ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Jessie Kennedy and Peter Barclay (Eds)

Interfaces to Databases (IDS-3)

Proceedings of the 3rd International Workshop on Interfaces to
Databases, Napier University, Edinburgh, 8-10 July 1996

Paper:

Interface Design: More Craft than Science? (Invited Paper)

Matthew Chalmers (Ubilab, Union Bank of Switzerland, Zurich)

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

Interface Design: More Craft than Science?

Matthew Chalmers

Ubilab

Union Bank of Switzerland, Bahnhofstrasse 45, 8021 Zürich, Switzerland

chalmers@ubilab.ubs.ch

<http://www.ubs.com/ubilab/Staff/chalmers.html>

Abstract

Interface design can overwhelm the effects of better or worse database engines. Progressively more systems use graphical interfaces but as with any new technologically-driven medium it takes time for notions of ‘good’ design to form and disseminate. Nowhere is this more obvious than in the nascent field of 3D interfaces where — much as in the days of font overkill which followed the introduction of laserwriters — 3D is often used for its own sake. This leads to designs that may initially be eye-catching but later turn out to be cluttered and difficult to work with. I will discuss some issues that make the design and evaluation of information visualisation systems complex such as numerical metrics for interface assessment, the use (and abuse) of metaphor, taking advantage of 3D perception, and shared information environments. This discussion will draw upon images and experiences from the Bead system in order to examine the discipline we are involved in: is it more craft than science?

1 Introduction

When we build interactive systems, we produce structures as complex as the buildings we work in. Essentially, both are structures where we obtain, store and produce information, and interact with other people. When we look at a building and say whether it is ‘good’ or not, we do not only look at the functional or engineering aspects. We think of the way it allows us to move and work, as well as its aesthetics, how weatherproof it is and whether it safely supports the weight of the people within it along with their paper, furniture and machines. We tend not to think of interfaces and computer systems in such varied and complex ways. We tend to look at our work as civil engineering and not architecture. Although engineers do consider the aesthetics of their designs, it is not given nearly the same weight in the practice or education of their discipline. Architecture consists of both the objective and the subjective or, in other words, the scientific in combination with the artistic. They have been blended together over centuries in a craft based on designing complex spatial structures, and I feel that this kind of approach is a better paradigm for interface designers to follow than pure, or traditional, computer science.

I would expect that the majority of people who attend this workshop are computer scientists, and a number are likely to think about their systems in traditional computer science terms. This can mean an engineering-like view based purely on technical proofs of the availability of functionality, non-interference in database operation, expressibility of data access in a given formal query language, and so forth. More modern HCI practitioners have taken on a wider architecture-like view, as exemplified also in textbooks such as [10]. Computer science, psychology, graphic design and sociology each play a part in the way to think about interactive systems and interfaces.

It matters how you think about interface design. Clearly, it influences what you will want to build. It defines what type of issues, criticisms and assessments you take on as relevant. It determines what you consider to be a good system or even a working system. Therefore I would encourage the reader to accept that what I and most others in this workshop do nowadays is best considered not as science, but as craft. Not engineering, but architecture.

I will put forward some issues and relate some experiences with my own system, Bead, which may serve to clarify my opinion in this matter. At best I might win over a few ‘converts’, but at least I hope to provoke discussion about how we design interfaces to database systems.

2 Bead

Part of the basis for this paper is the experience I have had developing a system for information visualisation called Bead. Bead lays out a set of objects to make a map or landscape, using either simulated annealing or force-directed placement. An example is shown in Figure 1. The most common data type mapped out has been textual documents — articles from a bibliography — although in the past year my group has also been applying Bead to financial data. In the following discussion, I will usually refer to bibliographic data.

Figure 1. A set of 831 bibliography entries has been laid out by making proximity approximate similarity in word usage. Coloured clusters, based on local density in the layout, have been added to the basic landscape model. Clusters serve as static imageability features for orientation and navigation. A few large clusters dominate, although several smaller clusters exist.

