Interaction on Graphs using non-conventional devices

Sebastián Escarza – Silvia M. Castro – Sergio R. Martig
VyGLab – Dpto. de Ciencias e Ingeniería de la Computación – Universidad Nacional del Sur
Av. Alem 1253 – 8000 Bahía Blanca – Buenos Aires – Argentina
{se,smc,srm}@cs.uns.edu.ar

ABSTRACT

Interactions become essential in Visualization because they enable users to adapt the visual representation to their own needs allowing them to explore the underlying information space. By using non-conventional devices, they face a wider fan of interaction possibilities to explore their data. But, there is still not an agreed base theory to describe the processes involved in the visualization pipeline data transformation and, from the interaction viewpoint, it does not exist an interaction classification that allows presenting them under a unified perspective.

The objective this work is to contribute to develop a Graph Visualization base classifying and defining the most representative interactions in the field. On this basis, it will be intended to lay the foundations to obtain a systematic development of interactive visualization tools using non-conventional devices. Then, an interaction over graphs' taxonomy is proposed, all their categories are placed in the reference model context and the most relevant aspects to design visualizations using this kind of devices, are analyzed.

Keywords: Interactions, Graph Visualization, Information Visualization, HCI, Devices.

1. INTRODUCTION

A few years ago, small volumes of information had been handled in different application domains and their direct analysis and processing was feasible. Nevertheless, while more and more information is generated, it is mandatory the use of techniques that allow an adequate data processing. The direct analysis of large data sets is not possible and data views should be provided in order to highlight aspects of users' interest.

In this context, Visualization provides useful choices to obtain a more effective data set exploration. With a visual data representation, the task of identifying patterns and extracting conclusions about them becomes easier. However, this representation should be adapted to the user needs and, consequently, it cannot be a static view. Therefore, interactions must be provided. The generation of a static data image is a limited approach compared with a visualization tool which provides to users the appropriate tools to navigate the data and interact with they representation.

Graph Visualization is an important Information Visualization field. In this field, the data sets can be modeled by a graph that is afterwards visualized. The graph's nodes represent data and the graph's edges represent relationships between this data.

This work proposes a classification of interactions and it embeds each class in the frame provided by the Unified Visualization Model (UVM) [6], a reference model for Visualization tools. Finally, a concrete application, which allows it to interact by using non-conventional devices, is analyzed.

2. INTERACTION IN GRAPH VISUALIZATION

The cognitive process of explore a data set starting from a simplified view and, then, introducing step by step into the representation details, is not possible without the support for needed interactions.

In Graph Visualization, applications share an agreed set of interactions. In the broader field of Information Visualization, by contrast, the identification of such set is difficult because the great diversity of application domains.

Some interactions in Graph Visualization go beyond this area and appear in other domains of Information Visualization. This kind of interactions is more concerned to the metaphor that is presented to user than the formalism of representation. The positioning of a camera in the representation space, for example, is an interaction provided in Graph Visualization and it is also present in other domains. More generally, every interaction that change the users viewpoint, their interest focus or the visualized detail, work independently of whether the data representation is a graph or not.

Other interactions are more involved with the formalism of representation, like the graph in this case. It is common that this kind of interactions be defined in terms of the graph's elements: nodes, edges, labels, and so on. As examples of these interactions, it can be mentioned the selection of a node, an edge or a path. Although the selection of visual elements is present in applications beyond Graph Visualization and, therefore, it can be considered a more general interaction; the elements that are selected depends on the application domain. In graph visualization, for example, user selects nodes or edges instead of simply visual elements.

The previously mentioned set of interactions is present in most of Graph Visualization tools. However, there is not a methodology that clearly defines the visualization building process. Actually, we only have different tools that have been built with independent approaches that not share a general framework and that have not been set in a common context.

The existence of a reference model to give us such a framework, is important to users from several domains attain effective interactions based in a single mental model. This reference model is also important to visualization designers in order to express clearly how to transform the data to get the view, what operations should be provided and what interaction styles should be between the user and the visualization tool.

