Semantics-based Color Assignment in Visualization

Martín L. Larrea
Departamento de Cs. e Ing.de la Computación, Universidad Nacional del Sur
Av. Alem 1253, B8000CPB Bahía Blanca, ARGENTINA
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
E-mail: mll@cs.uns.edu.ar
Sergio R. Martig
Departamento de Cs. e Ing.de la Computación, Universidad Nacional del Sur
Av. Alem 1253, B8000CPB Bahía Blanca, ARGENTINA
E-mail: srm@cs.uns.edu.ar
Silvia M. Castro
Departamento de Cs. e Ing.de la Computación, Universidad Nacional del Sur
Av. Alem 1253, B8000CPB Bahía Blanca, ARGENTINA
E-mail: smc@cs.uns.edu.ar

ABSTRACT

The active use and manipulation of visual representations makes many complex and intensive cognitive tasks feasible. A visual representation is able to convey relationships among many elements in parallel and it provides an individual with directly observable memory. A successful visualization allows the user to gain insight into the data, that is, to communicate different aspects of the data in an effective way. Even with today's visualization systems that give the user a considerable control over the visualization process, it can be difficult to produce an effective visualization. To obtain useful results, a user has to interrogate the visualization very precisely. A strategy to improve this situation is to guide the user with the selection of the parameters involved in the visualization. This paper presents the initial effort dedicated to achieve a visualization system that assists the user in the configuration and preparation of the visualization by considering both the semantic of the data and the semantic of the stages through all the visualization process. In this article we present a visualization system for file hierarchies where color assignment is made by a reasoning process through the use of an ontology. This work sets the way forward to integrate the visualization process with a reasoning process and configure a visualization based on the reasoner's results.

Keywords: Semantic, Visualization, Ontology, Color Assignment, RDF, OWL

1. INTRODUCTON

Computer technology allows the visual exploration of big information resources ([1], [2], [3]). Huge amount of data is becoming available on networked information systems, ranging from unstructured and multimedia documents to structured data stored in databases. This is extremely useful and exciting; but the ever growing amount of available information generates cognitive overload and even anxiety, especially in novice or occasional users. While computational power has increased exponentially, the ability to interact with useful information ([4]) has only increased linearly. In recent decades, such exponential increase in computing power has allowed to address much more complex and varied problems. Information is now massive, disparate, and disorganized. The dimensionality of data has also increased, requiring greater effort to identify and comprehend relationships relevant to a particular analytic task. Nowadays, a wide diversity of users access, extract, and display information that is distributed on various sources, which also differ in type, form and content. In many cases, the user has an active control over the visualization process, but even then, it is difficult to achieve an effective visualization. For example, because each visualization must provide a representation which helps to interpret the data or to communicate its meaning, it is important that the mapping from physical to perceptual dimensions to be under control. A strategy to improve this situation is to guide the user in the selection of the different parameters involved in the visualization. The Visualization field has matured substantially during the last decade; new techniques have appeared for different data types in many domains. With the use of visualization becoming more generalized, a formal understanding of the visualization process is needed ([5]). This work presents, at the moment of its publication, the first association between a visualization process and a reasoning process. Through an ontological reasoning we can determine the color of a visual object. In our case, this object is a node in a 3D tree visualization that represents a file system in a computer.

The remainder of this paper is structured as follows. In the next section we introduce some important concepts from the Semantic Web area. Section 3 gives the foundation's details for our research. On Section 4 the previous work is detailed and Section 5 begins with a description of our semantics-based color assignment model, including a brief description of the visualization application used to test it. Finally, Section 6 summarizes the work providing some closing remarks and directions for future work.

2. SEMANTIC WEB

The Semantic Web ([6]) is an evolving extension of the World Wide Web in which the semantics of information and services on the web is defined. This enables the Web to understand and to satisfy the requests of people and machines to use the web content.

At its core, the semantic web encompasses a set of design principles, collaborative working groups, and a variety of enabling technologies. Some elements of the semantic web are expressed in formal specifications. Some of these include the Resource Description Framework (RDF), a variety of data interchange formats (e.g. RDF/XML, N3, Turtle, N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain. The Web Ontology Language is a language to define and instantiate Web ontologies. Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related.

