A Semantics-based visualization building process

Martín Larrea[1,2], Sergio Martig[2], and Silvia Castro[2]

[1]Consejo Nacional de Investigaciones
Científicas y Técnicas (CONICET)
[2]Laboratorio de Investigación y Desarrollo en Visualización y
Computación Gráfica, (VyGLab)
Departamento de Ciencias e Ingeniería de la Computación,
Universidad Nacional del Sur,
Av. Alem 1253, Bahía Blanca, Argentina.
`{mll,srm,smc}@cs.uns.edu.ar`

Abstract. A successful visualization allows the user to gain insight into 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. This paper is a step forward 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 system for file hierarchies visualization where the color assignment and the configuration of the visualization technique are carried out by reasoning processes. This work sets the way forward for the integration of reasoning in the visualization process.

Keywords: Semantic, Visualization, Ontology, Spherical Layout, Color Assignment, Reasoning, RDF, OWL

1 Introduction

Computer technology allows the visual exploration of large information resources ([1]). 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. Today, a wide range of users access, extract and display information that is distributed over several sources, which also differ in type, structure 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. 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 decades; 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 ([3]). This work improves the one presented in [11] by including explicitly the semantic of the hardware, the user and the tasks in the visualization process. Our contribution is a new step forward to achieve a visualization system that assists the user in the configuration and preparation of the visualization. Through a semantic reasoning we can determine all the parameters needed for the creation of a visualization. In our case we considered the visualization of a file system using the Spherical Layout ([13]). The remainder of this paper is structured as follows. In the next section we give the foundation's details for our research. In Section 3 the previous work is detailed and Section 4 describes our semantics-based visualization creation model, including a brief description of the visualization application used to test it. In this section we consider the semantics of the data, the hardware, the user and the task. Finally, Section 5 summarizes the work providing some closing remarks and directions for future work. Because of space limitations we have not included an introduction to the Semantic Web and semantic reasoner terminology. For details about these concepts please see [11].

2 Semantics-based Visualization

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 ([4]), 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. 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. 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. Since 2006 we have been working on the integration of semantic information into the visualization process ([10], [11]) and our main goal is to define an unified semantics for the data model and the process involved. In Section 4 we describe the semantics defined and the ontologies that represent them. In this section we also show how we created a visualization by using the results from the semantic reasoner and the ontologies.

3 Previous Work

There are some good examples ([5], [6], [7] and [8]) of how semantic information is integrated into the visualization tasks. However, in all these cases 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 [9] 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.

4 Semantics-based Visualization Creation

A successful visualization allows the user to gain insight into the data. A successful visualization process takes advantage of the structure and the meaning of the data to create the most effective visualization. The structure of the data can be obtained from the data itself but not its meaning. Two sets may contain the same data, but if its meaning is different then the final visualizations will not necessary be the same. This is why we included the semantic about the data, a way to describe the data about the data.

A visualization is greatly affected by what the user want to do with it. For the same data set, also with the same meaning, one visualization may be most suitable for data exploration and another may be better for data comparison. By knowing what the user want to do and its meaning the visualization designer can create a better result. This is our reason to incorporate the semantic about the tasks.

Additionally, the response time of the interactions its crucial to obtain an effective visualization. If the user want to explore a 3D visual representation but there is no dedicated GPU on the computer, the user's experience would be negatively affected. Besides that, a 4 inches screen can not represent a visualization in the same way that a 42 inches screen does. A formal description of the system's hardware could help the visualization designer to enhance the user experience with the visualization. Then, in addition to the data and task semantics, we also included the semantic of the hardware, a description of the actual system's hardware.

All the previous semantics can be taken as input to the visualization process. All of them can change from one visualization process to another. But the visualization process can contain its own semantics as previous knowledge embedded in the system. The goal of this is to help the user in the decisions that depends on knowledge outside of the user scope. For instance, which colors combine better or which colormap to use to represent a data attribute. To demonstrate this we included the semantic of color.

This justify to extend our previuos work ([11]) by improving our system's architecture. Our previous work only included the color assignment process, but now we have also considered the rest of the visualization process, specifically the visualization technique configuration. We added new ontologies and included new steps where to applied the reasoning process. As in our previous work, we used our Brows.AR application as test case.

