VISUALISATION EFFECTIVENESS†

CLAIRE KNIGHT

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@durham.ac.uk

Keywords: Visualisation, Understanding, Evaluation, Usability

ABSTRACT

Providing evaluations of visualisations is one way to demonstrate that they support a purpose and are adequate for the role claimed for them. The problem in doing so is that there is no central source of evaluation issues that one can use a subset of for this purpose. There is also very little in the way of agreement over what constitutes a good visualisation hence the evaluation criteria differ. There are the human-computer interaction ideals, the slightly differing ones from usability engineering, those from the visualisation community, and also the need to be able to support the variable abilities of the users. Graphics, as the medium behind visualisation, may support greater bandwidth, but is also prone to more likes and dislikes than other forms of interface. The concept of visualisation effectiveness and therefore ways of evaluating visualisations provide the focus for this paper.

1. INTRODUCTION

The issue of evaluation is a thorny one in many areas of computer science. Never more so when interfaces that humans have to directly work with are involved. The other aspect of this is that there is a great variability between the users of any given system, even within apparently restricted domains. One of the visualisation issues that has only been partially addressed is that of visualisation evaluation. Some work has been done on providing guidelines for interfaces (rather than the visualisations themselves), and of applicability of visualisations but little on the whole package of usability of visualisations (the interface and the visualisation). A visualisation can only be considered to be effective if users can achieve results using it, without extra cost on their part compared to any alternative way of realizing the same result.

Visualisations can be considered to be effective from two main perspectives and it is those that will be the focus of this paper. Firstly, there is the suitability of the visualisation for the tasks that it is intended to support. Secondly there is the suitability of the visualisation at the representation and metaphor level as to how well the graphics supports the data. This will have a direct bearing on standard usability, i.e.

† Department of Computer Science, Technical Report 03/01, April 2001 *and*
Workshop on the Fundamental Issues of Visualization, Proceedings of CISST 2001, Las Vegas, June 2001.

how easy the interaction and use is perceived to be, and also on the tasks that the visualisation is supposed to support.

This paper first presents a brief overview of the pertinent literature. Much of this is from the information visualisation community as many ideas from human-computer interaction and usability engineering relate to the interface rather than the visualisation. Whilst these are important areas of the whole, it is necessary to be able to design and evaluate the visualisations in their own right. After this the concept of visualisation effectiveness is introduced, and then discussed in more detail as the issues that are important are presented. The limitation of any given visualisation is also introduced as a related idea before conclusions as to areas of future work are provided.

2. BACKGROUND

There have been a variety of views as to what constitutes a good visualisation. Some of these have focused on the provision of the ability to adhere to Shneiderman's visual seeking mantra [3] of overview first, zoom and filter, and then detail on demand, whilst others have focused on the cost-benefit of using the visualisations to locate information [4]. Whilst both of these views are about the visualisation's ability to present the information in certain expected forms, neither of them really address the issues of domain and task (i.e. the context) that will have an impact on the (perceived) usability and the time taken to locate information.

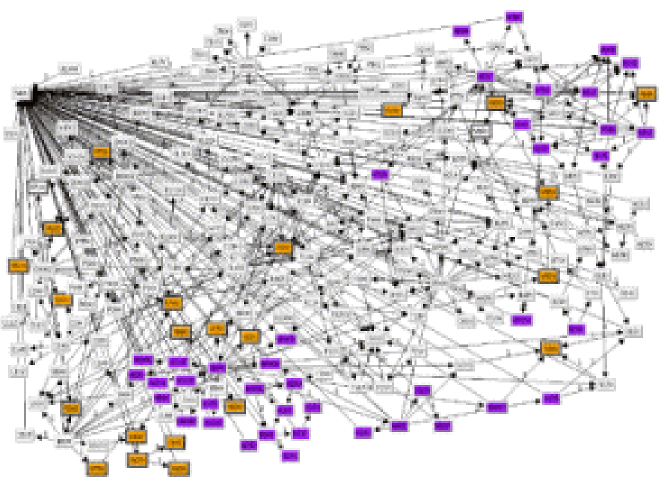

Figure 1 - Call graph of a medium sized software system

There is also the attitude of some who consider any given visualisation to be applicable to all data sets. There are characteristics of any data set that would dictate as to the structure of the graphics necessary to illustrate the component parts of that data set and the implicit and explicit relationships between those components. Examples of this started with some of the earliest forms of visual display; the graph structure. Software visualisation tries to address the identified shortcomings of this approach (for example [1], [5], [6], [11] with an overview in [8]) but also has to try and overcome the comfort factor of a familiar display, and the view that the representation of nodes and arcs should be suitable for all data that needs to be represented about software. There is plenty of evidence to show where the data used in this way has far exceeded the capabilities structure being used to display it as can be seen in Figure 1, but many still insist on using it. The same can be said of many examples of visualisations developed for the various areas to which visualisation is applied; some consider their technique to be entirely universal and do not consider the impacts of a separate domain, or the possibility that the data set may range in size from (theoretically) 0 to infinity.

