UNIVERSITY OF LONDON THESIS

Degree phD Year 2007 Name of Author BROCK ROBERT CRAFT

A Sketching-oriented Design Method for Information Visualization Software

Brock Robert Craft

A dissertation submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy

of the

University of London

Department of Computer Science

University College London

February 2007

Declaration

I, Brock Craft, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

The aim of this research is to describe a useful approach for supporting creativity and problem-solving in the design of Information Visualization software. This type of software is useful for helping people to understand large or complex collections of data by making the data easier to see and use. Because it can be so helpful, many people are motivated to create visualization software to address their own unique problems of understanding data. However, the techniques which visualizations use to enhance cognition of data are not widely known. Also, there are currently few resources which comprehensively describe a method for designing novel visualizations. Consequently, people who seek to build new Information Visualization tools are left to consult design examples, guidelines, and reference models, which do not adequately describe the visualization design process. The key question of the research concerns how Information Visualization methodologies should account for representation of the user, existing visualization design knowledge, and sketching. Given that the current methods of Information Visualization design are incomplete and show evidence of significant shortcomings, how can novice visualization design teams bridge these gaps by using methods from other design disciplines to successfully create effective visualizations? To investigate this question, several studies were conducted. Also, a design methodology called SoViz was developed. It incorporates a participatory design approach, using sketching and visualization design patterns to support creativity and problem-solving. A prototype was designed using the SoViz approach. The key contributions of this thesis are results which show that Information Visualization designers can benefit from using this method. The thesis presents the results of using SoViz to create an Information Visualization prototype and describes the theoretical consequences for Information Visualization methodology.

Acknowledgements

This research would not have been possible without the love and support of a great many people. I'm happy to be able to use a bit of space to thank everyone who has been so helpful to me. Because of them, it has been a truly rewarding experience.

Foremost, I would like to acknowledge the enduring support of my advisor and friend, Dr. Paul Cairns. I have always been able to seek Paul's advice and to receive carefully considered and thought-provoking feedback. He was also there to reel me in from the many missteps and tangents that arose during this adventure, and to clarify my thinking. This work would not have been possible without his steady, sure guidance. I'd also like to thank Professor Angela Sasse for her key insights and guidance during the tricky parts – not to mention a gently flowing stream of lattés – and for generally making my life as a PhD student easier and more pleasant. I would also like to convey my gratitude to the examiners of my dissertation, Professor Emeritus Robert Spence and Professor Janet Finlay, who examined my work with care and precision and provided extremely valuable feedback.

I'd especially like to thank the members of the UCL Beacon Project, who were willing and eager participants in this research. I am very grateful to James Hetherington and Peter Saffrey for helping me bring the work to fruition. Their enthusiasm and continued interest was invaluable.

My family and friends have also been a source of unending support. My parents have accepted my academic exile to a faraway land with kindness and love. My friends have also cheered me on and kept me going. Particular thanks go to Lidia Oshlyansky for always believing in my ability and helping me to get this project off the ground. I am also grateful for the support of Barbara Weidman, who spent many hours picking over the thesis to streamline the text and keep it readable. I also thank my dear comrades Jason Geistweidt and Robin Gravenhorst, who have been a continual source of intellectual stimulation and amusement, these many years. Thanks are also due to my colleagues at UCL Interaction Centre have made this experience both fun and stimulating.

During the course of this research, I consulted many people who are professionals and experts in their disciplines, some of whom are named in this thesis. Without reward, and without exception, these individuals have provided expertise, guidance, and suggestions to make my work stronger. I owe each of them a debt of gratitude. Their help has been both collegial and enthusiastic and has taught me the importance of giving back.

Table of Contents

Chapter 1: Introduction

1.1. Introduction

Over the past three decades, and in parallel with the rise of low-cost, high-performance computing, research into Information Visualization (IV) has gained momentum. Information Visualization software is useful for helping people to understand large or complex collections of data by making data easier to see and to interact with. Because this kind of software can be so useful, many people in industry and academia are motivated to create Information Visualization tools to address their own unique problems of understanding data. However, designers new to visualization have few resources to draw on. Although the techniques which visualizations use to enhance cognition of data are widely known, how they work and how to create them is not fully understood. The motivation for this research is that currently, there are few methodologies which comprehensively describe these techniques. The present discipline is laying the groundwork for them, as is evidenced by numerous examples of novel visualizations, design guidelines, taxonomies and reference frameworks. Also, there are a number of systems which have attempted to automate the creation of Information Visualization tools and thus, to make concrete some of the procedural steps which are important. However, what is often not acknowledged by researchers in this area is that although building Information Visualization tools is a technical process, in terms of quantity of software engineering, it is also a creative design process. In addition to describing the technical requirements such as system architecture, information visualization design methodology should also account for creativity and problem-solving, which are parts of the design process. This knowledge is under-represented in the literature. There is very little discussion of the use of design techniques to support creativity and problem-solving. This absence is particularly surprising, considering that visualizations which offer creative visual representations, supported by usable interactions, receive the highest acclaim.

This research is centred on how methodologies of visualization design should account for the representation of the user, existing design knowledge and sketching, as aids to the creation of necessary tools. To investigate this question, several studies were conducted. The key result is confirmation that visualization designers can benefit from taking a participatory design approach. This approach involves active participation of end-users, the use of sketching to support design activity and of existing Information Visualization design knowledge in the form of design patterns.