3. CLASSIFICATION OF INTERACTIONS OVER GRAPHS

Interactions, in the UVM [6] context, can be defined in terms of the states that they affect and the transformations that must be re-executed to adjust the visualization to meet the new user's requirements. In this sense, the UVM provides a theoretic frame that allows us to put interactions in context.

Orthogonally to this reference model, a taxonomy of interactions is needed to simplify their study and highlight their similarities and differences. Thus, it is possible to identify and apply common criterions, by grouping interactions in classes. This uniformity could be the starting point to define design methodologies for visualizations, which treat interactions systematically, instead of the ad-hoc approaches that exist nowadays.

We propose a classification of interactions on graphs. Each category, in addition to share common characteristics among its members, receives a uniform treatment that is given from the viewpoint of the UVM. Also, for each one, it is defined the UVM pipeline stage where every member must be processed. When the interaction involves more than one stage in the model, it is considered, for description purposes, that it affects the earliest of these stages.

3.1 Exploration Interactions

This set of interactions allows users to explore their data and assist them in the cognitive process that occurs when they make use of the visualization. Under this category, we have all the interactions, which provide detail on demand. The modification in this level of detail can be accomplished with a change in the data that is visualized or in its representation.

The ***semantic zoom*** allows the change of the detail of the elements that are present in the view. For example, it is possible to get more associated information to a node or edge by dragging the cursor over them or to modify the amount of associated information to a node that is showed. A change in the data representation can occur whenever the detail of information is modified. In this sense, the ***clustering*** is a resource that allows the visual complexity reduction by building groups, called clusters, from several entities. The interactions related to the ***clustering*** can be thought as a kind of ***semantic zoom*** because they allow greater or lesser detail. Expand and collapse nodes can be considered as interactions of this kind.

The ***filtering*** establishes which entities will be visualized. By contrast with semantic zoom, which modify the detail of each entity individually, it allows to change the level of detail of the view globally, in the sense that it adds or deletes visual elements from the given view.

The ***metrics based filtering*** is a filtering process driven by metric values associated to each entity. It is usual, for example, to visualize only the nodes that are nearer than some distance with respect to a certain reference node. It is also possible to filter using the semantic of data. For example a ***filtering*** could be performed for only visualize branches that belong to a city in a domain in which the graph represents a network of bank's branches.

Every Exploration Interaction is solved in the *Abstract Data to Data to be Visualized Transformation* of the UVM. When a change in the detail of the showed information occurs or when a filter is applied, the displayed data is modified and it becomes mandatory to select a new *Data to be Visualized* set to display, which occurs in the transformation mentioned. For example, if the user wants to get more information attached to some node (i.e. a semantic zoom case), the node representation is replaced with by another one that contains the desired data. If the user perform some metric based filtering over the nodes of a visualized graph, it becomes necessary to substitute the set of nodes currently displayed by the node set whose metric values satisfy the filtering criterion.

3.2. Selection Interactions

This kind of interactions allows the distinction of a subset of elements that are present in the view to accomplish some task over them. After a Selection Interaction, interactions corresponding to other categories of this taxonomy are usually applied.

The Selection Interactions involve two main aspects: first the user needs visual feedback to know which elements has been selected, and second, the selected elements must be marked in the underlying data set to maintain the selection itself for future interactions.

In Graph Visualization it is possible to select a node, an edge, a path or more generally, sub-graphs or subsets of nodes or edges. The selection, as interaction, can trigger other interactions due to user intervention or can start the execution of actions without his/her intervention. In the first case, for example, if the user selects a set of nodes, then he/she can change some visual attributes like color or size. In the second one, there are tools which, after a node selection occurs, and without user intervention, adjusts itself to keep the user's focus of interest.

The Selection Interactions processing involves re-executing the visualization pipeline from the set of *Data to be Visualized*. Accordingly to the UVM definition, all the data present in this set must be represented in the view. The UVM supports more than one view of the same *Data to be Visualized* set and it is necessary to ensure their consistence. Therefore, it is necessary to mark the selected elements in this data set. In order to provide the needed visual feedback to user, the *Visual Mapping Transformation* must be redefined to make visible the selection.