In the next paragraphs we describe in detail these semantics and how we created the ontologies representing them. Then we detail how the reasoning process uses these semantics in the visualization process. A review of our architecture can be seen in Fig. 1. We end this section with the description of how we adapted these elements to the Brows.AR application.

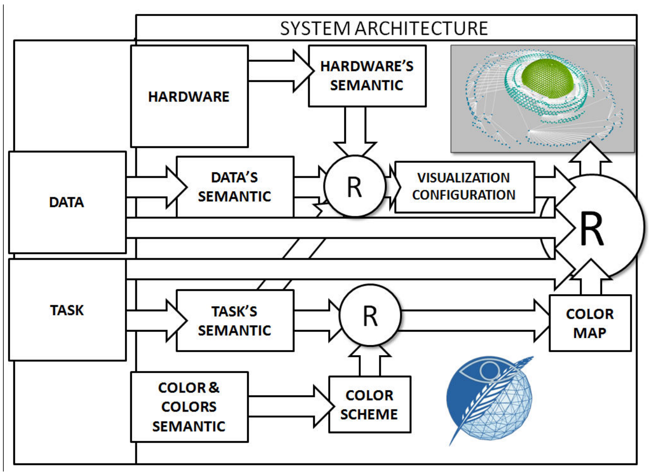


Fig. 1. The implemented system architecture.

4.1 The semantic of the data

We created the semantic of the data based on metrics about the information to visualize. These metrics can give us information about the data itself. Because we used Brows.AR as a test case, a file hierarchies visualization tool, our metrics are tree oriented. Our Data ontology contains 5 metrics. All these metrics are data properties on a concept name *Metric*.

Number of items (n), in this test case the number for folders and files.
Height of the tree (h), number of items on the longest path from the root to a leaf.
Width of the tree (w), maximum number of items on a level of the tree.
Ratio of the tree, Height/Width, (r).
Bounding box of the tree, Height*Width, (bb).

Because this is a test case, we limited the content of this semantic to include only metrics. The data's structure is implicit in the test case; a file system as a tree hierarchy. Each data element is a nominal one.

4.2 The semantic of the tasks

In our previous work ([11]) we showed how the color assignment could be accomplished by a reasoning process. In this paper, we extend that work to incorporate the semantic of the visualization tasks ([12]). For simplification we consider only one task, *filter*. As described in [12], *filter* is defined as: given some conditions on attributes values, select data cases satisfying those conditions. In our case, we use color to highlight those cases, that is to change the color of the visual elements that are selected by the filter. Our goal was to describe, through an ontology, how to calculate the color property on each visual element.

Importing the previous ontologies, a developer can create its own Task ontology using the concepts, relationships, properties and individuals related to these. As mentioned earlier, our task was filter with highlight, using color. To stand out an object with color it is necessary to know which is the background color and it is also important to know the color to use in the objects that will not be highlighted. To represent these elements, we include tree concepts in the Task ontology: *background*, *highlight* and *regular*. The *background* concept contains two object properties that relate to the *highlight* and *regular* concepts. *Background* represents the background color, *highlight* is the color for the filtered elements and finally *regular* is the color for the remaining elements. In order to set, in the ontology, that these last concepts are colors we establish them as equivalent to the *Color* concept.

The great benefit of this implementation is that the user is no longer responsable for the selection of the colors, a bad selection of the colormap may lead to a visualization where the highlighted elements do not seem highlighted because the contrast between the colors is not perceived. A novice user may choose colors based on what he/she thinks look nice, but does not represents the true goal of the task.

4.3 The semantic of the hardware

As we said earlier, a visualization occurs in a context and in this case that context is the computer's hardware. The same visualization will not accomplish the same results if it is shown in a 4 screen or in a 42 screen. The effectiveness of a visualization method may depend on the available hardware and the peripheral devices attached to the computational system. The complexity of the visual elements should be adjusted based on the 3D capability of the computer to improve the response time to the interactions. The Hardware ontology contains a concept name *Hardware* which contains the following data properties.

- An indicator of whether the computer has or has not a dedicated GPU (GPU).
- The height, on pixels, of the screen resolution (hp).
- The width, on pixels, of the screen resolution (wp).
- The number of pixels on display, ($pxs = hp * wp$).
- The size, on inches, of the computer display ($inches$).