Ontologies

An ontology defines a common vocabulary for researchers who need to share information in a domain ([7]). It includes machine-interpretable definitions of basic concepts in the domain and relationships among them. The Artificial-Intelligence literature contains many definitions of ontology; many of them contradict each other. For our purposes an ontology is an explicit formal description of concepts (called classes) in a domain of discourse, properties of each concept describing various features and attributes of the concept, and restrictions over these properties. An ontology, in conjunction with a set of individual instances of classes, constitutes a knowledge base.

Classes are the focus of most ontologies. Classes describe concepts in the given domain. For example, a class *shape* represents all the shapes. Specific shapes are instances of this class. A class can have subclasses which represent concepts that are more specific than the superclass. For example, we could divide the class of all shapes into circles, squares, and equilateral triangles. Alternatively, we can divide the class of all shapes into regular and irregular ones. Properties of classes and instances describe their characteristics and the relationships between them: a particular triangle can have a base size of 10 cm and a height of 15 cm. We could have two properties describing the triangle in this example: the properties base and height with the value 10 and 15 cm respectively. When a property value is of type *boolean, float, int, string, date, datetime* or *time* we called it *datatype property*. If the property value is a class instance, then we called it *object property*. Object properties are useful to establish relationships between classes and instances. Each property has a domain and a range; the allowed values for object properties are often called "the range of the property". The classes to which a property is attached are called the domain of the property. Ontology developers have adopted OWL as the language for ontology description.

OWL

OWL ([8]) stands for "Web Ontology Language" and is a language for processing web information. OWL is built on top of RDF ([9]) and it is written in XML. By using XML, OWL information can easily be exchanged between different types of computers using different types of operating system and application languages. An OWL ontology contains a sequence of annotations, axioms, and facts. Annotations on OWL ontologies can be used to record authorship and other information associated with an ontology, including imports references to other ontologies. The main content of an OWL ontology is carried in its axioms and facts, which provide information about classes, properties, and individuals in the ontology.

Semantic Reasoner

A semantic reasoner is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language, and often of a description language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining or backward chaining. For example, if an ontology gives the following information:

- All football players who won the 1986 world cup were men.
- Maradona was in the team that won the 1986 football world cup.

The reasoner may generate the conclusion: "*Maradona is a man*". This conclusion can be reached even though there is no explicit statement in the ontology of the fact that Maradona is a man.

3. SEMANTIC BASED VISUALIZATION

The user is an active participant in the visualization process and the goal of a visualization is to present data in a way that helps him to identify trends, features and patterns, generate hypotheses, and assign meaning to the visual information on the screen. Our main goal is the development of a visualization model that considers the semantics of both the data and the different stages in the visualization process. This model will transform data into information; according to Keller and Tergan ([10]), "information is data that has been given meaning through interpretation by way of relational connection and pragmatic context". The information is the same given the same meaning. This "meaning" can be useful, but does not have to be. Information may be distinguished according to different categories concerning, for instance, its features, origin and relations. By making these considerations, the visualization process will be able to determine the characteristics of an effective visualization and guide the user through the different stages. Our main goal is to define an unified semantics for the data model and the process involved ([11], [12], [13]). On Section 5 we describe how we transform color assignment rules into semantic information and use them to determine the color of some visual elements.

4. PREVIOUS WORK

The papers [14], [15], [16], [17] and [18] are good examples of how semantic information is integrated into the visualization tasks. However, in all these examples the role of the semantics is to improve the integration, querying and description of the visualization data; in neither case the semantics associated with the data is used to create the visualization or define its attributes. Only in [19] we can find a first approach to the use of the semantics as an aid to create a visualization. This work defines a customizable representation model which allows biologists to change the graphical semantics associated to the data semantics. The representation model is based on an XML implementation and uses an XML Schema definition that prescribes its correctness and provides validation features. Unfortunately this work is only intended for biological use; it does not take advantage of the RDF or OWL representation and does not include any reasoning process with the semantic information.

