour` representing `nationality`. The visualizations can be accessed electronically in the online appendix[12].

3.5 Virtual Entities

Virtual entities are indicated as blue source entities (see the last two entities in figure 1). They are virtual in the sense that they do not actually exist in the source file. However, the entities can be mapped to any artefact in the target. The value of the entity is specified as an expression. This expression is specified in a dialogue box which appears when the Value Transform button is pressed (shown as f(x) in figure 1). The expression is composed of standard operators (arithmetic, string etc.) applied to other entities in the source (whether virtual or concrete entities). Alternatively, a regular expression can be specified which can be useful for data cleansing (see below). The choice of name for the virtual entity has no bearing on the target, but is used to aid clarity for the user and must be specified if the virtual entity is referenced by any other entity.

A virtual entity may be created at any point in the source hierarchy. The hierarchical position of the virtual entity within the source is significant since virtual entities need to be mapped at defined points in the mapping process.



(a) SVG (2D visualization) (b) X3D (3D visualization)

Fig. 2. Sports Fans dataset Online Information Visualization

For example, having a virtual entity as a child of `fans` would mean that it is called once in the mapping process. But having a virtual entity as a child of `fans/person` would mean that it is called every time that person is represented (i.e. 26 times in the full example information set).

Virtual entities provide many uses. Many of these are common pre-defined tasks. As such, the system allows a quick, guided process for their creation. We call these *Virtual Entity Wizards*. These are detailed below:

Data Cleansing Virtual entities can be used for modifying the value of a source entity before the value is transferred to the target artefact. This is useful for data cleansing tasks. This example occurs in the `top-forty/chart/@weeks` entity in the Top 40 information set (see section 5) where the parenthesis surrounding the numeric values need to be removed.

New Child Entity Creation If you have many attributes for each object in the source data, you may want to split these into separate child objects. This is particularly relevant if you have more attributes per source entity object than you do attributes per target representation artefact. You can therefore map single source objects to multiple target objects.

Visualizing Structure When structure is present in the source, whether it is tree or graph based, we must consider how that structure should be visualized. In a tree based format, structure is conveyed as hierarchical levels of elements. In a graph based format, structure is conveyed as relationships between nodes. An initial method which we have considered for representing both hierarchy and relationships is child representation objects. Virtual entities can be used to position child representation objects relative to parent objects. Multiple virtual entities will be created by the wizard to deal with child object positioning and any arrows necessary.

Target Representation Artefact Constants Sometimes, there is the need to have target representation artefacts which have constant values (i.e. they are not related to source data entities). They may be required for aesthetic reasons, or to satisfy the requirements of the target format. An example is the `opacity` attribute in SVG which is set to zero by default. Setting this to a more suitable value (say 0.8) allows overlapping objects to be identified more clearly.

4 Constraint Rules and Semantic Assistance

4.1 Levels of Rules

There are 4 levels of rules which are applicable to the system:

Absolute Hard rules which cannot be altered. An example is that there can not be duplicate target entities. This is shown in figure 1, where there are two artefact attributes which are set to the same attribute (`svg/rect/@width`). This is indicated by the two red warning icons.

Preferential Soft rules (or guidelines) which are set by a developer when a new representation format is added to the system. Users can override them. An example is the general principle that the SVG attributes `svg/rect/@x` and `svg/rect/@y` are best for mapping to source data entities of Value Category orderedNumeric.

Learned We intend to apply ontology based machine-learning techniques[9][10] in future work.

User Specified This is a method to allow users to specify their own rules. These may be specific to the user or shared amongst all users.

4.2 Scope of Rules

Rules have 3 varying scopes of influence:

Inter Artefact Attribute Concerned with the relationship between target artefact attributes. For example, no two source entities may be mapped to the same target representation artefact.

Inter Artefact-Entity Concerned with the relationship between source entities and target representation artefacts. For example, you cannot map an entity object container (shown as brown entities in figure 1) to a representation artefact attribute. But you can map an entity attribute to a representation artefact object.