Also, since user interaction and visual presentation are the key elements which set visualizations apart from other kinds of software, design techniques which support these should be fostered and articulated by Information Visualization design methodologies. Yet, these are rarely described. Much of the current design knowledge only implies the characteristics of a 'good' visualization. The ethic that informs the design of visualizations is that a 'good' visualization will make data easier for the user to understand and learn from. Implicit in this notion is that visualizations should be easy to use. Thus, representation of the user is important in the design process. However, even though work in the human-computer interaction (HCI) community has shown techniques such as participatory design to be very effective for creating usable software, they are not discussed in the Information Visualization literature. Evaluations, such as usability testing, are rarely reported.

Happily, other disciplines such as architecture and engineering have recognized that creativity and problem-solving can be supported and fostered during the design process. Moreover, this research shows that they can be beneficial to Information Visualization designers. To address the deficiencies in methodological knowledge and to lay groundwork for the development of new Information Visualization design strategies, four studies were performed. This thesis reports the results of this work and argues that design techniques which have proved useful in other design disciplines can also be used effectively in the early stages of Information Visualization design.

1.2. Nomenclature

As with any emerging discipline of study, the terminology of artefacts and concepts is often in flux. The field of Information Visualization is no exception. For the sake of brevity, it is common for practitioners to use the terms 'information visualization' and 'visualization' interchangeably, even though they have rather different meanings from a technical point of view. In fact, the domain of Information Visualization can be construed as a sub-domain of software visualization, an important distinction which will be discussed in Chapter 2. This ambiguity of terminology reflects and perpetuates a lack of consensus about the boundaries of the discipline. Currently, this uncertainty has led to the emergence of yet another area of research, referred to as 'visual analytics', which bears many of the hallmarks of Information Visualization and which may eventually subsume it. For the purpose of this document, we shall use the terms 'Information Visualization' and 'visualization' interchangeably.

1.3. Problem Statement

Design of visualizations is a complex and difficult activity. Thus, it is unlikely that any single methodology can address all of the aspects which are important. Inevitably, there will be holes in the design knowledge corpus. These can be seen as an opportunity for improving the current design knowledge and importantly, for making it easier for others to create new and useful visualizations for their own needs. The problem can be stated this way:

Given that the *current methods of Information Visualization design are incomplete* and show *evidence of significant shortcomings*, how can novice visualization design teams *bridge these shortcomings* by using *design methods from other disciplines* to create *successful visualization designs*?

The main contentions of this problem statement are italicised. They can be summarized as follows. This thesis holds that current *Information Visualization design knowledge* is composed of four major sources: examples, guidelines, design patterns and reference frameworks. It will show by two experiments that this knowledge is *incomplete and has major shortcomings*, owing to confusion about terminology, failure to account for the user and a lack of techniques to support ideation, creativity and problem-solving. The thesis will then show that other design disciplines, such as engineering and architecture, have grappled with these problems and have effectively used *other design methods, notably sketching* as a design aid to enhance design cognition and creativity. It will show how other design disciplines have characterized the design process generically as one of exploring and then reducing design alternatives. It will be argued that visualization designers can make use of design methods from other disciplines to *bridge these shortcomings* by incorporating sketching and visualization design patterns into visualization design activity. To study this possibility, a design framework called 'SoViz' was created. A research strategy employing qualitative methods was developed for studying the effects of using SoViz in a visualization design setting. This research was conducted as a third experiment, in the form of a case study. The results are presented here to demonstrate the effectiveness of using sketching and design patterns for visualization design: the case study *successfully* generated eight *visualization designs*. As a further measure of success, one of these designs was selected for implementation as a prototype. This thesis will demonstrate the *success of the visualization design prototype*, as measured by a fourth experiment. It will also show that the results of the case study contribute meaningfully to visualization design theory. It will present an analysis of the results which provide evidence that the use of sketching and design patterns is effective in the visualization design process. Sketching is shown to be useful during the exploring alternatives phase, whilst visualization design patterns are shown to be useful during the reducing alternatives phase. This contributes to and supports existing theoretical knowledge about visualization design.

1.4. Research Questions

The problem statement can be formulated as a group of several, shorter research questions. The chapters of this thesis address these research questions through literature review, experimentations, and analysis of results. The research questions are:

1. *What are the shortcomings of existing methodologies of Information Visualization design?*

The answers to this question, provided by literature review and experimental results, led to the formulation of **research question 2**:

2. What design techniques used in other disciplines can be used to enhance creativity and problem-solving in Information Visualization design teams?

The answers to this question are provided by literature review, which motivates an inquiry summarized by **research question 3:**

3. How can these design techniques be incorporated into a research approach for a real visualization design problem?

This question is answered by the formulation of a visualization design framework, called 'SoViz' and a two-pronged qualitative research strategy, using a *bona fide* visualization design problem. The results of the research lead to **research questions 4 and 5:**

4. What are the practical outcomes from using SoViz?

5. What is the significance for theories of Information Visualization design methodology?

These questions are answered by interpretation of the results of the case study and the production and evaluation of a visualization prototype.

1.5. Contributions

This thesis offers some answers to these questions in the form of the following contributions to the information visualization domain:

1. Identification of shortcomings in current Information Visualization design knowledge (Chapter 2);
2. Evaluation of two visualization design approaches: one which is commonly accepted, and one which has been newly proposed, and identification of their shortcomings (Chapter 3);
3. Development of a visualization design framework called 'SoViz', which accounts for design shortcomings of creativity and problem-solving (Chapter 4);
4. Development of a rationale for and a description of a two-pronged research method for Information Visualization design activity (Chapter 5);
5. Evaluation of the SoViz design framework by applying it to a *bona fide* Information Visualization design problem (Chapter 6);
6. Production and evaluation of a successful visualization prototype to assess the SoViz framework (Chapter 7).