5. SEMANTICS-BASED COLOR ASSIGMENT

Our goal is to define an ontology to describe the relationship between the data elements and the colors associated to their representation. We begin by defining the class *Color*; this represents the colors that we will be used to paint the visual elements. We utilize the RGB system to represent each color; hence, there are three datatype properties in this class, *red, green* and *blue.* These properties are of type integer with values between 0 and 255. To represent the data elements, we define the class *Data*. This class has two datatype and one object property; the datatype properties are *description,* a short description about the data element, and *filetype,* the file extension associated with the data element. The link between the *Data* and the *Color* classes is made through a class called *VisualRepresentation.* These relationships can be read as: *"Every data element has a visual representation which has a color property".* Of course, there are many others properties of a visual representation, e.g. shapes, but we will address them in Section 6. Now, in order to link the three classes we need object properties. As we said, the *Data* class has one object property which is called *inRepresentation*. This property is a relationship between the concepts *Data* and *VisualRepresentation.* In *VisualRepresentation* we have the object property *inColor* which links a visual representation with a color. Figure 1 gives an overview of the described concepts and relationships.

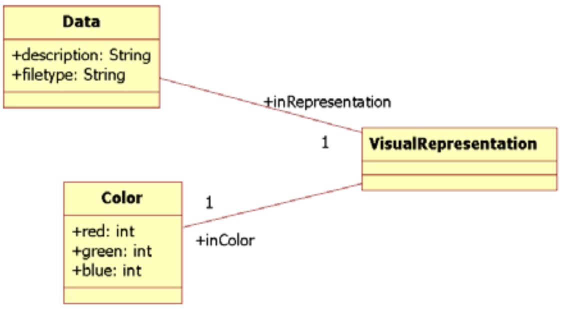

Figure 1. Concept diagram from the ontology. It represents the statement: *"Every data has a visual representation which has a color property*". This ontology is available at [20].

We need, in order for the reasoner to work with the ontology, to add more information, particularly the proper information for each file type that we wish to include in the visualization and its associated color. For each file type we will create a new subclass of *Data* and necessary and sufficient condition ([21]) on the new class in the form of *"filetype has extension*" where *extension* is the file extension that we are representing and *filetype* is the datatype property inherited from *Data*. Now, suppose that we want to represent "pdf" files with red. We have to create a new subclass of *Data* called *PDF* (for our purposes, classes names are not important but we will try to make them as meaningful as possible). In this new class we set a new necessary and sufficient condition in the form of *filetype has "pdf*"; in the description property we can load the *"PDF Document"* value. We will also create a new subclass of *Color* called *RED*; its datatype properties values will be 255 for *red*, 0 for *green* and *0* for *blue.* The next step may not seem as trivial as the previous ones. So far we have created two subclasses, one for *Data* and the other for *Color* and we have established that they are related through the *VisualRepresentation* class. Now we need to create a subclass of *VisualRepresentation*, this new class will be the link between the classes *PDF* and *RED*. We will call it *VisualRed*. We also need to create subproperties of *inRepresentation* and *inColor*, that is *inRepresentationRed* and *inColorRed* respectively. The domain of *InRepresentationRed* will be the *PDF* class and its range *VisualRed.* The domain of *inColor_Red* will be *VisualRed* and its range *Red.* On Figure 2 we can see the ontology model with the new subclasses added.

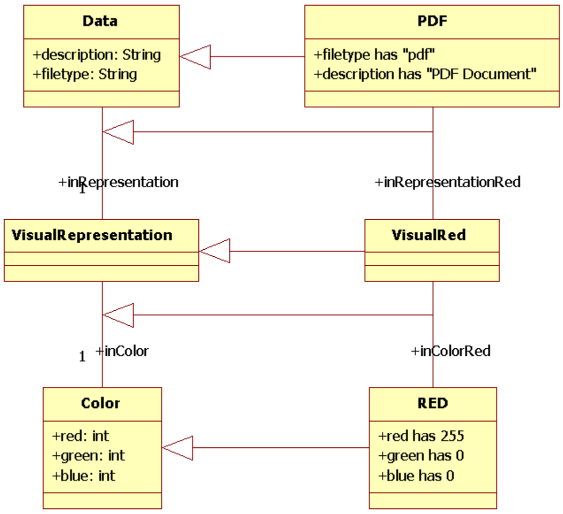

Figure 2. Ontology model as seen in Figure 1 with the inclusion of a file type "pdf" that will be represented in color red. Three new subclasses must be created and two new subproperties. If we want to add a new file type with a new color, we have to create another 5 elements.