It has already been states that the usability aspects of the overall interface are not the focus of this paper. It should also be made clear that the focus of the paper is not to concentrate on the adequacy of the lower level graphics or the glyphs. Perceptual issues from a cognitive science perspective can be useful feeders into this process but the measurement and evaluation of them requires properly controlled studies. It is also an issue that the perceptual whole may have a different outcome that the sum of the perception of the parts; i.e. the perception of the glyphs in isolation differs from when they are seen as part of a large graphical display. Others have done much more work in this area, such as Wittenbrink et al. [9] with uncertainty data, Globus and Uselton [7] with numerical analysis and algorithms, and Tufte [10] in a variety of ways. These would be suitable as a sub-part of the data adequacy part of the effectiveness equation presented in the next section. This work seeks to view the visualisations at a higher level.

3. *Effectiveness Equation*

There is a need to try and describe what is a good visualisation without recourse to the interface in which a visualisation is embedded. This is a separate issue and one which could reasonably see a visualisation considered effective but the whole application (visualisation and interface) as unusable. It is also necessary to consider more than the speed taken to locate information. This type of measure has to be careful to take into account the abilities of the users, the domain experience they have, their experience with the visualisation interface and the visualisation representation used, and finally their appreciation of the use of such graphics for their tasks. There is also the issue of when a visualisation is intended to support more general browsing activities as well as pure goal directed activities. The former would obviously take longer and thus cause statistically poor results however usable the visualisation may be.

Since this work has stemmed from visualisation work in the software domain (visualisation of various facets of software systems for the purposes of understanding), there are evaluation influences (as previously mentioned) from information visualisation, but also from usability engineering, human-computer interaction, virtual reality, and program comprehension. There is a problem in that system and software visualisations are often seen as being all things to all people! In reality several factors from each of these areas can be considered important.

A high level view of the effectiveness of a visualisation has been developed based on this prior information and the experience of working with software visualisations and industry. This effectiveness can be expressed in the form of a simple equation:

Effectiveness = suitability for task(s) + suitability of representation, metaphor, and mapping based on the underlying data.

A simple example of where current work that applies to software visualisations [2] falls short is that the display is considered to only be 2D, thus many of the issues surrounding 3D navigation and orientation are not addressed, and the answers to the existing criteria could be misleading if the number of graphical dimensions are not taken into consideration. Another is through the concept of usability. Many usability studies do consider the task the user is trying to achieve, but do not cover the possibility of investigative type tasks that are common when carrying out comprehension. The user may jump around the data, revisit areas, and the ultimate success of the task may actually be that something is not there; such as there is no direct impact of a change (again within the software visualisation domain).

This effectiveness equation is a broad definition and the content of the two extensive categories is described further in this paper, along with some of the issues that ought to be considered in relation to the categories.

4. *Domain Influences*

The domain of any application or interface can be important because of the assumptions involved. These assumptions are an important part of knowing and working in a domain because of the contexts that can be inferred in given situations. By inferring information and knowledge time can be saved. The problem comes when this information is encoded, such as in a visualisation, and then that visualisation is applied without modification or consideration of such embedded assumptions. Obviously some domains, or sub-parts of a domain, are similar enough that this is possible. In certain cases it may also be that the visualisation or interface does not rely on domain assumptions (regardless of what else it may rely on) therefore reuse across domains is both possible and desirable.

Another aspect of domains that may influence a visualisation are the tasks that are necessary and therefore need to be supported. Because of the specific nature of most tasks, primarily because of the goals, there is a need to make sure that these are the ones best supported. Should a visualisation be able to easily incorporate support for more tasks then all the better in terms of wider usage, users, and applicability, but this is essentially an added bonus. Whilst the importance of the task and domain should be considered influential when considering a visualisation, and for any system an understanding of requirements is necessary, this does not suggest that a full task analysis is necessary. Indeed for retrospective evaluations of various visualisations for comparison it would not make sense to carry out full tasks analysis because of the impact of introducing the visualisation. As has often been stated in the software engineering community, the introduction of any software system into an environment has the effect (desired or not) of changing that environment.

It should be more important to the visualisation designer/evaluator as to whether a combination of the task support and the fit with the data creates an effective visualisation. This is also another reason a full task analysis would be of limited effect given the costs involved. The data and the tasks of a visualisation are inextricably linked and obviously then have an impact on each other.

5. The Importance of Data

The structure of any dataset is important. It is necessary to consider this for storage and access, irrespective of whether a visualisation is involved. It is also very necessary to consider this for visualisations partly because of the access required to the data to be able to visualise it, but also to efficiently represent the data items and their relationships (explicit or otherwise). These relationships may well be the key to providing an extra level of understanding, of reducing the amount of information being presented in a coherent manner, or even allowing apparently hidden connections or impacts to be seen. Indeed all of the things for which visualisations are presented as being good for.

The structure of the data allows assumptions to be made about the type and amount of data expected at any given point. These assumptions then feed back into the process of finding suitable metaphors and representations for the visualisations. In the same way that the domain can cause assumptions to be embedded within visualisations, the data can also cause the same knowledge encoding. Often this process is transparent to the creator of the visualisation because of their familiarity with both.

