A Semantics-based Visualization Building Process

Martín Larrea¹, Sergio Martig¹, Silvia Castro¹,

¹ 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 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 the visualization of file system hierarchies where the color assignment and the configuration of the visualization technique are carried out by reasoning processes. This work set 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 and 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 ([2]). This work improves the one presented in [3] 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 ([4]).

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 model, including a brief description of the visualization application used to test it. In this section we considered 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 reasoned terminology. For details about these concepts please see [3].

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 ([5]), “information is data that has been given meaning through interpretation by way of relational connection and pragmatic context”. 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 ([6], [3]) 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 reasoned and the ontologies.
3 Previous Work

There are some good examples ([7], [8], [9] and [10]) on 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 [11] 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 biologist to change the graphical semantics associated to the data semantic. The representation model is based on an XML implementation and used 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 necessarily 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 the 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 is 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 cannot 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 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.
Because of these considerations we extended our previous work ([3]) 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 apply 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 used 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 system hierarchy visualization tool, our metrics are tree oriented. Our Data Ontology contains 5 metrics. All these metrics are data properties on a concept name \textit{Metric}.

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

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 ([3]) 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; particularly we 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.

By 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 the background color is; and it is also important to know which color to use in the objects that will not be highlighted. To represent these elements, we included three 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 color we establish them as equivalent to the Color concept (See Section 4.4).

The benefit of this implementation is that the user is no longer responsible for the selection of the colors because 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 thinks looks nice, but does not represent 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 in order to improve the response time of 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 color

In this work, we expanded the work done in [3] 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 represents a primary color component. 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 neighbor. The opposite of a color \(c\) is another color \(d\), whereas \(d\) is facing \(c\) on the color wheel. The next to a color \(c\) is a color \(t\) which is the following one to \(c\) on the color wheel, also known as neighbor. The concept `color` can be easily extended by new data and object properties.

We also created the Colors Ontology, a new ontology that imports the previous one. The role of the Color Ontology is 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 The reasoning process

Having established the semantic elements in our architecture, we can now show how these elements are used by a reasoned 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 Ontology 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 Ontologies 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 so we ask the user to pick the visualization’s background color. Once this color is chosen, a color scheme is composed with the selected background color, its opposite and its neighbor, based on the Colors Ontology. There is no reasoned involved in this step.

With the color scheme and the semantic of the task, the reasoned can create a 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 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 and the reasoned knows that this is equivalent to the highlight concept in the Task Ontology.

Because of this, the color opposite to the selected one 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 Semantics.** The Spherical Layout technique supports different configurations of the final visualization. In our implementation of the layout there are multiples choices to graphically represent nodes, edges and visual aids:

- 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 one, follow by the cube; it is also possible not to map nodes visually. This gives us four possibilities of representation for the nodes: not mapped, points, cubes and spheres.
- Edges can be represented by two types of lines, a single line whose only visual property is color and a cylinder, which allows mapping more visual properties. The latest one is the most complex visual element. It is also possible not to map edges; For edges we have three possible representations: not mapped, lines and cylinders.
- In this implementation of the Spherical Layout the nodes are uniformly distributed on the spheres’ surface; to achieve this goal we discretized the surfaces of the spheres with triangles and place 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. The result from the reasoning process in this
stage is an instance of a concept called configuration; this concept indicates which visual elements to use in the visualization. This set of rules allows us to control how the visualization is created to make the most out of the current hardware.

**Visualization Creation.** The last intervention of the reasoned 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 [3], the reasoned can decide which color should be used for each data element, based on the filter task and the color map. Using the configuration concept the reasoner can determine 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 displayed. This concept includes data properties to handle the different choices for node and visual aid.

```
IF [hardware.GPU=True]
   THEN
      IF [data.size == "LARGE"]
         THEN
            Use Cubes for Nodes.
            Use Lines for Edges.
            Use Nothing for Visual Aids
            //the task is not considered in this case
         ELSE
            IF [hardware.monitor.size == "LARGE"]
               THEN
                  Use Spheres for Nodes.
                  Use Cylinders for Edges.
                  Use Nothing for Visual Aids.
                  IF [TASK=="FILTER"]
                     THEN
                        Use Triangles as Visual Aids only for the filtered elements.
                     END
               END
            ELSE
               IF [hardware.monitor.size == "REGULAR"]
                  THEN
                     Use Cubes for Nodes.
                     Use Lines for Edges.
                     ELSE //hardware.monitor.size == "SMALL"
                     Use Nothing for Nodes.
                     Use Lines for Edges.
                  END
                  IF [TASK=="FILTER"]
                     THEN
                        Use Triangles as Visual Aid only for the filtered elements.
                     END
               ELSE
                  Use Triangles as Visual Aid for all the elements.
               END
            END
         END
      ELSE
         Use Points for Nodes.
         Use Nothing for Lines.
         Use Nothing for Visual Aid.
         IF [TASK=="FILTER"]
            THEN
               Use Triangles as Visual Aid only for the filtered elements.
            END
      END
   END
ELSE //hardware.GPU == FALSE
Use Points for Nodes.
Use Nothing for Lines.
Use Nothing for Visual Aid.
IF [TASK=="FILTER"]
   THEN
      Use Triangles as Visual Aid only for the filtered elements.
   END
END
```
Fig. 2. Rules created to relate the semantics of the data, the hardware and the task with the configuration of the visualization technique.

4.6 Brows.AR Application

We developed Brows.AR, an application for the visualization of file system hierarchies in 3D based on the Spherical Layout ([4]). The Spherical Layout is a 3D generalization of the Radial Layout. Instead of circles, as in Radial Layout, we considered concentric spheres on whose surfaces we located 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 considered a spherical wedge and the nodes are allocated on the surfaces defined by this wedge.

In order to test our layout, we created a 3D representation of a file system 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 nodes but with a high level of transparency. The color of each node 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 the supported interactions see [4].

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 was ask. The Reasoner class used Protégé ([13]) and Jena ([14]) APIs to interact with the ontologies. The reasoning service was provided by the Pellet ([15]) 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 reasoned. If a particular data element is not in the cache, the application asks the reasoned for the corresponding displayElement instance; then the pair (data element, displayElement instance) is saved in the cache. Because all the edges are handled uniformly, the reasoned asks only once about this option at the beginning of the process.

5 Conclusions

We designed several ontologies related to the semantic of the visualization process. We included the semantic of the data, the tasks, the hardware and the color. Within the visualization process, we used a semantic reasoned to create the final
visualization. This architecture was integrated into the Brows.AR application, a 3D visualization of file systems. With this integration, we created a visualization system that was able to assist the user in the preparation and configuration of the visual representation.

A visualization system should ensure that, even if the user is not a visualization expert, the generated visualization will be the most suitable for the user and the data domain. This work presents a breakthrough in the visualization research because of the integration between the visualization process and the knowledge on visualization creation. The use of Brows.AR as a test case proved that it is possible to use a semantic reasoner to create a visual representation.

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

6 Acknowledgement

This research was partially supported by the PGI 24/N028 and 24/ZN19, Secretaría General the Ciencias y Tecnología, Universidad Nacional del Sur, Bahía Blanca, Argentina.

References