4.4 The semantic of the color

In this work, we expanded the work done in [11] to enhance the color representation. The new Color ontology contains one concept, *Color*. The role of this ontology is to express all the information related to color. The *Color* concept contains 3 data properties, *red*, *green* and *blue*. Each one of these represent its primary color component. These data properties have *integer* as range and *Color* as domain. There are two object properties with domain and range in *Color*, these are *next* and *opposite*. Based on the color wheel, it is possible to define, for each color, an opposite and a neighbor. The opposite to a color c is another color d, whereas d is facing c on the color wheel. The next to a color c is another color t which is the following one to c on the color wheel. The concept *color* can be easily extended by new data and object properties. After this, we created the Colors ontology, a separated ontology that import the previous one. The role of the Color ontology is to contain all the necessary information in order to describe a generic color. The role of the Colors ontology is to contain all the colors as individuals or instances of the *Color* concept. All individuals contain specific values for their properties. For our test case we included 18 colors in the Colors ontology.

4.5 Reasoning Process

Having established the semantic elements in our architecture, we can now show how these elements are used by a reasoner to create results that will aid in the visualization creation. We began describing the role of the reasoning process in the color assignment process. The color assignment is accomplished using the Task semantic through the reasoner. We then describe how the visualization technique is configure by the reasoner. In this stage the semantic of the hardware and data are used as input to the reasoning process. Finally we end with the description of the visualization creation *per se*.

Color Assignment Using Task Semantic. The Color and Colors Semantic contain the formal representation of a color, and all the colors as individuals. The first step for color assignment is to select a color scheme, to do this we ask the user to pick the visualization's background color. Once this color is chosen, the Color Scheme is composed of the selected background color, its opposite and its neighbor, based on the Colors ontology. There is no reasoner involved in this step. Once we have the Color Scheme together with the semantic of the task, the reasoner can create the Color Map. The Color Map represents which colors will be used and how. What the reasoner does is to take a color, the one selected by the user as background, and to see that the concept *Color*, from the Color ontology, is equivalent to the concept *Background* from the Task ontology. Thus the reasoner knows that what holds true for a *Color* and a *Background* also does for the selected color which is an instance of both concepts. The selected color has an *opposite* relationship, as specified in the Color ontology, and the reasoner knows that this is equivalent to the *highlight* in the Task ontology.

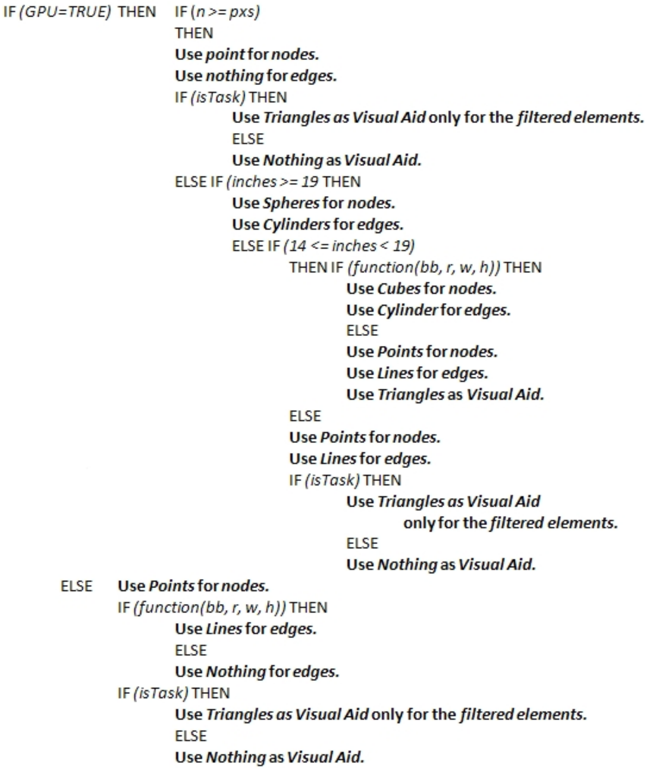
Because of this, whatever color is *opposite* to the selected one, that color is the one that will be used to highlight elements in the visualization. The same process takes place for the *regular* relationship. The next step is to create the technique configuration based on the semantic of the data, the hardware and the task.