1.6. Scope

These contributions make clear the scope of this thesis. The domain of research is Information Visualization. The contributions are intended to support practitioners in the Information Visualization community. The scientific visualization domain is relevant, in that it is concerned with visual representation of data and with data modelling of physical processes. However, the claims in this thesis are about theories which have emerged within the Information Visualization literature. Also, the field of human-computer interaction (HCI) is relevant, though not the focus of this research, for two reasons. The first is that this thesis claims that the visualization design process should include the input of end-users, a major tenet of HCI literature. The second reason is the ethic which drives research in visualization methodology. A successful visualization is intended to be a usable one. Systems which are not usable do not meet the objectives of visualization design, as is evidenced by a growing trend toward usability testing and evaluation of new visualizations which are reported in the literature. So, although literature in HCI strongly informs this research, the contributions of this work are most useful for the Information Visualization community.

1.7. Thesis Structure

Within the next seven chapters, this thesis aims to answer the research questions. Figure 1.1 presents an overview of the thesis. At a high level, the thesis can be conceptualized as having two major sections. In the first major section, composed of Chapters 2 and 3, the domain of visualization is described and shortcomings in visualization design knowledge are identified. This is done through both literature review and two empirical studies. The literature review presents an overview of the development of Information Visualization and the current knowledge about visualization design methods. A contemporary definition of Information Visualization is presented and its relevance to this thesis described. This is followed by an illustration of the shortcomings of current reference models for visualization design, which are evaluated by experimentation in Chapter 3. Two studies were performed to evaluate two methods: visualization design guidelines and design patterns. The conclusions drawn from these experiments lead to the second major part of the research.

This second section, composed of Chapters 4–7, comprises the bulk of the thesis. The results of an additional literature review of design methods are discussed in Chapter 4, which also shows how sketching can be characterized as a fundamental part of the design process. The consequences of this are important for information visualization design. This leads to the development of the SoViz sketching-oriented framework discussed in section 4.8, particularly the way it overcomes shortcomings of visualization design methodology. In addition, a research method is developed to study this new visualization design framework. Chapter 5 describes this research method. The two-pronged, qualitative research strategy is described and justified as preparation for a case study. This is the third study in this thesis and is presented in Chapter 6. The results of the case study show that the SoViz method is successful at supporting visualization design. To further test this, a visualization called 'CalViz' was developed, with the expectation that a successful visualization would further show the value of using the SoViz approach. Chapter 7 describes the development CalViz through all phases of the SoViz design process. It also presents the results of an experiment which was conducted to assess whether CalViz was a successful prototype. The results indicate that CalViz was successful in communicating with and motivating some of its intended users. This result substantiates the claim that SoViz is a valuable design method for visualizations.

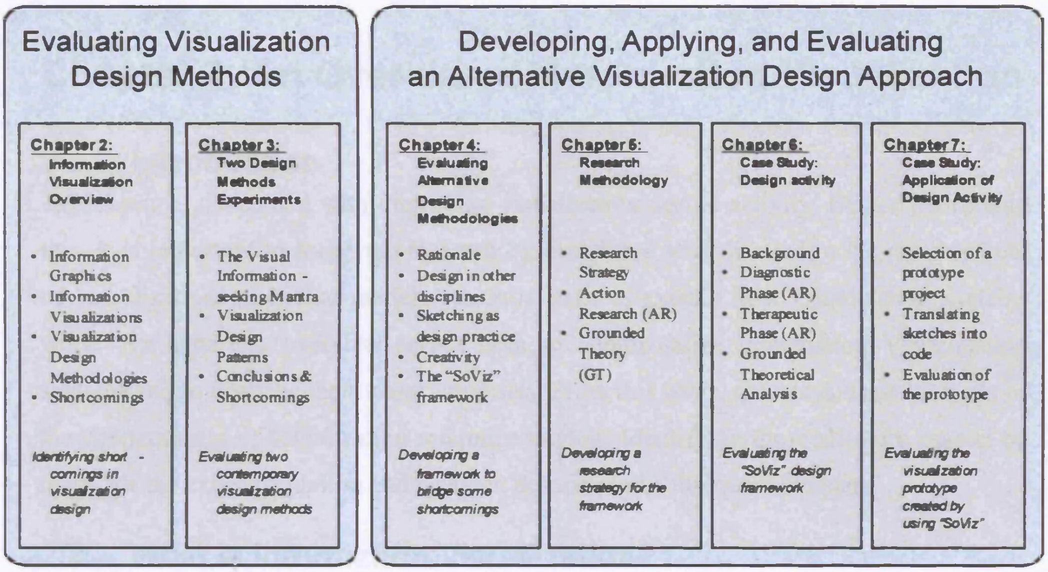


Fig. 1.1: Thesis structure, showing the topics discussed in each chapter and the contribution of each (italics).

Chapter 8 concludes the thesis. A summary is presented, highlighting relevant key claims and contributions to Information Visualization design, and areas for further research are suggested.

Chapter 2: An Overview of Information Visualization

2.1. Introduction

This thesis is concerned with improving visualization design activity. Before addressing this, it is important to frame the research by describing what has led to the development of visualizations and what models currently exist to explain how visualization systems work. An historical overview serves both to contextualize Information Visualization systems and to describe their major attributes. From this basis, an analysis can be made of the shortcomings of visualization reference models. Identifying these allows a case to be made for the experimentation and research described in subsequent chapters.

2.2. What is Information Visualization?

Visualizations of any kind are simply external representations of mental ideas and concepts. This broad description can encompass any manner of visual representation, from a painting to a pie chart. With the advent of inexpensive, high-performance computing, powerful software systems have been developed to create visual representations of very complicated data and entities. Information visualization systems are a sub-class of software visualization systems, which are designed to represent and manipulate abstract information for the purpose of gaining a better understanding about it. They draw upon innate faculties of human visual processing and use novel, abstract representations of data to enhance cognition and communication.