3.3. Representation Interactions

All the data present in a view is represented through visual elements with certain graphical properties. These elements are laid out in the space defining the spatial substrate. The layout of visual elements is a critical aspect of the visual mapping because it is usually defined as a function of a subset of the underlying data attributes. The visual element's graphical properties can be defined arbitrarily or also can be used to map some other data attributes.

The Representation Interactions allow users to change the spatial substrate definition and the visual representation of the data. A change in the visual representation can involve modifying the visual elements or their graphical properties. Using this kind of interactions, users can re-position nodes, select the visual elements that will represent nodes or edges, modify element's graphical properties such as color, size, and so on. Also, they can modify the way in which the labels are represented by defining textual properties like the font type, font size, style, etc. Setting the relative position of a label with respect to a node or edge it is also an interaction of this kind.

Representation Interactions are solved in the *Visual Mapping Transformation* of the UVM. This transformation adds information to the data about how it will be visually represented. Any modification in the way of representation makes necessary to re-apply this transformation in order to adjust the visual mapping to the user needs. For example, if color is being used to represent some attribute of the nodes and the user wants to modify the used color map, it is necessary to change the transfer function which is applied in the *Visual Mapping Transformation*.

3.4. Navigation Interactions

Interactions in this category allow user to explore the representation space by modifying what the user sees. By contrast, Exploration Interactions makes it possible to explore the underlying data.

In every visualization tool there is an area in the view, which is the user's focus of interest and which is observed from some viewpoint. Navigation Interactions allow it to modify the interest focus and the viewpoint of the user. These modifications can be performed freely or driven by constrains.

In driven navigation, for example, it is possible to fix the camera target in some node (the user's focus) by constraining all camera transformations in order to keep the focus always visible. Another way of driven navigation consists on to avoid camera displacements out of the graph structure by constraining the camera translations.

Navigation Interactions are processed in the *Visualized Data* state of the UVM. This kind of interactions does not require re-executing previous stages. Once the *Visualization Transformation* is applied, the view is built. After that, the last step in the visualization pipeline is executed. This is the graphic processing of the representation obtained (the result of the graphics pipeline execution) and it is accomplished in the *Visualized Data* state. Navigation Interactions affects this graphic process but not the visual representation itself. Consequently, interactions like ***geometric zoom*** and ***camera rotations or translations*** are solved in the *Visualized Data* stage.

4. INTERACTIONS AND DEVICES

Interaction is defined as a reciprocal action, effect, or influence. In Information Visualization context, it is related to the process in which user and visualization mutually affect. Users affect visualizations by making use of instructions that modify some of their parameters and visualizations affects users by providing them another data perspective as a reply to their actions.

In this forth and back process, two distinct components can be identified: input devices (by which user affects visualization) and output devices (by which user get the desired data view). The most common case is to have a conventional display as output device and a keyboard plus a mouse as input ones. This configuration limits the human-computer interaction because constrain users to the bidimensional desktop space and to a single pointer to affect the view.

Many devices are developing and improving every day to avoid these limitations. Touchscreens, holographic displays and virtual reality glasses are only few examples. From the interaction point of view, using a 3D pointer should allow a more natural interaction with tridimensional representations and having several pointers on the same environment, should allow to multiple users to collaborate in the handling of the data representation, resulting in a multi-user visualization.

In this context, an important distinction must be made between the visual representation of the data and its final representation in a particular output device and also between the interaction provided by the visualization and the action performed over some input device. Let’s consider, for example, a 3D-graph visualization application. It initially builds a 3D-graph representation from which user can begin to interact. In a conventional display, the execution of the graphic pipeline will project the 3D-graph geometry into a 2D space and then it will rasterize the resulting fragments in order to obtain a pixel composite image in the display. However, if there were a holographic 3D display the mapping of the 3D-representation would be different because such a device can directly represent the generated 3D space.

On the input side, it occurs something similar. Let’s consider the same case of 3D-graph visualization. Let’s suppose that this visualization defines *a node selection interaction*. User moves the mouse cursor onto the desired node and then clicks over it. This action starts a process which transforms the 2D screen coordinate of the mouse click into coordinates in the 3D representation space (inverse projection) and, finally, with the obtained 3D position, the node is selected. If a 3D input device were used, the 2D-3D transformation would not be required.