Visualization Technique Configuration Using Semantic. The Spherical Layout technique supports different configurations of the final visualization. In our implementation of the layout there are multiple choices to graphically represent:

- **Nodes.** Nodes can be represented by a point in space, a cube or a sphere. The only visual property for points is color, so they are the less visual complex element in the technique. The sphere is the most complex visual element, followed by the cube. It is also possible not to map the nodes, visually. This give us four possibilities of representation for the nodes: not mapped, points, cubes or spheres.
- **Edges.** Edges can be represented by two type of lines, a single line whose only visual property is color and a cylinder, which allows to map more visual properties. The latest one is the most complex visual element. It is also possible not to map the edges. For edges, we have three possible representations: not mapped, lines or cylinders.
- **Visual Aids**. In this implementation of the Spherical Layout the nodes are uniformly distributed on the spheres' surfaces; to achieve this goal we first discretized the surfaces of the spheres with triangles and placed the nodes in the barycenter of some of these triangles. As a visual aid in the visualization it is possible to show such triangles.

We defined a set of rules to relate the semantics of data, hardware and task with the configuration of the visualization technique. Based on the semantics defined earlier we created the rules shown in Fig. 2; in these rules we considered that the user may or may not want to perform the task *filter*. This is represented by the condition *isTask*, that when true means to filter certain elements and otherwise means to do nothing. The *function* that appears in the set of rules determinates if it is possible to fit the tree in the visualization viewport. The result from the reasoning process in this stage is a concept called *Configuration* indicating which visual elements to use in the visualization. This set of rules allow us to control how the visualization is created to make the most out of the current hardware.

Visualization Creation The last intervention of the reasoner is in the creation of the visualization *per se*. The inputs to this process are the *Configuration* concept, the Color Map and the data itself. As we did in [11] the reasoner can decide which color should be used for each data, based on the *filter* task and the Color Map. Using the *Configuration* concept the reasoner can determinate how the elements will be shown in the visualization. For this stage we created a new concept called *DisplayElement* which represents how a data element will be



```
IF (GPU=TRUE) THEN   IF (n >= pxs)
                     THEN
                     Use point for nodes.
                     Use nothing for edges.
                     IF (isTask) THEN
                         Use Triangles as Visual Aid only for the filtered elements.
                         ELSE
                         Use Nothing as Visual Aid.
                     ELSE IF (inches >= 19 THEN
                         Use Spheres for nodes.
                         Use Cylinders for edges.
                         ELSE IF (14 <= inches < 19)
                             THEN IF (function(bb, r, w, h)) THEN
                                 Use Cubes for nodes.
                                 Use Cylinder for edges.
                                 ELSE
                                 Use Points for nodes.
                                 Use Lines for edges.
                                 Use Triangles as Visual Aid.
                             ELSE
                             Use Points for nodes.
                             Use Lines for edges.
                             IF (isTask) THEN
                                 Use Triangles as Visual Aid
                                     only for the filtered elements.
                                 ELSE
                                 Use Nothing as Visual Aid.
    ELSE     Use Points for nodes.
             IF (function(bb, r, w, h)) THEN
                 Use Lines for edges.
                 ELSE
                 Use Nothing for edges.
             IF (isTask) THEN
                 Use Triangles as Visual Aid only for the filtered elements.
                 ELSE
                 Use Nothing as Visual Aid.
```

Fig. 2. Rules created to relate the semantics of the data, the hardware and the task with the configuration of the visualization technique.

displayed. This concept includes data properties to handle the different choices for node and visual aid.

4.6 Brows.AR Application

We developed Brows.AR an application for the visualization of file hierarchies in 3D based on the Spherical Layout ([13]) 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 allow 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 the case of very large trees, it is possible to remove the nodes and edges from the visual representations and to leave only the triangles, providing an overview of the hierarchical structure and improving the application performance. For details about the implementation and interactions see [13].

4.7 Brows.AR Semantic Add-on

In order to integrate the semantic information with our application we created a class called *Reasoner*; its main method is *ask*. The *Reasoner* class uses Protégé[1] and Jena[2] APIs to interact with the ontologies. The reasoning service was provided by the Pellet[3] API. The constructor of the *Reasoner* class takes one parameter, a *JenaOwlModel* which is a representation of an ontology model. To improve the performance of the last stage in the visualization process, we used a hash table as a cache memory to keep the information retrieved from the reasoner. If a particular data element is not in the cache, the application asks the reasoner for the corresponding *DisplayElement* instance. Then the pair *(data element,* DisplayElement *instance)* is saved in the cache. Because of all the edges are handled uniformly, the reasoner asks only once about this option at the beginning of the process.