The system approximately represents the similarities and dissimilarities between the articles by their separation in the landscape i.e. the high-dimensional distances defined by similarities in word usage are approximated by low-dimensional (three-dimensional) distances in the visualisation. At any moment, one can look at the discrepancy between the current low-dimensional distance between two articles and their high-dimensional distance, and so decide whether to add a force to push them further apart or pull them closer together in order to reduce this error. An additional gravity-like force is used to attract the documents towards a ‘ground plane’. This system of forces is implemented using spring models in order to lay out the landscape, and work on layout algorithms based on error minimisation has been the focus of a substantial part of the work on Bead in the past years. Algorithms have been developed to take the complexity of each iteration of the layout process for N objects down from the standard $O(N^2)$ to $O(N \log N)$ and most recently to $O(N)$ [3].

Once the errors of document separation in the landscape have been minimised, one has a set of positions of graphical objects which are meshed together using Delaunay triangulation to make a mostly flat landscape. Thereafter a number of features are used to enrich the landscape with features for legibility and imageability [4] when visualised in a shared virtual environment. These include static features which serve as landmarks such as the landscape’s shoreline, local areas of roughness, and coloured clusters of related articles. Dynamic features are also shown as each user moves through the shared virtual environment, for example pop-up titles and topics. These features appear and disappear in accordance with each user’s field of view, the local frequency of occurrence of words, and ‘popularity ratings’ of documents based on histories of word search and document selection. An example scene is given in Figure 2, below.

3 Numerical Metrics of Quality

The spring models used for layout give rise to a convenient measure of global layout quality, *stress*, widely used in assessing multidimensional scaling and related techniques. This is the mechanical stress of the spring system, and is essentially the sum of squared errors, normalised by the sum of squared low-dimensional distances to favour compact layouts and to allow some comparison between different data sets. Bead brings the stress of small sets of articles (of the order of $N=100$) down to around 0.1. Another order of magnitude in N leads to a stress of around 0.2.

In the course of a layout, stress is the key value used to determine how well the program is progressing. Sometimes

Figure 2. Static imageability features such as clusters help with orientation and navigation, but dynamic features are often better for revealing detail. A few documents are dynamically and randomly chosen to be highlighted with a medium colour and have their titles shown, with a sampling bias based on nearness to the eye. Topic words are also dynamically placed in the scene according to density of occurrence in the field of view and word usage history. Clicked-on words start searches which colour documents white.

the stochastic nature of the layout process will lead to small variations in the final stress of repeated runs even when all input parameters and data are held constant. There are many such parameters controlling the layout process e.g. the stiffness of the springs, their damping, and the metric of high-dimensional distance. The latter feature has been found to have a strong influence in generating low stress values. Having a metric which leads to good discrimination between objects is important.

Recently some layouts were made comparing high-dimensional distance metrics while holding other parameters constant. Unsurprisingly, rather different layouts were generated, with some deviation in stress values. The layouts were fed into a visualisation tool for examination, and what we found rather surprised us: some of the layouts with the average or slightly higher stress values seemed to us to be clearly (although subjectively) better than others.

In what we felt to be the best layouts, clusters of objects were better separated from each other, and internally were more tightly bunched. Also, when comparing localised pairs within clusters, there seemed to be a better fit or ‘sense’. The separated tighter clusters meant that they also served better as landmarks. They were more identifiable and distinct, and added imageability features could accentuate this to good effect. We began to consider the way the stress metric worked, and whether we might take into account cluster tightness, but then began to wonder how one might consider the effect of pop-ups and labelling regions with topics. Also, there are features such as font size and colour, the size, colour, shape and behaviour of each document, and so on. Collectively and interdependently these convey the feel of the set of documents, along with the layout positions, and thus form an essential part of the final presentation of data to the users. Their presence or absence in different layouts can often override differences in stress.