As with the domain, it is worth considering where a visualisation can be reused across data sets. It may be that part of the visualisation does not rely on assumptions about the data, or that the assumptions can easily be replaced with others relating to the change of dataset.

6. Visualisation Limitations

Some in the visualisation community may believe that the ultimate visualisation would be one that is regarded in the same way that static two-dimensional nodes and arcs are used to represent a variety of graphed data. The problems with this are two fold. The first is that this approach has identified deficiencies such as scaling that techniques such as zooming, or fisheye have only partially addressed. The second is that the visualisations that are being created today may be ever more sophisticated in terms of graphics, dimensions, and display technologies, but in order to support this sophistication the is a need to rely on data and/or task assumptions.

Some limitations are likely in any visualisation therefore there is a need to recognise and identify them, even highlight them for future users. Therefore people can see where they need to adapt and enhance the visualisation; or to consider using some other visualisation. Limitations in a visualisation should not be seen as a bad thing (often lowers scores in evaluations) if the main purposes of the visualisation (tasks and data) are fulfilled effectively. It seems there may have to be a shift in attitudes to the evaluation of visualisations. In the same way that a wordprocessing package is good for producing documents, it is less good for storing large amounts of data. In this case a database would be a better choice of application. To try and use one for the functionality required of the other would be considered to be foolish. The concept of *horses-for-courses* is not new, and the same common sense should be applied to visualisations.

7. CONCLUSIONS

Because of the assumptions that the domain and the data structure impose on any visualisation there is a need to:

a) Know about the domain

b) Know the limitations of the visualisation being created/designed/evaluated

These assumptions could be removed from the process but it would then be an issue as to whether the visualisations that resulted were actually the best ones possible. Generic solutions generally provide better results once they have been specialised to the specific needs of a problem. There is also not a problem with embedded assumptions within visualisations if their presence is acknowledged and worked with. By harnessing them then visualisations can be made more powerful for a specific problem, but also have the facility for being altered or re-specialised for other problems.

This paper has tried to highlight some of the issues of judging whether a visualisation is effective or not. There are obviously important results from human-computer interaction and usability literature that can and should be used for the general interface aspect of the visualisation application. What is necessary is to find a way to provide independent guides as to whether a visualisation has fulfilled the aims for which it was created, and even to facilitate comparative studies. There is obviously a need to be refine the contents of the domain and task categories, but this is a step towards acknowledging that visualisations are not necessarily universal, but that this is not necessarily a bad thing.

ACKNOWLEDGEMENTS

This work has been supported by the EPSRC project VVSRE; **V**isualising **S**oftware in **V**irtual **R**eality **E**nvironments. Thanks also to Malcolm Munro for support on related research.

REFERENCES

[1] **C. Knight** and **M.Munro**, *Comprehension with[in] Virtual Environment Visualisations*, Proceedings of the IEEE 7th International Workshop on Program Comprehension, pp4-11, May 5-7, 1999.

[2] **M.-A. D. Storey, F. D. Fracchia**, and **H. A. Müller**, *Cognitive Design Elements to Support the Construction of a Mental Model During Software Visualization*, Proceedings of the 5th IEEE International Workshop on Program Comprehension, pp17-28, May 28-30, 1997.

[3] **B. Shneiderman**, *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*, Proceedings of the IEEE Conference on Visual Languages, September 3-6,1996, pp336-343.

[4] **S. K. Card**, **P. Pirolli**, and **J. D. Mackinlay**, *The Cost-of-Knowledge Characteristic Function: Display Evaluation for Direct-Walk Dynamic Information Visualizations*, in Readings in Information Visualization: Using Vision to Think, Stuart Card, Jock Mackinlay, Ben Shneiderman (Editors), Morgan Kaufmann, February 1999.

[5] **P. Young** and **M. Munro**, *Visualising Software in Virtual Reality*, Proceedings of the IEEE 6th International Workshop on Program Comprehension, pp19-26, June 24-26, 1998.

[6] **S. G. Eick**, *Maintenance of Large Systems*, Chapter 21 of Software Visualization : Programming as a Multimedia Experience by John Stasko (Editor), Blaine A. Price (Editor), Marc H. Brown (Editor), MIT Press, February 1998.

[7] **A. Globus**, and **S. Uselton**, *Evaluation of Visualization Software*, Computer Graphics, Vol. 29., No. 2., May 1995.

[8] **C. Knight**, *System and Software Visualisation*, To appear in the Handbook of Software Engineering and Knowledge Engineering (2 Volumes), mid 2001.

[9] **C. M. Wittenbrink**, **A. T. Pang**, and **S. K. Lodha**, *Glyphs for Visualizing Uncertainty in Vector Fields*, IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 3, pp266-279, 1996.

[10] **E. R. Tufte**, *The Visual Display of Quantitative Information*, Graphics Press, February 1992 reprint.

[11] **T. Ball** and **S. G. Eick**, *Software Visualization in the Large*, IEEE Computer, pp33-43, April 1996.