Clearly, due to the great diversity of devices, it is convenient to make a distinction between the building of the view and the process used to represent this view in every output device; and between the interaction provided by the visualization and its implementation in every input device.

5. NON-CONVENTIONAL DEVICES: THE P5 GLOVE

With the objective of make an initial approximation to the interactions using non-conventional devices in mind, it was implemented a graph visualization tool prototype that allows user to interact through the P5 virtual reality glove. From this initial approximation, it has been possible to obtain results, many aspects of the problem has been recognized and several conclusions has been extracted which are detailed along the next paragraphs. This non-conventional device was chosen entirely due to its availability.

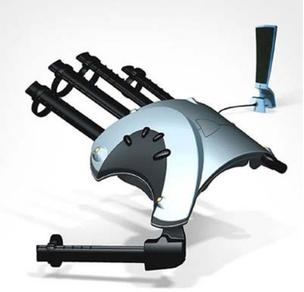

Figure 1: The P5 Glove

The P5 glove is composed of an infrared sensor, which is placed in the desktop and a glove which is worn by user. The glove is wired to the sensor connected to the system by an USB interface. The glove allows user to capture the position, pitch, yaw and roll of his/her hand in 3D space and, for each finger, to get a quantification of its bend. Additionally, the glove has four programmable buttons to assign functions. The positional and rotational data is acquired by the infrared sensing capability and the fingers’ bend and buttons’ state are acquired via the wire (Figure 1).

Against a traditional mouse, in which a 2D space (the desktop) is mapped into another one (the screen), with the P5 glove it is possible to map a 3D space into another 3D one. By making use of the third coordinate and acquiring the hand rotation and the fingers’ bend, many new possibilities can be exploited.

In order to give an idea to users about where they are positioned, it is needed to provide a 3D cursor. If the input device is the P5 glove but the output device is still a conventional display (as in our prototype case), visual cues become necessary to help user to solve the ambiguity that appears when a 3D space is mapped into a 2D one. In our prototype, these visual cues consist of a cube-shaped cursor. Its size helps to solve the depth ambiguity. Also, the prototype provides an orthogonal axis system that is moved with the 3D cursor. This system helps user to solve the position of the cursor inside the representation space. Finally, and with the objective to make the node selection easier, the nearest graph node to the cursor is highlighted. In this way, the user is given an idea of what node should be selected if a selection were performed (Figure 2).

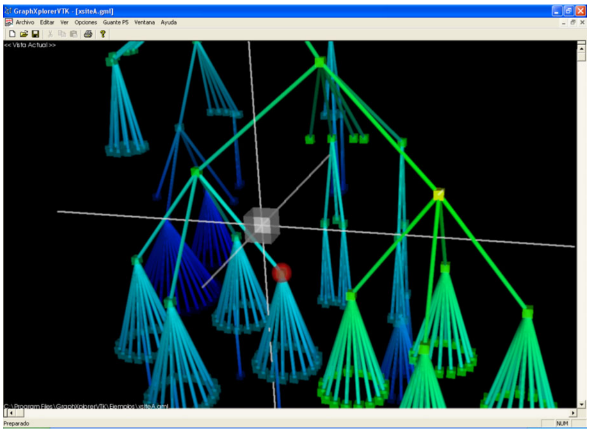
GraphXplorerVTK - [xsiteA.gml]
Archivo Editar Ver Opciones Guante P5 Ventana Ayuda
<< Vista Actual >>
C:\Program Files\GraphXplorerVTK\Ejemplos\xsiteA.gml
Preparado
NUM

Figure 2: Visual cues for 3D interaction: In the application capture can be appreciated the cubic-shaped cursor, the orthogonal axis system and the sphere that highlights the nearest node to the cursor.