I was reminded of a paper by Bryce Allen where the effectiveness of a much simpler (textual) display of bibliographic information could be varied significantly merely by varying the order of title, author, &c. [1]. He noted that “by making a minor change in the way an information system presents data, designers can change the pattern of usability of the system. Simply altering the order of presentation of data elements in a bibliographic display can transform an information system from a ‘one-size-fits-all’ system to one in which certain users, because of their higher levels of perceptual speed, are able to achieve significantly better performance.” The possible presentational variations in the structure of his displays were small compared to those of the moving perspective views of the Bead information landscape.

There is more to this issue than saying that stress is not the right metric for assessing the spring system. In general, if information is in the ‘right’ position on a screen, it doesn’t mean that the user will perceive and use it the way we expect. It can be skipped over because something else nearby looks more obvious, interesting or relevant, because the

surrounding items mostly don't seem interesting enough, or because the user has already seen so many 'relevant' items that their interest is beginning to move on elsewhere.

When checking the progress of an ongoing layout process, or when comparing two layouts which have the same input parameters and employ the same output techniques, stress offers a useful tool to assess quality. However, one should be aware that more general use of such numerical measures — based on simple mechanistic features such as geometric distance, presence on the screen or ranking in lists — ignores the overriding perceptual and design effects which may transform the effectiveness of the system in the hands of the user.

4 Perception and Metaphor

To some extent the information design in Bead was founded in the familiar linguistic notion of similar items being metaphorically 'close together' [7]. In addition, there were features that had familiar spatial interpretations: rough areas of the landscape were less reliable and more difficult to handle, peripheral topics in the data tended to be represented by documents pushed out to the periphery of the shore, while dominant topics were more central. Our everyday familiarity with landscapes, and also the historically developed culture of maps and their interpretation, suggested that the landscape metaphor was a promising approach to take in enriching and improving the Bead visualisations.

Moving to the landscape model for Bead increased average stress levels when compared to the earlier 3D point cloud model, but the landscape was superior in terms of legibility and imageability [2]. It did not matter if similar documents were closer to each other inside the point cloud model, as occlusion made most of them difficult or impossible to see. Patterns and clusters were 'not there' as it was so difficult to perceive them. Moving through the structure didn't help much as one could not maintain the relationship of one's position to a useful ground plane or to the patterns within the data that had been laboriously calculated by the system.

Perceptually speaking, we do not live in a fully 3D world. In the past I have sometimes referred to our everyday experience as being '2.1D'. The variations on the surface of the earth such as hills and valleys are relatively small compared to the extent of the earth's surface. Gravity makes the vertical axis a special case, as pointed out by J.J. Gibson [5], and we should not expect that attributes mapped to height will be treated in the same way as those defining the base plane in a visualisation.

Another way of looking at this is to disambiguate 3D vision and 3D structure. We see in a three-dimensional way, with perspective compressing the apparent size of objects which are further away and expanding objects which are close. We use 3D vision very skilfully in the 2.1D world. We can use perspective to rapidly flick between an overview of large areas of the environment and detail on closer objects. One of the main reasons for motion is to reallocate the finite resource of what we can see in detail. We move towards what is distant or peripheral, and smoothly make our view of it more detailed. Perspective lets us maintain the context of what is more distant in the environment, including where we came from and where we might move on to next. Motion is thus a declaration of interest, a request for more detail. These techniques are hard to apply in strongly 3D structures which are as large and complex in height as they are in length and breadth. Occlusion stops us getting an overview and using our skills in manipulating perspective and wayfinding. In nature this is not often a problem, as we rarely use or find large scale 3D structures.

In a metaphorical way, Bead employs some features of landscapes because I want to use 3D vision skills to work with a 2.1D information structure. As a linguistic tool, metaphor requires a certain slackness in interpretation, as otherwise it is not expressive enough to be useful. Excessively tight interpretation of a metaphor where every single feature is assumed to be shared between the two types of object is doomed to failure. In visualisation, photorealism is a means to end rather than an end in itself.

