Visualisations; Functionality and Interaction

Claire Knight and Malcolm Munro

Visualisation Research Group, Research Institute in Software Evolution,
University of Durham, Durham, DH1 3LE, UK.
Tel: +44 191 374 2554, Fax: +44 191 374 2560
{C.R.Knight, Malcolm.Munro}@durham.ac.uk
<http://vrg.dur.ac.uk/>

Abstract. The interface to any visualisation application can essentially be split into two; the visualisation itself, and the interface that is associated with the interface. Despite the fact that the data plays a major part in any visualisation application, this is generally accessed only through either the interface or the visualisation. An important issue then arises over the location of functionality in the application. To be able to provide a usable and effective visualisation application (including data integration) where are the various selections, filters, customisations, and interrogation mechanisms located; the visualisation or the interface? The interaction mechanisms involved play a part in guiding this decision, but the amount of coupling between the visualisation and interface is also important.

1 Introduction

Visualisations are very direct interfaces to the data that forms their basis. Because of the variety of display and representation, and the problems of multiply overloading functionality onto glyphs there is usually a requirement for a separate, more traditional, interface that works in conjunction with the visualisation graphics. This can be extremely beneficial from a usability perspective; it provides an easier, more familiar, route into using any visualisation, and there can even be replication of functionality to allow for different users. The problem comes when designing such visualisations and their interfaces; where is the best place for interaction and functionality?

The likelihood is that there will be a split of functionality between the interface and the visualisation, and that in order to facilitate this functionality in the visualisation suitable interaction will be required. There is also the issue of which functionality to place where. Whilst some of this may be dependent on the data, on the tasks being supported, and even the user base of the visualisation it may be that general guidelines can be developed.

This paper presents some initial guidelines that have stemmed from the process of working with three-dimensional system and software visualisations. These have required the use of standard interface techniques as well as the various procedures that are concerned with visualisations. Three dimensional interfaces may suffer more from the perceptual issues associated with interfaces because of the expectation that

much human-computer interaction occurs within the desktop metaphor, and the more obvious separation of being in a virtual space compared to clicking on windows. Desktop visualisation and virtual reality may lose the two-dimensionality aspect through clever use of graphics but there is still the screen restriction and the normal user positioning and devices to use the systems. This means that it is necessary to have controls and input areas somewhere thus the need for the issues presented in this paper to be considered.

2 Concatenation or Combination?

An important subject when discussing visualisations, interfaces, and for want of a better term, visualisation applications, is where the division actually lies. For the user of a visualisation application (visualisation and the interface) this distinction should be trivial and relatively transparent. Indeed it should not really concern them at all. However, it is vital that this is considered by those who design and create visualisation applications as this will have a direct impact on the visualisation and the visualisation application usability [2].

There is obviously some degree of coupling between the interface and the visualisation because they work together to create the visualisation application. There is also the need to tailor the interface options to work with the data (as any visualisation has to) and to allow change of those aspects considered to be parameters to the visualisation. From a more established aspect, there is the expectation of what various parts of the interface will do. This has a direct impact on the visualisation if that interface links to the visualisation. Slider bars are popular ways of controlling various aspects of visualisations because of their familiarity and ease of use. A novel use of these would likely cause interface and visualisation usability problems for this very reason. In this case the control would be better integrated in another aspect of the application.

Essentially the uncertainty lies with whether a complete visualisation application consists of:

visualisation + interface [Concatenation]

Or whether it can be seen as:

visualisation □ interface [Combination]

This is a subtle difference, but the important distinction is whether there is a bolting together of the two components, or whether they are actually combined together with a higher degree of coupling and awareness than a standard API would provide.

The concatenation of two components can create an application, but as to whether it is a good one because of the lack of awareness is an area that would require more investigation. Combination provides a more coupled interface and visualisation and thus defeats some of the principles of reuse but may create a better overall application. Also, with component technologies much improved, there is no reason why major parts of the interface (for example) cannot be sub-components that can then be reused with some form of overall data and visualisation specific co-ordination and parameterisation.

3 Visualisation Interaction

Visualisations have more use if they are interactive [1] as they then allow users to be able to work with, investigate, browse, interrogate, and generally view the various aspects of the data without relying on predefined fixed views. Whilst it is necessary to provide one or more representations and then the views that are valid for this representation in combination with the data (essentially the visualisation) there is still much user freedom if interaction is possible.