For every new file type *ftype* that we want to represent using a color *c*, we must create three new subclasses, one for *ftype*, one for *c*, and finally, one for the visual representation that join them together. We will also have to create two new subproperties, one from *inRepresentation* and the other from *inColor.*

Once the ontology is set, we focus on how to use a reasoner to obtain the associated color of a file extension. The following steps describe this process. We will assume that we want to know the color in the visualization corresponding to the file extension represented by the variable *ftype*.

We started creating a new subclass of the *Data* class, and called it *temp.* Then we added a necessary and sufficient condition to *temp* in the form of *"filetype has ftype"* where *filetype* was the datatype property of the *Data* class. Once this is set, we asked the reasoner to get the inferred subclasses. In our ontology, if there is a class defined for the *ftype* extension, the reasoner will return such a class. Otherwise the result will be the class *owl:Nothing.* If the returned class (called *Cftype)* is different from *owl:Nothing* then we can get its associated color by following the object properties *inRepresentation* and *inColor.*

Brows.AR application

We developed Brows.AR (Figure 3) an application for the visualization of file hierarchies in 3D based on the Spherical layout ([22]). The Spherical layout is a 3D generalization of the Radial layout. Instead of circles, as in Radial layout, we consider concentric spheres, on whose surfaces we locate the nodes. In the Radial layout each node, except the root, is allocated in a 2D sector within the sector assigned to its parent; in the Spherical layout we consider a spherical wedge and the nodes are allocated on the surfaces defined by this wedge. With this application we create a 3D representation of a directory structure; to enrich the visual representation, we allowed the user to see the triangles that were used to place the nodes; these triangles are painted with the same color used for the node but with a high level of transparency. Node's color is based on the file type that the node represents. In case of very large trees, it is possible to remove the nodes and edges from the visual representations and leave only the triangles, providing an overview of the hierarchical structure and improving the application performance. For details about its implementation and interactions see [23].

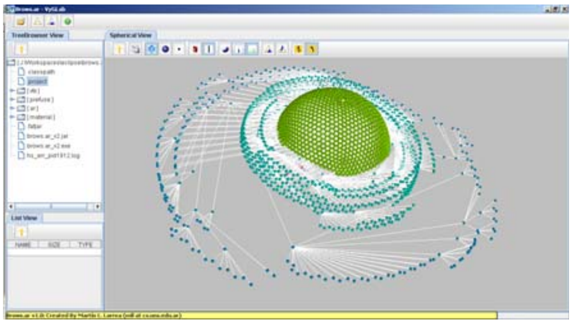

Figure 3. Brows.AR screenshot. A file hierarchy visualization tool based on the Spherical Layout ([22], [27]).

Brows.AR ontology add-on

In order to integrate the semantic information with our application we created a class called *Reasoner;* its main method is *ask.* This method takes a file extension as input and returns a color. This color is used to paint the visual element representing a file with the given file extension. The *Reasoner* class uses Protégé ([24]) and Jena ([25]) APIs to interact with the ontology. The reasoning service was provided by the Pellet ([26]) API. The *ask* method turns into code the steps described earlier in this section. The constructor of the *Reasoner* class takes one parameter, a *JenaOwlModel* which is a representation of an ontology model.

To improve performance we used a hash table as a cache memory to keep the information retrieved from the reasoner. If a particular filetype *ftype* is not in the cache, the application asks the reasoner for the associated color *c.* Then the pair *(ftype, c)* is saved in the cache and *c* is returned.

Figure 4 shows part of the main code from the *Reasoner* class. Lines 1 – 3 create a temporary new subclass from *Data* with the name *temp*. Lines 4 – 7 add the necessary and sufficient condition in the form of "*filetype has ftype*" using the *OWLHasValue* interface from Protégé. Line 8 asks the reasoner for the inferred subclass of the temporary class and lines 9 – 10 retrieve such class.