As Gibson and others such as Don Norman [8] have discussed, potential use will define how we perceive an object. Something affixed to a door must afford my grasping and pulling it before I would perceive it as a handle for opening the door towards me. Metaphors often fail when there are similarities in structure but not in use, and this problem seems to be increasingly common in 3D interfaces.

A number of systems (which I will leave unnamed) have appeared which basically show a graphical structure which is isomorphic to the database structure or query result structure, but nevertheless the design is poor. For example, a ranked list of objects matching a database query might be mapped to a set of floors in a building, best match highest, and then each object has its subfields mapped to different rooms in the building where other properties or attributes are shown, for example as objects within the rooms or images on walls and surfaces. The mapping between the two structures is fairly obvious and might be shown to be formally correct, but I would contend that the design is bad

because the floors and walls inhibit the ability to scan and compare items, to look for patterns, and so forth i.e. the metaphor fails because the graphical representation does not let us use our 3D vision skills — the uses or activities that make visualisation powerful. Such naive mappings, dressed up as metaphors, may show off fancy graphics engines, but they don't serve well as visualisations. We have to think about how the structures will really be used and perceived, as graphical representation is a linguistic issue with all the complexity and subtlety that implies. Isomorphism is not enough.

Lastly, perception does not only involve the fleeting appearance of objects in our visual field, or other transient sensory information. It also involves cognitive aspects such as memory. Have you noticed that if you get a new car, cars of the same model start to become so much more noticeable than before? Furthermore, we may infer some feature of our environment because of consideration of the sensory and memory information we have at hand. To use a paraphrased example from [9], I might be new to Edinburgh and have been told that a particular restaurant is on Grindlay Street. Then, after meandering through the town I suddenly find myself in front of that restaurant. I might infer the name of the street and my position in the city without having to read any other sign or map.

Interfaces often stop us from making use of everyday memory and inference abilities. They tend to present information to us without regard to what we have done before, for example in seeing how a current query relates to earlier work, thereby allowing comparison or combination of earlier results. They tend not to allow us to get a view of the database that lets us see how each item or query fits in to the wider scheme of things.

Bead lets us employ spatial memory as the landscape is a persistent referent upon or in which successive searching and browsing activities are carried out. Memory of the locations of previous work, especially in relation to landmarks such as shorelines and clusters, can be used to help find one's way through the model toward a document or set of documents. By having a global structure — a 'sense' to the layout — then people can infer useful features. For example if a currency trader enters a query about 'banks' and finds one match in an area with a topic word such as 'river' and another in an area labelled 'finance' then he can immediately infer something about the content and utility of each.

5 Shared Information Environments

Another aspect of everyday work which is too often ignored in database interfaces is that of shared use. We use the information of others so often in our workaday world that we hardly notice it. Especially with more complex information, people's opinions and actions are often more reliable than our own data-centred analyses. When I am thinking about going to see a film, then reviewers' and friends' opinions will be as or more important than a plot summary.

With almost all information stored in databases, it is clearly important to avoid interference so as to maintain database integrity and quality. Unfortunately this principle tends to extend too far up into interfaces, ignoring the way that information is used in the real world. We can't see if other people are working on the parts of same data, or in what way. We can't see the history of an object such as who created it, who has been working on it, and who has been reading it. An interesting example of what can be done in this regard this was given in [6]. Documents had traces of past use shown on them, showing which subregions had been the focus of earlier work either in creation or in reading. This helps answer practical and important questions such as: Which sections of a document have been read by various categories of users? Who were the last people to read a section, and when?

The pop-up topics and titles of Bead use history information to help interpret and display the visualised data. Histories of past use of visible documents are examined in order to find which words and documents were 'significant' in that area i.e. what were people looking for in that area, and what documents were interacted with in that area? This approach tends to sidestep some of the problems of text understanding and summarisation that dog information retrieval and AI-based approaches.