This then solicits the question of what interaction is allowed for a given visualisation. This is in two main areas; (a) interrogation style interaction with the graphics and hence the data, and (b) navigational interaction within the visualisation. With current technologies there is the ability to be able to incorporate various interaction mechanisms, but there is need to evaluate which are necessary for usability and operation rather than because it is technically feasible.

The first difference that any visualisation is likely to have over a standard graphical user interface is the number of degrees of freedom for movement and thus interaction. Whilst a normal interface may involve several discrete windows, or some amount of scrolling to incorporate all of the necessary data, this is the most that a user has to deal with. The contents of the windows (such as in a word processor) are not covered by this definition. In the same way that a two dimensional visualisation is a possibly infinite canvas contained within a window, any data interface is the same. There is the ability to scroll (pan) around this space, to zoom, to (if provided) utilise overviews such as table of contents or high-level images of the complete visualisation, and to be able to interact with the data contained within this creative space. Three-dimensional visualisations provide not only an extra dimension to consider, but also the rotation around these three axes; roll, pitch, and yaw. There is not this issue with two-dimensional visualisations, because whilst there might be the ability to rotate the visualisation around one of the axis there is generally the requirement that the other axis is fixed in order the display is still visible to the user. Therefore three-dimensional visualisations actually add another four degrees of possible movement. The navigation around the space should therefore take this into account. For two-dimensional displays standard mouse and keyboard interaction should not pose any problems. On the other hand, three dimensions provides an added challenge to the process of orientation and movement. This implies that the use of navigational aids is necessary in order to prevent a feeling of being lost [5, 7]. Pettifer and West [10] also relate the problem to the systems and metaphors in use today.

“Losing a cursor on the desktop is one thing, losing yourself in cyberspace is quite another.”

Three-dimensional worlds are potentially infinite whereas desktops are of generally finite space even if current implementations are able to cover several screens.

The interaction with graphical objects is also based around a familiar interaction method. Generally most three-dimensional visualisations are integrated into desktop applications therefore the use of gloves, and virtual spaces such as a CAVE [8, 9], are not the largest concern. Such interfaces would require more consideration than this paper is able to provide, and also would likely be able to learn much from the Virtual Reality community over interaction and usability. The usual interaction with

graphical objects (regardless of dimension) is through selection, usually with a mouse click, and then the selection of the required detail or data refinement. These can be chosen through a variety of mechanisms; interface interaction with the selected graphic, the use of drop-down menus, right or multiple clicking.

The various ways of integrating with the visualisation for navigation and data interrogation have been summarised, but the issue for anyone who designs visualisations is to make justified decisions as to which are integrated into a visualisation application. An instinctive answer would be that incorporating all variations of interaction would provide the most usable interface because it would cater with the different demands of different users. This may not actually be the case because there would reach a point where the interaction styles would have to be overloaded on order to provide all of the necessary interaction. In other words, the same actions would have different functions depending on the context. Whilst data context may provide this (such as what detail is available from a specific type of graphical object in the visualisation) to do this with pervasive operations like mouse clicks is likely to cause problems for users. There is the associated issue that this then makes the interface more confusing and it then has a steeper learning curve. If a visualisation is too complex, compared to even rudimentary other analysis or understanding techniques, then it is not going to be used. This means that the related benefits that visualisation can provide of augmentation and amplification will be lost because of unrelated issues relating to the interface.

4 Functionality Location

Regardless of the permitted interactions with the interface and the visualisation of the visualisation application there are decisions as to be made as to where it is best to locate the functionality. Without any functionality, it is irrelevant what interaction and interface niceties are present as they will never have any impact on the data being visualised. Because interaction may be duplicated between the visualisation and the interface then it means that the functionality that is located behind that interaction needs to be consistent. Without some degree of detailed coupling between interface and visualisation this could not be guaranteed. They would both need to utilise the same routines (at an implementation level) therefore encapsulating this and making it directly accessible to both visualisation and interface, with adequate linking to enable the impact of that interaction (such as filtering the data) to actually affect the visualisation and interface.