Such systems have made it possible for people to work with larger and more complicated sets of information. But even as the sizes of data sets increase, people must be able both to understand the nature of such information and to draw conclusions about it. That is to say, good Information Visualization systems must help people to think about data. To understand how information visualization systems do this and the challenges associated with their design, it is important to review the roots of information representation and the cognitive benefits of external, visual representations.

2.3. Background

Humans have been making abstract, visual representations since the development of written languages several thousand years ago. Letters, ledgers, mathematical calculations, and many other activities are all facilitated by making various kinds of marks as visual abstractions of ideas. These have served as vectors of immediate or delayed communication, as reminders or records, even as tools for thinking about problems (Card, et al., 1999). This last example is perhaps where representations have proved most useful, because they enhance the human ability to solve problems.

A prosaic example of this is simple, written arithmetic. It is a trivial task to add two numbers together mentally, say 23 and 15. However, for more complicated additions, the mental load becomes burdensome. Imagine adding the numbers 2,359 and 8,954. Though it is possible to do this without resorting to pencil and paper, most people find it easier to use written notation (and some simple rules for manipulating the notation) to make this calculation manageable. Written representations can thus be seen as a tool for supporting cognition because they transfer the mental load of a task into the real world. Marks on a page can serve as shorthand and allow a complicated problem to be decomposed into smaller, more manageable chunks.

One of the reasons that written arithmetic aids cognition is that it uses the physical positions of numbers in the decimal counting system as signifiers of meaning. Marks in the first columnar position can represent values from zero to nine. Marks in the second column represent tens. Marks in the third column represent hundreds, and so on. The physical positions of the marks, in addition to the marks themselves, act as signifiers of meaning and thereby aid cognition and computation. The ability to perceive the physical position of the marks is a basic human faculty resulting from the physiological and cognitive components of the organs involved in visual perception, which we can refer to simply as the visual processing system. This system has remarkable attributes, all of which work in unison to allow us to orientate ourselves in the world. The ability to perceive separate objects, their positions in space, their lightness, brightness, contrast, texture, colour, contour, movements, and other characteristics is a physiological capacity which we can use not only to locate ourselves, but also to solve problems (Ware, 2000). As in the example above, in combination with symbolic representation, they also can be used to enhance our cognition about even those abstract problems which have no physical manifestation. A comprehensive cognitive theory of human problem-solving in which such symbolic systems are fundamental is proposed by Newell and Simon (1972).

The visual attributes of marks have long been used to enhance cognition but took an important turn in the 18th century with the work of Scottish engineer and political economist, William Playfair (Tufte, 2001). He argued that abstract, visual representations of data are more effective than simple listings of numbers in tables. He created some of the earliest examples of charts which used visual characteristics, such as shape, colour, and contour as abstract representations. An example is the chart in Figure 2.1, which

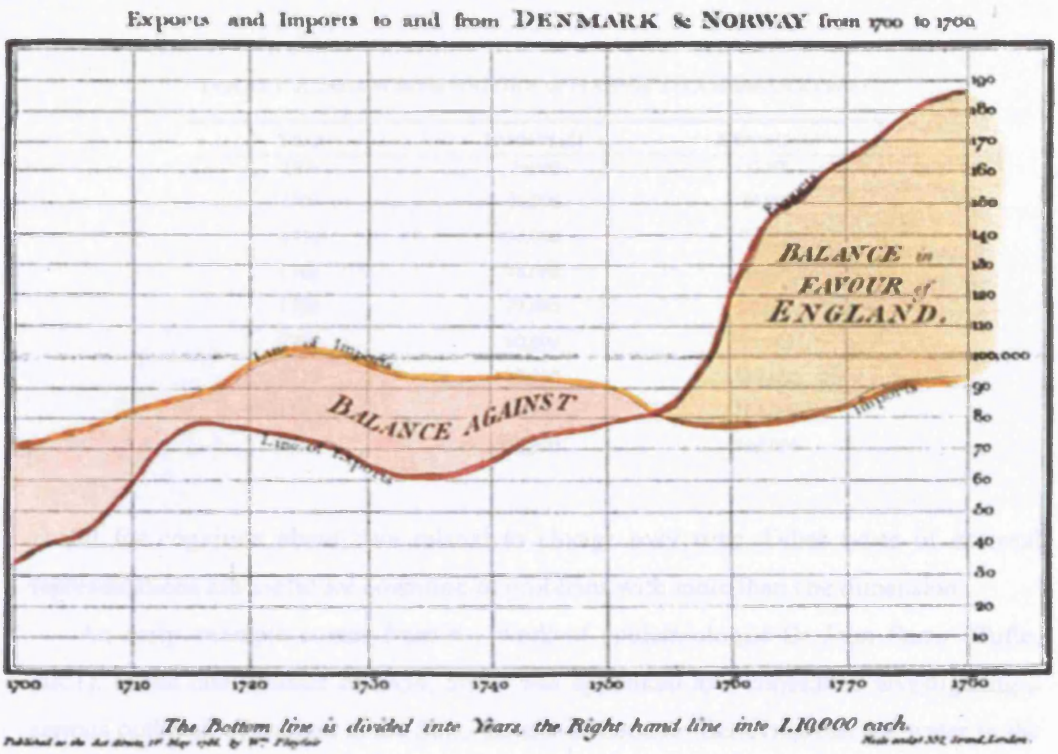


Fig. 2.1: Playfair's chart representing exports form to Denmark and Norway (Tufte, 2001).