6. CONCLUSIONS

We have designed an ontology model for the assignment of color to visual representations. This model was integrated in the Brows.AR application, a 3D visualization system for file hierarchies. This tool combines both the visualization and the reasoning processes through the use of semantic information. Undoubtedly, the great benefit of this combination is the definition of an unified semantics for the data and the visualization process, in order to create a visualization system that will be able to assist the user in the preparation and configuration of the visualization. This visualization system should ensure that, even if the user is not an expert in Visualization, the generated visualization will be the most suitable for the user and the data domain.

```
1. OWLNamedClass DataClass =
        owlModel.getOWLNamedClass("Data");
2. OWLNamedClass tempSubClass =
        owlModel.createOWLNamedSubclass("temp", DataClass);
3. OWLIntersectionClass t =
        owlModel.createOWLIntersectionClass();
4. OWLDatatypeProperty fileTypePropery =
        owlModel.getOWLDatatypeProperty("filetype");
5. OWLHasValue hs =
        owlModel.createOWLHasValue(fileTypePropery, filetype);
6. t.addOperand(hs);
7. tempSubClass.setDefinition(t);
8. Collection inferredSubclasses =
        this.pelletReasoner.getSubclasses(tempSubClass);
9. Iterator it = inferredSubclasses.iterator();
10. OWLNamedClass inferredSubclass =
        (OWLNamedClass) it.next();
```

Figure 4. Main code segment from the *Reasoner* class. Lines 1 – 3 create a temporary new subclass from *Data*. Lines 4 – 7 add the necessary and sufficient condition in the form of "*filetype has ftype*". Line 8 asks the reasoner for the inferred subclass of the temporary class and lines 9 – 10 retrieve such class.

Although for this color-assignment case the class *VisualRepresentation* is not necessary, we incorporated it into the ontology in order to add new features to the visual representation. The work done at [27] reviews the elements of visual variables and its components and it is our purpose to add these components to the ontology in the future. We are also looking at the inclusion of the concepts of transparency, colormaps and internationalization features as described in [13], and the support for different color representation systems other than RGB, such us HSL, HSV, etc.

7. ACKNOWLEDGEMENT

This research was partially supported by the PGI 24/N020 and 24/ZN19, Secretaría General de Ciencia y Tecnología, Universidad Nacional del Sur, Bahía Blanca, Argentina.

8. REFERENCES

[1] Jacques Bertin. "Semiology of graphics". University of Wisconsin Press, 1983. ISBN: 0299-090-604.

[2] Edward R. Tufte. "The visual display of quantitative information". Graphics Press, Cheshire, CT, USA, 1986. ISBN:0-9613921-0-X

[3] William S. Cleveland. "The elements of graphing data". Wadsworth Publ. Co., Belmont, CA, USA, 1985. ISBN:0-534-03730-5

[4] Ji Soo Yi, Youn ah Kang, John Stasko and Julie Jacko: "Toward a Deeper Understanding of the Role of Interaction in Information Visualization", Volume 13, Issue 6. Pages 1224-1231. 2007. ISSN: 1077-2626

[5] 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, page 598.7, Washington, DC, USA, 2004. IEEE Computer Society.

[6] W3C Semantic Web Activity. 28 April 2009. http://www.w3.org/2001/sw/

[7] Natalya F. Noy and Deborah L. McGuinness. "Ontology Development 101: A Guide to Creating Your First Ontology". Stanford University. http://ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf

[8] "OWL - Web Ontology Language Semantics and Abstract Syntax". W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-semantics/

[9] "Resource Description Framework (RDF) Model and Syntax Specification". W3C Recommendation 22 February 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[10] Tanja Keller and Sigmar-Olaf Tergan. Visualizing knowledge and information: An introduction. In Sigmar-Olaf Tergan and Tanja Keller, editors, *Knowledge and Information Visualization*, volume 3426 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2005.

[11] Martín Larrea, Sergio Martig and Silvia Castro, "Semantic Based Visualization". Proceedings of the VIII Workshop de Investigadores en Ciencias de la Computación. 2006, pages 473-475. ISBN: 950-9474-355.