Functionality can also be only available from one of the two main parts of a visualisation application. In these cases the question becomes not one of implementation and ensuring consistency and accuracy, but of being able to locate the functionality in the best place for its use (usability) and most effectiveness based on the actions it actually encompasses. There are four possible choices:

1. *From the visualisation only*
2. *From the interface only*
3. *From a combination of the visualisation and interface (requiring both)*

4. *From both the visualisation and the interface (duplication)*

Since the effectiveness of a visualisation (application) has dependencies on both the underlying data and how well the representations reflect this, and on the tasks that it is intended to support [4], then this may dictate which of these options applies to each piece of functionality. It may also be there is user demand for the inclusion of functionality in a particular place. Since the interface is generally included to augment the visualisation, then there will probably be little functionality that is restricted to the interface, but the same cannot be said of the visualisation. As a rule of thumb to guide the decision making process the following applies:

1. *If the functionality has a local impact (such as interrogating a graphical object) then it should be attached to the visualisation. Duplication through interface provision is also feasible.*
2. *If the functionality has a global impact (such as over all of the visualisation) then it is best located on the interface but combined with the visualisation.*
3. *If the functionality is related to the application (such as preferences) then it should only be attached to the interface.*

These guidelines make the distinction between global and local. It would make little sense if one wanted to change all glyphs representation a particular type of data to have to do each one individually. Likewise, selecting one object in order to obtain more detail about it does not make sense if applied to the global scale of a visualisation. There is some crossover, such as when the detail of an object then dictates that this should then become the focus of a new query (for example). In cases such as this a local impact would move to having a global one and therefore the integrated functionality guideline above would come into play. Despite all of this, these are only guidelines and doubtless certain data sets will dictate that they cannot be followed directly.

5 Conclusions

This paper has examined the interaction between the visualisation and interface components of a visualisation application. This is important to be able to consider where is the best place to locate functionality in the application. This has impacts on usability and therefore use of the application, and also for implementation and design of visualisations.

Related work by the authors [3] involves the development of a component-oriented framework for visualisations and data sources. This allows for multiple data sources to be integrated with multiple visualisations. The development of the framework so far dictates that the various visualisation views that are applicable and their control panel (the interface in the context of this paper) are a complete component unit. This allows for the required level of coupling considered necessary to fulfill the various guidelines expressed in this paper.

This framework development provides a perfect opportunity for developing further these ideas, and for then investigation aspects such as linking between visualisation styles and views (that are not already linked), the impact that cross data and cross visualisation annotation has on the location of functionality, and the issues associated with dealing with multiple data sources and multiple visualisations. This paper provides much further work, and also seeks to try and add science to the creative process that currently surrounds visualisations [6]. There is always going to be a create element to any visualisation because of the graphic design involved, but there is not reason why solutions cannot be engineered through the other aspects of designing and creating visualisations.

Acknowledgements

This work has been supported by the EPSRC project VVSRE; Visualising Software in Virtual Reality Environments.

References

1. Chalmers, M., Design Perspectives in Visualising Complex Information, Proceedings IFIP 3rd Visual Databases Conference, March 1995.
2. Neilson, J. (1993). Usability Engineering, Academic Press Professional Publishing, San Francisco, 1993.
3. Knight, C., and Munro, M., m-by-n Visualisations, Submitted to IEEE Visualization, Information Visualization Symposium 2001, University of Durham, Department of Computer Science Technical Report...
4. Knight, C., Visualisation Effectiveness, submitted to Workshop on Fundamental Issues of Visualisation, Proceedings of CISST, 2001, University of Durham, Department of Computer Science Technical Report...
5. Ingram, R. and Benford, S., Legibility Enhancement for Information Visualisation, Proceedings of IEEE Visualization '95, October 30 - November 3 1995.
6. Card, S., Mackinlay, J., and Shneiderman, B., (Editors), Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann, February 1999.
7. Dieberger, A., Providing spatial navigation for the World Wide Web, Spatial Information Theory, Proceedings of Cosit '95, pp93-106, September 1995.
8. Leigh, J., and Johnson, A. E., Supporting Transcontinental Collaborative Work in Persistent Virtual Environments, IEEE Computer Graphics & Applications, Vol. 16, No. 4, pp47-51, July 1996.
9. Leigh, J., Rajlich, P. J., Stein, R. J., Johnson, A. E., and DeFanti, T. A., LIMBO/VTK: A Tool for Rapid Tele-Immersive Visualization, Proceedings of IEEE Visualization '98, October 18-23, 1998.
10. Pettifer, S., and West, A., Deva: A coherent operating environment for large scale VR applications, Presented at the first Virtual Reality Universe conference, April 1997.