5 Conclusions

We have designed several ontology models related to the visualization creation as a representation of the semantic in the visualization process. We included the semantic of data, task, hardware and color. Within the visualization process, we used a semantic reasoner to create the final visualization. This architecture was integrated in the Brows.AR application, a 3D visualization system for file hierarchies. The benefit of this integration is the definition of an unified semantics for the visualization process, in order to create a visualization system that will be able to assist the user in the preparation and configuration of any visualization. This visualization system should ensure that, even if the user is not a visualization expert, the generated visualization will be the most suitable for that user and the data domain. This work presents a break trough in the visualization research, because of the integration between the visualization process and the knowledge on visualization creation. The used of Brows.AR as a test case prove that it is possible to use a semantic reasoner to create a visual representation.

As future work, on the semantic about data, we are looking to include Strahler number and its bifurcation ratio as part of the data's metric. The semantic of hardware will be extended to include more input & output capabilities and we will expand the semantic about the tasks to include all the tasks described in [12].

---

[1] Protégé Web Site. http://protege.stanford.edu/

[2] Jena - A Semantic Web Framework for Java. http://jena.sourceforge.net/

[3] Pellet: The Open Source OWL DL Reasoner. http://clarkparsia.com/pellet/

6 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.

References

1. Tufle, E., R.: The Visual Display of Quantitative Information. Graphics Press (1986)
2. Yi, S., J., Khang, Y., Stasko, J., Jacko, J.: Toward a Deeper Unsertanding of the Role of Interaction in Information Visualization. Volumen 13. Issue 6, pp. 1224-1231 (2007)
3. Duke, D., J., Brodlie, K., W., Duce, A., D.: Building an Ontology of Visualization. In: Proceedings of the Conference on Visualization '04. pp. 598. IEEE Computer Society (2004)
4. Keller, T., Tergan, S.: Visualizing Knowledge and Information: An Introdution. In: Knowledge and Information Visualization. pp. 1-23, Springer (2005)
5. Zhao Xu, Huajun Chen, Zhaohui Wu.: Applying semantic web technologies for geodata integration and visualization. In: Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005)
6. Kashyap, V., Cheung, K., Doherty, M. Marshall M.S., Luciano, J., Stephens, S., Herman, I., Hookway, R.: An ontology-based approach for data integration. An application in biomedical research. In: Real World Applications of Semantic Web Technology and Ontology, Semantic Web and Beyond. Springer Verlag / Heidelberg (2008)
7. Kalogerakis, E., Christodoulakis, S., Moumoutzis N.: Coupling Ontologies with Graphic Content for Knowledge Driven Visualization. In: VR '06, Proceedings of the IEEE Conference on Virtual Reality, pp. 43-50, IEEE Computer Society (2006)
8. Weng, Z., M., Dell, D.: Integrating Visual Ontologies and Wavelets for Image Content Retrieval. In: Proceedings of the 9th International Workshop on Database and Expert Systems Applications. pp. 379, IEEE Computer Society (1998)
9. Bouzeghoub, M., Goble, C., A., Kashyap, V., Spaccapietra, S.: Semantics for Grid Databases. In: First International IFIP Conference on Semantics of a Networked World. Lecture Notes in Computer Sciente, Springer (2004)
10. Larrea, M., Martig, S., Castro, S.: Visualizacin Basada en Semntica. In: Proceedings of the XII Workshop de Investigadores en Ciecnias de la Computacón. pp. 270-274. Red UNCI (2010).
11. Larrea, M., Martig, S., Castro, S.: Semantics-based Color Assignment in Visualization. In: Journal of Computer Science and Technology. pp. 14-18, ISTEC (2010)
12. Lee, B., Plaisant, C., Sims Parr, C., Fekete, J., Henry, N.: Task Taxonomy for Graph Visualization. In: Proceedings of the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization. pp. 1-5. ACM (2006)
13. Larrea, M., Martig S., Castro S.: Spherical Layout - Layout for 3D Tree Visualization. In: Proceedings of the IADIS 2007, Multi Conference on Computer Science and Information Systems. pp. 91-98 (2007).