shows the balance of trade between England, Denmark, and Norway, over a period of 90 years. It uses a visual abstraction of the number of Pounds Sterling and Years to demarcate a visual area. This abstraction permits a comparison of the two resultant areas, which are immediately visible because the visual processing system is quite good at immediately identifying regions of shape and colour. Also, the change in direction of the lines over time creates an obvious contour. Compare this chart to a table (Table 2.1) which represents the same information and it becomes obvious that the former has advantages over the latter. Not only do the visual attributes of the chart make the change in trade balance immediately obvious, the chart has the benefit that its lines represent essentially continuous data very efficiently. Although the table could be presented with a finer granularity of time, greater accuracy would increase both the number of figures in the table and the difficulty in perceiving the trade balance. Since the cognitive task in question is understanding the trade balance, it is reasonable to argue that the chart is more effective than the table.

This example – and the many others in Playfair's work – shows how the innate abilities of human visual perception can be used to improve sense-making of large amounts of information very quickly and easily. In this case, the visual representation was

TABLE 2.1: A TABULAR REPRESENTATION OF PLAYFAIR'S TRADE BALANCE CHART

YEAR	IMPORTS (£)	EXPORTS (£)
1700	70,000	35,000
1720	81,000	58,000
1730	95,000	77,000
1740	98,000	65,000
1750	93,000	66,000
1760	90,000	78,000
1770	79,000	117,000
1780	84,000	164,000
1790	92,000	185,000

useful for cognition about data related to change over time. Other types of external representations are useful for cognition of problems with more than one dimension.

An early example comes from the work of epidemiologist Dr John Snow (Tufte, 2001). In the late summer of 1854, Snow was appointed to a committee investigating a serious outbreak of cholera in the Soho district of London. Believing that the water in the area was the source of the disease, he spoke to residents door-to-door and determined that most of those who had died had drunk water from a pump in Broad Street. He succeeded in convincing the committee to have the suspected pump disabled and soon afterwards, the epidemic subsided. Later, he used a novel visual representation to illustrate the problem.

In this case, geography plays an important role in understanding and communicating the data. Though he could have created a chart reporting the daily death toll in a time-sequential fashion similar to Playfair's chart, Snow had suspected that proximity to the Broad Street pump played an important role in the cholera deaths. Both proximity and number of deaths were important dimensions of the data. In published reports communicating his findings, Snow used a dot map to plot the deaths in the area. A revised version of this map is in Figure 2.2. Points plotted on the map correspond to the locations where cholera victims had died. Crosses (×) are plotted at the locations of the area's 13 water pumps. The proximity of the dots to the water pumps shows that most of the deaths centred on the suspected pump, which is denoted by a cross to the right of the 'D' in Broad Street. Because discrimination of proximity and area are innate visual faculties (Hoffman, 1998), the relationship of the cholera deaths to the suspect pump are made evident. As with the example from Playfair's work, it is easy to see that a table containing the same data would not tell the same story. Notably, and contrary to popular lore, Snow did not use this map to deduce that the deaths were caused by water from the

Broad Street pump but rather, used it to communicate clearly that deaths were occurring in a particular area, a fact which he used to support his theories of water-borne disease communicability (Brody, et al., 2000). However, it is easy to conclude that such a map could be used for deductive reasoning, and indeed, Snow's contemporary, Edmund Cooper, used similar plots to deduce and demonstrate that foul air from the area's sewer ventilation shafts was not a cause of cholera deaths (ibid.).

These early visual abstractions mark the beginnings of the use of graphical representations to communicate data information, a domain which is now referred to as *information graphics* or *data graphics*. Notable early examples exist from the work of Minard, Nightingale and many others, all of which use visual abstractions as data representations (Spence, 2001). The commonality among them is that they rely upon visual processing capacity to enhance cognition and communication of data and they do this by mapping data to visual attributes. Much of the early work appears to have been based on the clever intuition of the people who created the visual representations. There is

Fig. 2.2: An adaptation of Snow's map of cholera deaths in Soho (Tufte, 2001).

little evidence that the designers involved had explicitly explored why these abstractions were effective at cognitive level. Also, nobody had catalogued which marks were most effective at communicating different types of information. One of the first to explore this was Jacques Bertin.

Bertin, a French cartographer, made a systematic inventory of the types of visual marks which were available to the cartographer for representing data, primarily to indicate the results of statistical and demographic information. Further, a method for selecting these was presented. This work was published as *Sémiologie Graphique* (1967) and was based upon Bertin's extensive experience with graphics and maps to encode data into visual form; although this draws upon psychophysical principles of visual perception, it was not based upon laboratory research and contains no references. Yet, Bertin's work is important for information visualization in two ways. His taxonomy of visual marks which can represent data attributes is significant because it was the first exhaustive exploration of the subject. He also showed how data presentation can be rearranged to reveal meaning which might not be immediately apparent. Information Visualization makes use of both of these methods to enhance cognition and communication of data.

Bertin referred to the elements of his taxonomy as 'retinal variables' or 'visual variables'. They are presented in Figure 2.3. The elements are 'visual' because the visual attributes of marks can be used to signify meaning and draw upon *visual gestalts*. Visual gestalts are the visual characteristics of objects which are immediately perceived as part of the visual processing system and which do not rely upon any interpretation. These are listed on the left of the figure. For example, size is immediately perceptible without having to draw any conclusions. It is not necessary to think about whether one circle is larger than another. Barring the effects of optical illusions, which Bertin's work does not account for, it is simply apparent as a result of the way human vision works (Bertin, 1967). Similarly, value, texture, colour, orientation, and shape operate at a perceptual level.