However, the P5 glove has limitations from the device point of view. Users have to keep their hand in the sensing area. Therefore, they have to avoid that an object (including their own hands) interposes between the glove and the sensor. This conditions user’s behavior and makes interaction less natural. Additionally, after a long time of use, the glove becomes uncomfortable. Keeping the arm in the air during long periods of time, originates tiredness and the interactivity is affected. Also, due to that only a measure of the fingers' bends is acquired, it is impossible neither to capture specific gestures such as the gesture of taking a tiny object with the thumb and index finger ends nor to acquire the separation between fingers.

Nevertheless, although its limitations, this device provides additional possibilities that should be explored in order to be used as visualization input device.

6. MAPPING GESTURES INTO INTERACTIONS

In previous sections, it was emphasized the convenience to separate the representation space generated by the visualization from its representation in a particular output device, as well as the convenience to do the same separation between the interaction provided by the visualization and its implementation in certain input device. Therefore, mappings must be defined between the generated views and their representations for output devices, and between the interactions provided and their implementations for input devices.

In this section we describe a mapping between some interactions provided by our graph visualization prototype and their implementations for the P5 glove. We call *gestures* to these implementations of interactions in devices. The adopted approach to define the mapping between gestures and interactions has as main objective to attain a natural user-visualization interaction. However, because it is a relatively unexplored field, there are not methodologies that guide the design process.

The relationship among gestures and interactions must be a function. For each gesture, it must be one single associated interaction. However, it is possible that many gestures triggers the same interaction or that there were interactions without associated gestures and conversely.

In some devices, the uniqueness of an interaction associated to a gesture is implemented directly. For example, a right-button drag, in a conventional mouse, it is clearly distinguished from a left-button drag. Nevertheless, with a device like the P5 glove, which provides more possibilities, the conditions that define a gesture must be explicitly established to ensure the mutual exclusion between them. In particular, in the P5 glove case, for each gesture it must be defined the position and rotation of the glove with respect to the sensor, the bend of each finger and the state of each button.

Another aspect to get into consideration with non-conventional devices is the association between the parameter acquired from the device and its corresponding interaction parameter. For example, let's consider an interaction that allows rotating the camera by an angle. This angle will be modified as a function of some device parameters. With a mouse, the value of one coordinate it is often mapped to the angle value. But in another kind of devices such as the P5 glove, it must be decided what to map to this angle value; there exists different possibilities like a hand's position coordinate, a hand's rotation angle or a finger's bend value.

In general, each input device has a set of parameters, which are transmitted to the application. Given a gesture, a subset of them must be used to univocally establish which interaction will be triggered in reply to the gesture, another subset must be mapped into the interaction parameters to control the interaction and a third subset could be ignored for certain interactions.

With all these considerations in mind, we can define the interaction support and the associated gestures provided by our prototype. The selected interaction set for our prototype, represent a basic subset of common interactions in Graph Visualization. Navigation and Selection Interactions has support for the P5 glove. Using the P5 glove it is possible to place the camera in the 3D-representation space and to select the node, which represents the user's focus of interest. The prototype provides free and driven navigation and in the latter case the camera target is constrained to the current node (the most recently selected node, i.e. the user's focus of interest).

With respect to gesture design, it is important to notice that every interactive system must define a *neutral gesture*, i.e. at least one gesture must not be associated to any interaction. When the user is idle, a way of "no interaction" must be provided in order to he/she can appreciate the generated view without modifications. In the case of our prototype, this is achieved extending all fingers (Figure 3a). During the *neutral gesture* the visualization only moves the cursor over the representation, as it happens when no button is pressed with a mouse.

By keeping the thumb, the index and middle fingers extended and flexing the ring and little ones, it is possible to re-position the glove in front of the sensor without modify the cursor's position, emulating the effect of to pick up the mouse from the desktop (Figure 3c). This gesture, called *re-position gesture*, is pre-implemented in the device driver and it becomes useful to back to the sensing area when we accidentally move out of it.

The **selection interactions** are limited to a single node selection. The selected node is called the current node. The associated gesture implies to do the click movement with the index finger (the remaining fingers must be extended) analogously to the mouse click but without buttons to be pressed (Figure 3b). Once the gesture is performed, the nearest node to the 3D-cursor position becomes the current node.