Stand Alone Rules which apply to individual artefacts. For example, the `svg/rect/@width` attribute should always have a value if there is a `svg/rect` object present.

4.3 Semantic Assistance

VizThis can automatically create mappings between source entities and target representation artefacts. We describe this as *semantic assistance* which is a way of increasing the *automaticity* of the mapping process. VizThis maps entities to artefacts based on the Value Type and Value Category as described in Section 3.2. For example, entities which are classified as orderedNumeric(ON) in the source are best mapped to target representation artefacts which are also orderedNumeric(ON). Despite this being a fairly simple technique it has produced encouraging results. In the next section we describe a user test for evaluating the efficiency of the semantic assistance by measuring the quality of the resulting visualizations.

5 Test Cases for Evaluating VizThis

In order to evaluate the efficiency of the VizThis interaction metaphor, we propose to measure 3 aspects:

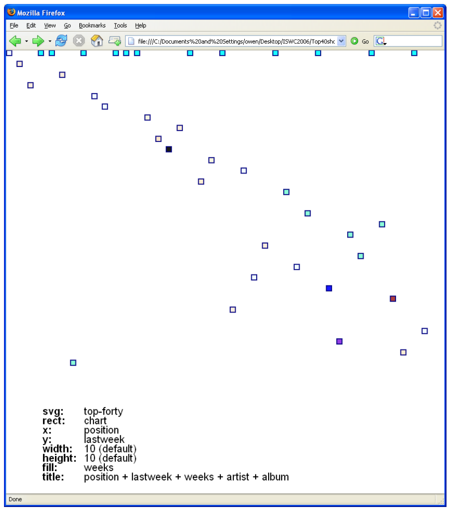
1. The usability of the user-interface itself.
2. The number of entities which can be mapped automatically and successfully.
3. The subjective quality (or cognitive value) of the resulting visualizations.

In this paper, we choose to investigate point 3 as it is deemed to be the highest priority. This is done by conducting an informal usability test to measure the cognitive value of three different visualizations. The user test was conducted according to the principles of discount usability engineering[16].

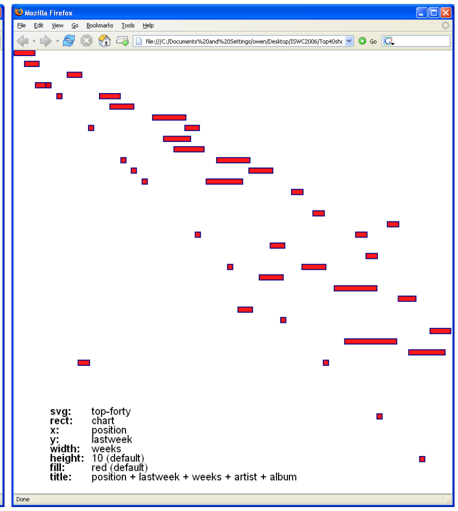
We conducted an informal user evaluation on 6 subjects who came from a technical but non-computer science background. The purpose of this test was to evaluate how well the VizThis approach can produce cognitively useful visualizations with varying levels of human involvement (thus measuring the value of the semantic assistance which is provided by the system). The first visualization (figure 3a) was produced by the system with no human involvement. The second visualization (figure 3b) was produced with a human user performing some data cleansing. The third visualization (figure 3c) 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[2] XML feed[3]. A fragment of this showing the first song is below:

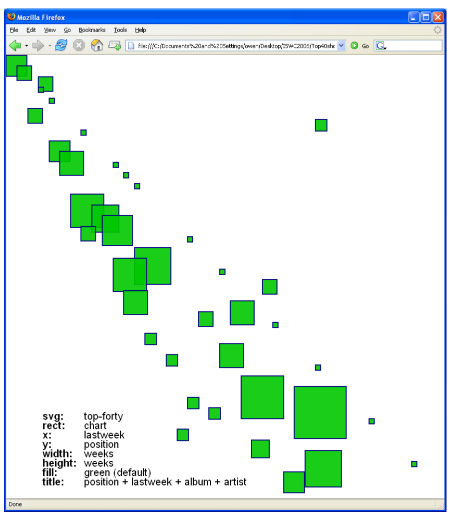
```
<top-forty>
    <chart position="1">
        <lastweek>1</lastweek>
        <weeks>(5)</weeks>
        <image>http://www.bbc.co.uk/radio1/media/images/artists/gnarlsbarkley/
        060323_cd_crazy_70.jpg</image>
        <artist>Gnarls Barkley</artist>
        <album>Crazy</album>
        <uri>http://www.gnarlsbarkley.com/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



(a) No human intervention



(b) Data cleansed only



(c) Data cleansed and mappings tweaked

Fig. 3. BBC Top 40 chart music Information Visualizations

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 the mapping table of data entities to representation artefacts (shown in the bottom left corner of each visualization).

Visualization A The following cognitive insights were observed by subjects:

- 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.

Visualization B Subjects also noticed the following:

- Five subjects noticed small squares exactly on the line x=y representing new entries.
- Five subjects noticed that different widths of the bars indicated the number of weeks the song has been in the charts.

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 by the system. We believe that through further development and added heuristics, we can increase the value of VizThis towards the level offered by visualization mappings created by humans.

6 RDF/OWL and Graph Based Formats

In this paper, we have given worked examples of the visualization of *tree based* information using the VizThis interaction metaphor. However, for the purpose of developing user interfaces for the Semantic Web, the ability to deal with *graph based* data formats such as RDF/OWL is of paramount importance.

There are two main methods in which tools can deal with mapping graph based data. The first generation of mapping tools used an expandable/collapsable tree-based metaphor. Example tools include Protégé[17], OntoEdit[20] and KAON [15]. These were later superseded by second generation tools which used nodes to represent each object, with nodes linked via weighted connections (arcs). These are also known as net based user interfaces and are featured in the evolution of the KAON infrastructure as implemented in MAFRA toolkit[19]. This technique represents nodes in a far less constrained manner which better fits with the schema of graph based data sets. However, these interfaces can quickly become very crowded with many node links complicating the display.

We propose to keep the tree-based interaction metaphor and handle graph based data formats such as RDF by first converting them into a tree based format such as XML. For RDF, this conversion can be done using a tool such as NormKit (a component of the Harmonise Mapping Framework[4]) which has the ability to convert RDF data into XML. The advantage of this approach is that it provides users with a simple and familiar user interface while still keeping the majority of the semantics conveyed by the original RDF. However, there are two disadvantages with this approach:

1. No Circular References. An object cannot reference itself, either directly or via a series of other objects. If this were to happen, then the tree would become infinitely deep.
2. Entity Repetition. If an object is linked more than once, its data will be repeated. This causes problems with data integrity and also is inefficient on storage space.

The two problems listed above do not pose a significant challenge to VizThis. Firstly, if a circular reference is encountered, the system will only create data for one iteration "around the circle", thus allowing all data to be presented, but not at the expense of infinite depth. Secondly, for the entity repetition problem, since information visualization is a read-only task, then the integrity of the data is not at stake.

For these reasons, we believe that a tree-based interaction style for visualizing graph based information would provide a suitable interface in VizThis. We will be addressing this in future work.

7 Conclusions

In this paper, we have described a user interaction paradigm which has been created from the application of Ontology Mapping techniques to the area of Information Visualization. Through the formalisation of this process with Ontology Mapping techniques, we believe that a number of advantages can be achieved. These advantages reduce the work burden on the user who is creating the visualization and ultimately may allow us to realise the goal of "Tailorable Information Visualization". The advantages are:

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 rules. 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 markup language, whether it is tree or graph based, a number of different source formats from many domains can be supported. This generality also applies to the target format used.

Multimedia 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 Ontology Mapping:

Specificity When we view Information Visualization using Ontology 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, RDF/OWL). 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.

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.

In summary, the VizThis interaction metaphor has produced some encouraging results which have been evaluated in a small-scale, informal, qualitative user evaluation. We believe that with further work the technique can evolve into a more robust and practical front-end interface to the Semantic Web.

Acknowledgments

This work took place during a research fellowship at the Knowledge Engineering and Decision Support Research Group (GECAD) at the Polytechnic Institute of Porto, Portugal. This research is funded by a University of Wales Swansea Postgraduate Research Studentship under the supervision of Dr. Phil Grant and Prof. Min Chen. The authors would also like to acknowledge FCT, FEDER, POCTI, POSI, POCI and POSC for their support to R&D projects and the GECAD unit.

References

1. Altova mapforce. http://www.altova.com/products/mapforce/data_mapping.html, 2006.
2. Bbc - radio 1 - top 40 music chart. http://www.bbc.co.uk/radio1/chart/singles.shtml, 2006.
3. Bbc - radio 1 - top 40 music chart - xml feeds. http://viper.bsg.webcoding.co.uk/top40.xml, 2006.
4. Harmonise mapping framework. http://www.sourceforge.net/projects/hmafra, 2006.
5. Spotfire dxp. http://www.spotfire.com, 2006.
6. Claudio Riva Alexandru C. Telea, Alessandro Maccari. An open toolkit for prototyping reverse engineering visualization. In *IEEE EG VisSym '02. Eurographics, 2002*, 2002.
7. Geert-Jan Houben Alexandru C. Telea, Flavius Frasincar. Visualizing rdf(s)-based information. In *Procs of IV"2003*, Hawthorne, NY, 2003.
8. Isabel F. Cruz. Tailorable information visualization. *ACM Computing Surveys*, 28(4), Dec 1996.
9. AnHai Doan, Pedro Domingos, and Alon Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. pages 509–520, Santa Barbara (CA), USA, May 2001. Santa Barbara (CA), USA.
10. AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Learning to map ontologies on the semantic web. Honolulu, Hawaii, USA, May 2002. Honolulu, Hawaii, USA.
11. Vladimir Geroimenko and Chaomei Chen, editors. *Visualizing the Semantic Web*. Springer-Verlag, London, UK, 2006.
12. Owen Gilson. Vizthis appendix. http://swui.semanticweb.org/swui06/posters/Gilson/.
13. Ben Shneiderman Jock Mackinlay, Stuart K. Card. *Readings in Information Visualization : Using Vision to Think*. Morgan Kaufmann, 1st edition, January 1999.
14. 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.
15. Boris Motik, Alexander Maedche, and Raphael Volz. A conceptual modeling approach for building semantics-driven enterprise applications. Lecture Notes in Computer Science, pages 1082–1099, Irvine (CA), USA, Oct 2003. Irvine (CA), USA, Springer.
16. Jakob Nielsen. Guerrilla hci: Using discount usability engineering to penetrate the intimidation barrier. http://www.useit.com/papers/guerrilla_hci.html, 1994.
17. Natalya F. Noy, Ray W. Fergerson, and Mark A. Musen. The knowledge model of protege-2000: Combining interoperability and flexibility. volume 1937 of *Lecture Notes in Computer Science*, pages 17–32, Juan-les-Pins, France, Oct 2000. Juan-les-Pins, France, Springer-Verlag.
18. Natalya F. Noy and Mark A. Musen. Anchor-prompt: Using non-local context for semantic matching. pages 63–70, Seattle (WA), USA, Aug 2001. Seattle (WA), USA.
19. Nuno Silva and Joao Rocha. Multi-Dimensional Service-Oriented Ontology Mapping. *International Journal of Web Engineering and Technology*, 2(1):50–80, 2005.
20. York Sure, Mechael Erdmann, Jrgen Angele, Steffen Staab, Rudi Studer, and Dirk Wenke. Ontoedit: Collaborative ontology development for the semantic web. volume 2342 of *Lecture Notes in Computer Science*, pages 221–235, Sardinia, Italia, Jun 2002. Sardinia, Italia, Springer-Verlag.