[12] Martín Larrea, Sergio Martig and Silvia Castro, "Semantic Based Visualization". Proceedings of the IX Workshop de Investigadores en Ciencias de la Computación. 2007, pages 289-293. ISBN: 978-950-763-0750.

[13] Martín Larrea, Sergio Martig and Silvia Castro, "Semantic Based Visualization: A first approach". Proceedings of the X Workshop de Investigadores en Ciencias de la Computación. 2008, pages 289-293. ISBN: 978-950-863-101-5

[14] Zhao Xu, Huajun Chen, and ZhaohuiWu. "Applying semantic web technologies for geodata integration and visualization". In Jacky Akoka, StephenW. Liddle, Il-Yeol Song, Michela Bertolotto, Isabelle Comyn-Wattiau, Samira Si-Said Cherfi, Willem-Jan van den Heuvel, Bernhard Thalheim, Manuel Kolp, Paolo Bresciani, Juan Trujillo, Christian Kop, and Heinrich C. Mayr, editors, ER (Workshops), volume 3770 of Lecture Notes in Computer Science, pages 320–329. Springer, 2005.

[15] V Kashyap, K-H Cheung, D Doherty, M Samwald, MS Marshall, J Luciano, S Stephens, I Herman, and R Hookway. "An ontology-based approach for data integration - an application in biomedical research". In J Cardoso, M Hepp, and M Lytras, editors, Real-world Applications of Semantic Web Technology and Ontologies, SemanticWeb and Beyond. Springer-Verlag, Heidelberg, 2008.

[16] Evangelos Kalogerakis, Stavros Christodoulakis, and Nektarios Moumoutzis. "Coupling ontologies with graphics content for knowledge driven visualization". In VR '06: Proceedings of the IEEE conference on Virtual Reality, pages 43–50,Washington, DC, USA, 2006. IEEE Computer Society.

[17] Trong Dung Nguyen, Tu Bao Ho, and DucDung Nguyen. "Data and knowledge visualization in knowledge discovery process". In VISUAL '02: Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems, pages 311–322, London, UK, 2002. Springer-Verlag.

[18] Z. M. Weng and D. Bell. "Integrating visual ontologies and wavelets for image content retrieval". In DEXA '98: Proceedings of the 9th International Workshop on Database and Expert Systems Applications, page 379, Washington, DC, USA, 1998. IEEE Computer Society.

[19] Mokrane Bouzeghoub, Carole A. Goble, Vipul Kashyap, and Stefano Spaccapietra, editors. "Semantics for Grid Databases", First International IFIP Conference on Semantics of a Networked World: ICSNW 2004, Paris, France, June 17-19, 2004. Revised Selected Papers, Volume 3226 of *Lecture Notes in Computer Science*. Springer, 2004.

[20] Ontology model for Brows.AR application. "Every data has a visual representation which has a color property". Martín Larrea Website. Available for download at: http://cs.uns.edu.ar/~mll/temp/2009-martinlarrea-browsar-colorassigment-001.owl

[21] Guus Schreiber. "OWL restrictions". http://www.cs.vu.nl/~guus/public/owl-restrictions/

[22] Martín Larrea, "Thesis Overview: Spherical Layout for 3D Graph Visualization". Journal of Computer Science and Technology, 2006, volumen 7, number 1, pages 112-113. ISSN: 1666 6038.

[23] Martín Larrea, Sergio Martig and Silvia Castro, "Spherical Layout – Layout for 3D Tree Visualization". Proceedings of the IADIS 2007, Multi Conference on Computer Science and Information Systems, pages 91-98. ISBN: 978-972-8924-39-3.

[24] Protégé Web Site. Last visited on 28/04/2009. http://protege.stanford.edu/

[25] Jena – A Semantic Web Framework for Java. Last visited on 28/04/2009. http://jena.sourceforge.net/

[26] Pellet: The Open Source OWL DL Reasoner. Last visited on 28/04/2009. http://clarkparsia.com/pellet

[27] Carpendale, M. S. T.: "Considering Visual Variables as a Basis for Information Visualisation", University of Calgary, Department of Computer Science, 2001-693-16, 2003

Received: June 2008. Accepted: Nov. 2009