With respect to **navigation interactions**, it is possible, by using the *grab gesture*, to directly manipulate the graph. It is possible to position it (only in free navigation) and to rotate it arbitrarily (Figures 3d and 3e respectively). To do this gesture, user must close his/her hand and virtually "grab" the graph. Then, it is possible to translate it and rotate it as well as user translate or rotate his or her hand. In driven navigation, the graph is anchored by the current node, which is kept centered in the view, and only it is possible to perform rotations.

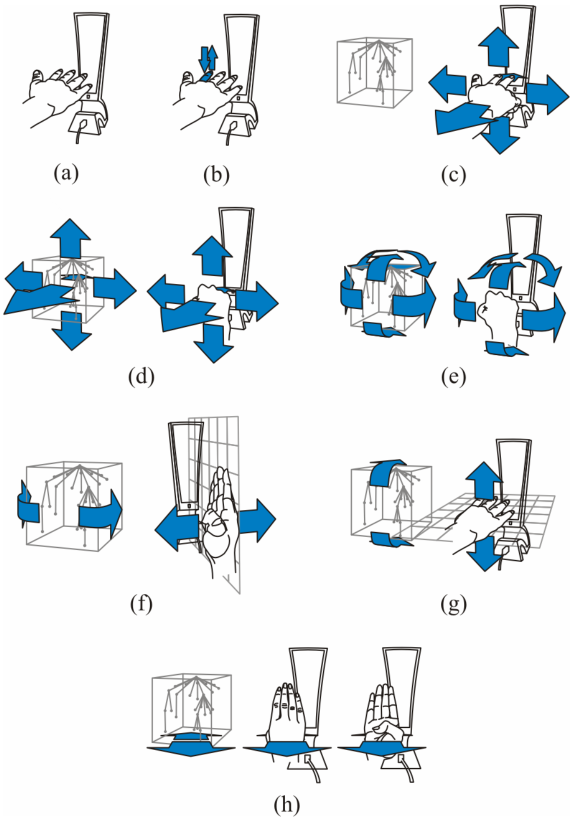

Figure 3: Gesture design for our prototype: a) Neutral Gesture. b) Node Selection Gesture. c) Re-position Gesture. d) Grab Gesture: Translation. e) Grab Gesture: Rotation. f) Azimuth Gesture. g) Elevation Gesture. h) Dolly Gesture.

Additionally, it is possible to use some gestures to modify the camera's *azimuth*, *elevation* and *dolly* values. In these gestures, user must bend the thumb and keep extended the other fingers, and the plane formed bye the palm must be positioned orthogonally to the sensor axis system. The hand's movement in direction of the palm or its back controls the modification of the azimuth, elevation or dolly values (Figures 3f, 3g and 3h respectively).

The effectiveness of the interactions using non-conventional devices relies, to a large extent, on the *naturalness* of the associated gestures. With intuitive gestures that map their parameters into the interactions ones in an expected way, it is possible to improve the usability and to make easier to users their interaction with the visualization. In this sense, to attain *naturalness* in the gesture design, the interaction with the 3D visual representation should be designed taking into account the user interaction with the equivalent real world tangible object. By taking real elements, which are known by the users, it is feasible to build very effective metaphors that exploit their knowledge of the world to reach a far more natural interaction.

Apart from *naturalness*, it is necessary semantic *uniqueness* for each gesture. In order to avoid confusions, the gesture design must reach mutual exclusion between them. In this sense, the parameter values that define certain gesture should be selected in a way that does not overlap with other ones. Often, to reach a less rigid interaction, tolerances are incorporated to these mentioned values. This makes more difficult the task to attain this *uniqueness*.

A third aspect to be considered is the *precision* in the interaction. Although an interaction's associated gesture can be very natural; it lacks of utility if cannot be performed with *precision*. In this sense, it should be noticed that humans are better prepared to do precise movements than to measure time intervals with precision. Therefore, to reach some interaction with the representation, it is more precise to map a movement of the device than to map the time interval that a button is kept pressed. Accordingly to this principle and taking advantage of this perceptual characteristic we decide to use, for our gesture design, the movement of the P5 glove to move the cursor and to rotate and translate the representation.