The elements are also 'variables', in that they can be associated with different types of data to convey meaning. These meanings are: association, selection, order, and quantity. Bertin's argument is that these can be used to represent data in specific ways on a two-dimensional plane and that these meanings are immediately perceptible, without the need to refer to a key or legend. For example, on a two-dimensional map, the size of a circle can be associated with the population *quantity* of cities. Differences in populations can be thus made immediately apparent. Similarly, all the regions of a country which

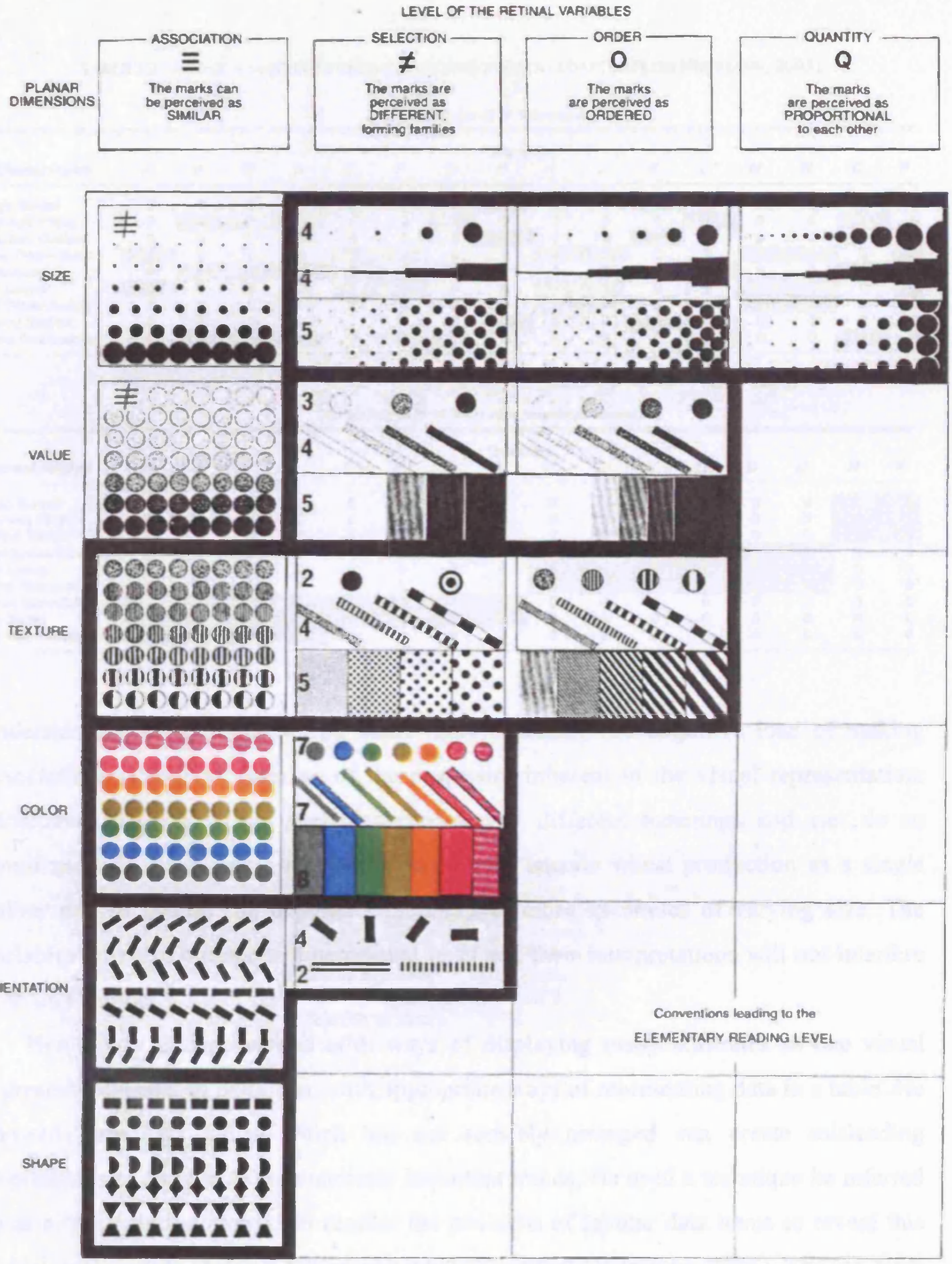


Fig. 2.3: Bertin's (1967) taxonomy of visual marks which can represent data.

produce wheat can be encoded with the same colour, which gives an immediate indication that they are *associated.* One may need to refer to supplementary text to understand that the coloured encoding indicates wheat production but it should go without saying that the different regions which are similarly encoded are related in some way. Bertin's argument is that by using visual perceptual principles to encode data, the

TABLE 2.2: USE OF A MATRIX TO REVEAL PATTERNS IN DATA (ADAPTED FROM NIERMANN, 2005).

Characteristics of 16 Townships

Characteristics	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Townships* *H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*
High School	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Agricult. Coop.	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0
Railway Station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
One-Room-School	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
Veterinary	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0
No Doctor	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
No Water Supply	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Police Station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Land Reallocation	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0

Characteristics in 16 Townships (reordered version)

Characteristics	*N*	*J*	*M*	*I*	*P*	*F*	*E*	*Townships* *A*	*B*	*O*	*L*	*G*	*D*	*C*	*H*	*K*
High School	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Railway Station	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Police Station	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Agricultural Coop.	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
Veterinary	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
Land Reallocation	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
One-Room-School	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
No Doctor	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
No Water Supply	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

understanding of the data can be made more efficient. The cognitive load of making associations is reduced because of the properties inherent in the visual representation. Moreover, different visual variables can convey different meanings and can do so simultaneously. In a single map, it is possible to encode wheat production as a single colour and to encode the population *quantity* of cities as circles of varying size. The variables will still operate at a perceptual level and their interpretations will not interfere with one another.