Although these are only our first steps in using histories in Bead, we feel that finding out what tasks and activities people were involved in when accessing and using data, and perhaps how and why they changed it, may give very useful information to subsequent users. Similarly, awareness of the ongoing activities of others is an essential part of everyday work. There still remain many issues unresolved, for example the obvious questions of privacy and invasiveness, and also showing the ownership and status of an object e.g. we might not be able to see so much detail of a rough draft of a document or design, but more refined versions may be represented so as to invite access for other people's comments. Awareness of future use may also be important. For example, I may see a document such as a report I am working on as being more significant if I see that my boss plans to look at it tomorrow morning.

Such issues are handled in quotidian ways in our real world workplaces, but although CSCW research has made some

advances in addressing them, we have yet to fully address the challenge of handling them in the design of interfaces to database systems. Our designs should relate to the culture of work: of ownership and privacy, of history of use, current action, and future plans. If our systems better fit the way work happens in the real world then they will be the better for it, as they will better suit the people who do that work.

6 Conclusion

In considering some of the ways in which designing good interfaces to database systems is hard, I have tried to push on the notion that there are more issues which are relevant to interface design than those from computer science. As designers we should be aware of functional aspects but we should also be aware of the range of complexities and subtleties that arise when we attempt to convey information to people, and to support their work. We should use numerical and computational metrics of quality, but know the limits of their utility. We should use metaphor, but with the knowledge that using such a rhetorical device requires appreciation of how it will be used and interpreted. We should be aware of how important a role use plays in the acceptance of graphical metaphors, especially when we feel the pull towards greater naturalistic ‘realism’. We must face up to the way that buildings and networks make people less and less likely to be isolated in their work, and the way that social issues come in to effect.

We should accept that handling this wide range of issues *is* our craft. We should be competent practitioners of informatics, but also familiar with social and psychological issues including the perceptual, linguistic and aesthetic — each of which is unlikely to soon be reducible to systematic, objective formulations of knowledge i.e. to science. Therefore, to paraphrase Michel Beaudouin-Lafon, we should not call what we do ‘computer science’ any more than writers consider that they do ‘pen science’ or architects ‘brick science’. We have to raise ourselves up and gain a broader perspective. That is what our craft is all about.

7 Acknowledgments

Thanks go to my colleagues in the Visualisation group at Ubilab, past and present, for helping me move forward with Bead and explore these ideas. I’d also like to thank a number of non-Ubilab people for recent conversations on the architecture/engineering issue, in particular Stu Card, Steve Draper, Mountaz Hascoët-Zizi, Peter Pirolli, Ben Shneiderman and Lisa Tweedie.

8 References

1. Allen B. Perceptual Speed, Learning and Information Retrieval Performance, In: Proc. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 94), Dublin, July 1994, pp 71–80.
2. Chalmers M. Using a Landscape Metaphor to Represent a Corpus of Documents, In: Proc. European Conference on Spatial Information Theory. Published as: Frank A & Campari I (eds.), Spatial Information Theory, Springer Verlag Lecture Notes in Computer Science 716, 1993, pp 377–390
3. Chalmers M. A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data. To appear in Proc. IEEE Visualization 96, San Francisco, October 1996
4. Chalmers M Ingram R & Pfranger C. Adding Imageability Features to Information Displays. To appear in Proc. User Interface Systems & Technologies (UIST 96), Seattle, November 1996
5. Gibson JJ. The Ecological Approach to Visual Perception, Lawrence Erlbaum, 1979.
6. Hill W, Hollan J, Wroblewski D, McCandless T. Edit Wear and Read Wear, In: Proc. Human Factors in Computing Systems (CHI 92), ACM, 1992, pp 3–9
7. Lakoff G & Johnson M. Metaphors We Live By, University of Chicago Press, 1980
8. Norman D. The Psychology of Everyday Things, Basic Books, 1988
9. Passini R. Wayfinding in Architecture, Van Nostrand Reinhold, 1992
10. Preece J et al. Human-Computer Interaction. Addison Wesley, 1994