7. CONCLUSIONS AND FUTURE WORK

In this work, the most representative interactions from the Graph Visualization area have been classified. Besides that interaction taxonomy has been proposed. This taxonomy was expressed very naturally in terms of the UVM. Putting interactions in context of the mentioned model allows expressing their impact in the visualization process, the affected states and the transformations that must be re-executed. This framework provides a common vocabulary to meet the user needs with the visualization's designer's tasks.

In this sense, it is expected that many interaction taxonomies coming from several fields of Information Visualization, can be unified, to build a general taxonomy for interactions in visualization. This work contributes to this objective with a taxonomy for interaction over graphs.

With respect to the use of non-conventional devices, a graph visualization tool prototype that has support for one particular device has been implemented. As consequence of this experience, it was possible to begin to identify the main aspects of the problem and some design decisions that contribute to improve the tool quality. For example, the division between the interaction and the gesture allows it to attack the problem in a more modular way. The designer can work on some interaction independently of the input-output devices and he can implement some interaction in certain device while he abstracts himself from the rest.

Some aspects only become visible after the development of such a tool. By unifying and generalizing these aspects we hope to start to draft a base theory. The experimentation with the use of our tool allows us to conclude that the use of non-conventional devices can provide greater naturalness to interaction. With the introduction of nearer-to-user metaphors, the visualization tool become easier to use and the slope of its learning curve is decremented. However, although the interaction design with non-conventional devices offers a new horizon of possibilities, it presents challenges. Apart from naturalness, semantic uniqueness and precision in interactions are also central points in this kind of developments.

By having a concrete implementation allows us to evaluate the tool from the usability viewpoint and to measure the impact on HCI. In this sense, it is planned to make usability tests to quantify the improvement that results from use the glove in Graph Visualization against the traditional keyboard and mouse configuration. For reach these objectives, several kinds of graphs and many typical tasks of the area will be taken in consideration. Another observation is the fact that most of potential users of such a system have a greater degree of experience with the keyboard and the mouse. This fact should be considered during the usability test's result's analysis.

The lack of design guides, that orient the gestures development and allow defining the most adequate associations with interactions, the great diversity of devices and the different needs of each application domain establish an ad-hoc and heterogeneous context. However, these isolated attempts will allow synthesizing the important aspects with the final objective to unify the common aspects and to lay the foundations in order to define methodologies that dictate the steps for future systematic developments, advancing the state of the art.

7. ACKNOWLEDGMENTS

This research is partially supported by the PGI 24/N015 y 24/Zn12, Secretaría General de Ciencia y Tecnología, Universidad Nacional del Sur, Bahía Blanca, Argentina and the PICT 2003 Nº 15043, ANPCyT, SeCyT, Argentina.

8. REFERENCES

[1] Card S., Mackinlay J. and Shneiderman B., editors. "*Readings in information visualization: using vision to think"*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[2] Curino R., "*Interacciones en el contexto del Modelo Unificado de Visualización"*, Tesis de Grado para la obtención del título de Licenciada en Ciencias de la Computación. Dpto. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, 2005.

[3] Di Battista G., Eades P., Tamassia R., and Tollis I.G., *"Graph Drawing: Algorithms for the Visualization of Graphs"*, Prentice Hall, 1999.

[4] Escarza S., "*Interacciones mediante dispositivos no convencionales en Visualización de Grafos"*, Proyecto Final de Grado para la obtención del título de Ingeniero en Sistemas de Computación. Dpto. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, 2005.

[5] Herman I., Melançon G., Marshall S., "*Graph Visualization and Navigation in Information Visualization: a Survey"*, IEEE Transactions on Visualization and Computer Graphics, Vol. 6, No. 1, pp. 24-43, 2000.

[6] Martig S., Castro S., Fillottrani P., Estévez E., "*Un Modelo Unificado de Visualización"*, Proceedings, pp. 881-892, 9º Congreso Argentino de Ciencias de la Computación. 6 al 10 de Octubre de 2003. La Plata. Argentina.

[7] Shneiderman B. and Plaisant C. "Designing the User Interface: Strategies for Effective Human-Computer Interaction" (4th Edition). Pearson Addison Wesley, 2004.