Bertin was also concerned with ways of displaying many attributes in one visual representation and, in particular, with appropriate ways of representing data in a table. He showed how data tables which are not sensibly arranged can create misleading impressions or can fail to communicate important trends. He used a technique he referred to as a 'permutation matrix' to reorder the positions of tabular data items to reveal this hidden information. Table 2.2 demonstrates this concept (Niermann, 2005). The top table shows a list of 16 different townships and attributes which they may or may not have. This binary state is indicated by a "1" or a "0" and grey or white shading. It is easy to determine that township A does not have a High School. However, a change in the presentation of the rows in this table leads to further conclusions about the data set as a whole.

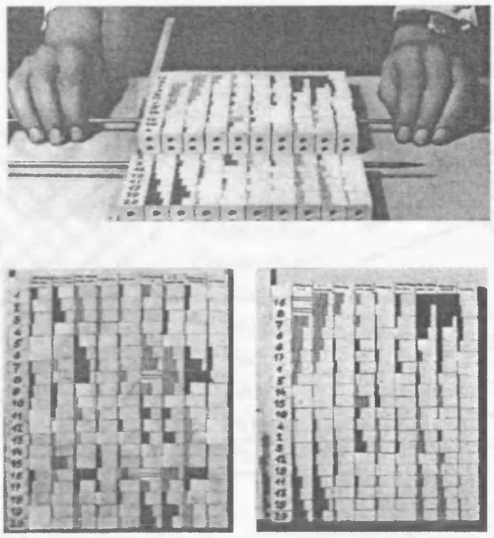
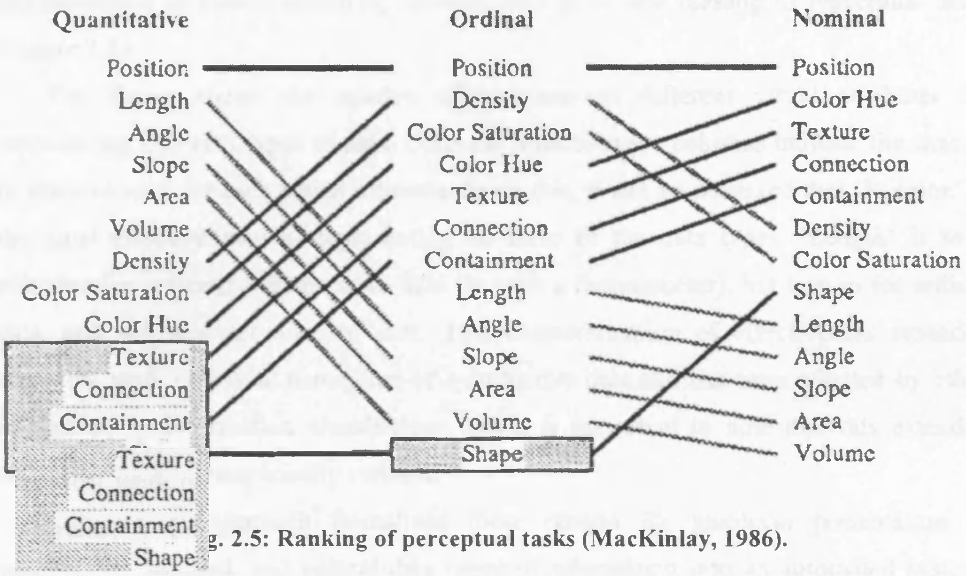


Fig. 2.4: Dominos for physically diagonalizing a data matrix.

It is possible to rearrange the nominal data in column 1 to reveal interesting patterns. The bottom table shows that townships with one-room schools are not likely to have a water supply or a doctor. Also, all towns with agricultural co-operatives have veterinary services. Though it is possible to determine these facts from the top table, the patterns of association among these townships are not obvious at a perceptual level. Reordering the matrix reveals information at a perceptual level, by the visual variables of *proximity* and *value*.

Bertin explored the possibilities of this approach and developed a method of 'diagonalization' for reordering tables to reveal inherent patterns. His research group built tools (Figure 2.4) to physically perform matrix permutations on larger sets of data, but limitations begin to crop up with large data sets. Bertin (1981) observed that as the magnitude of the data matrices increases above 120 x 120 cells, it becomes more difficult to perform these permutations, though experimental equipment for matrices of 500 x 100 cells was devised. Bertin recognized that computers would be more useful for extending these limits, as they would vastly increase the speed and efficiency of making such manipulations; an early system to achieve this called AMADO (Analyse MAtricielle des DOnnées) was developed to explore this (Personal communication, 2006).

MacKinlay drew upon Bertin's work for the purpose of automating the graphical presentation of relational data with software. He proposed that graphical presentations of



g. 2.5: Ranking of perceptual tasks (MacKinlay, 1986).

information are very similar to formal languages in that their syntax and semantics can be precisely described (Mackinlay, 1986). Such a formal language can be implemented in a computer system. Extending this analogy, the elements of a graphical language can be constructed into graphical sentences. On this basis, criteria can be determined for both the *expressiveness* and *effectiveness* that a graphical language can provide. For MacKinlay, expressiveness refers to the ability of graphical sentences to present exactly and only with the desired information. Effectiveness can be evaluated by many criteria, but he was primarily concerned with efficiency of interpretation. The faster the meaning of a graphic can be understood, the better.

He characterized graphical expressiveness as a formal structure which encodes the syntax of a graphical language. But while expressiveness can be precisely defined, effectiveness is dependent upon the observer. To supply these criteria, MacKinlay drew upon the work of Cleveland and McGill (1984), which showed that 'people accomplish the perceptual tasks associated with the interpretation of graphical presentations with different degrees of accuracy' and that accuracy is related to visual attributes. For example, it is possible to map quantitative data to a linear scale, using the position of a marker on the scale to indicate a quantitative data value. A thermometer is an ordinary example of this. However, mapping temperature data to levels of grey would be less effective, because our perceptual acuity for levels of grey is not as precise as that for position. MacKinlay extended the work of Cleveland and McGill to include the representation of non-quantitative, nominal data, in a new ranking of perceptual tasks (Figure 2.5).

This figure shows the relative effectiveness of different visual attributes for representing different types of data. Diagonal lines between columns indicate the change in effectiveness for each visual attribute. From this, it can be observed that 'Position' is the most effective way of representing all three of the data types. 'Length' is very effective for indicating quantitative data (as with a thermometer), but less so for ordinal data, and still less for nominal data. This characterization of effectiveness extended empirical work on visual perception of quantitative data and has been adopted by other researchers in information visualization, but it is important to note that this extended model has not been empirically verified.

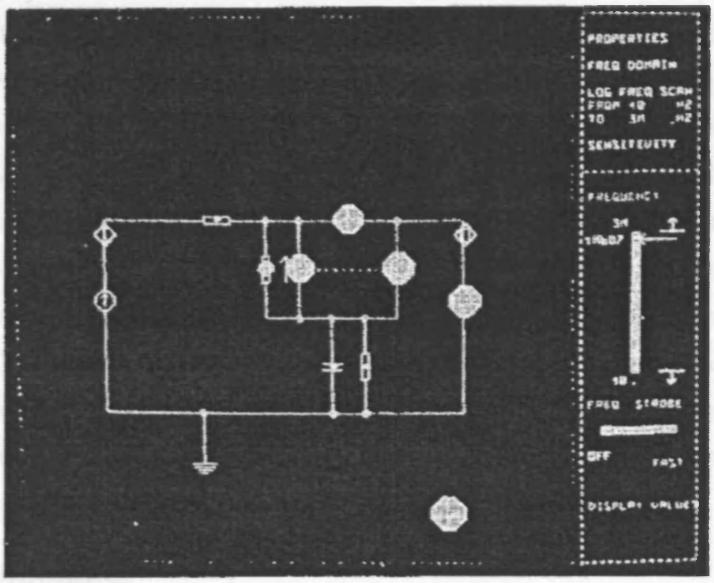
MacKinlay's approach formalized these criteria for graphical presentation of quantitative, nominal, and ordinal data types of information into an automated system, APT (A presentation tool). The system determined the best graphical presentation for tables of data, based upon expressiveness and effectiveness criteria. This approach brought the cognitive benefits of external representations to the domain of automated computation and computer-generated visual presentation. Although it was not the first system to do this, it was the first to acknowledge that selection and combination of visual elements could be approached systematically and that formal descriptions of graphical language could have relevance to computer visualization.

Others researchers approached the problem of computer-based data presentation from a different perspective. Spence and Apperley (1977) had also proposed a system which would map data attributes to visual elements on a computer display, but with an emphasis upon user interaction. Their work acknowledged the importance of appropriate visual representations as described by Bertin, as well as the importance of transforming data representations. They combined these into a single system. Crucially in terms of Information Visualization, their approach was to allow this transformation to be manipulated in real time by the human agent involved in perceiving the data representation.

Spence and Apperley were working in the domain of electrical engineering, in which the process of analogue circuit design involves frequent testing of alternatives, depending on often ill-defined design requirements. At the time, models of circuit behaviour could be calculated automatically by computers and the output of such models would be generated as numerical tables. But the testing of alternatives was difficult because of the

0.30	0.44	0.15	0.62	0.52	0.52	0.77	0.35	0.51	0.61
0.17	0.27	0.18	0.81	0.32	0.81	0.63	0.32	0.82	0.21
0.32	0.32	0.95	0.55	0.42	0.41	0.92	0.77	0.63	0.92
0.36	0.15	0.89	0.75	0.67	0.44	0.35	0.36	0.61	0.99
0.56	0.39	0.35	0.89	0.35	0.32	0.74	0.62	0.81	0.91
0.39	0.17	0.25	0.45	0.33	0.61	0.66	0.22	0.11	0.15
0.47	0.29	0.85	0.44	0.81	0.82	0.13	0.17	0.63	0.21
0.45	0.48	0.25	0.27	0.19	0.98	0.91	0.31	0.37	0.45

(a)



(b)

Fig. 2.6: The MINNIE system. (Spence and Apperly, 1977).

time delay between receiving the tabular results of a proposed design and inputting new, alternative values for circuit components. The need to input these new values and then to wait for a response from the computer inhibited efficient exploration of design alternatives. Also, the tabular output did not match the visual external representations which electrical engineers had developed over the previous century to aid their cognition of circuit design problems. To overcome these limitations, Spence and Apperley designed a system (MINNIE) which integrated the output of computer models and the visual representations which electrical engineers were used to. Moreover, the system allowed an engineer to test design alternatives interactively, by manipulating the data inputs on the computer screen using a light pen, in near-real time. This interactivity decreased the delay in the feedback loop between the engineer and the computer, with several effects. One result was that more design alternatives could be explored in less time. Another perhaps more important outcome was that the effects of making changes to one part of a circuit design were made visible in other parts of the circuit. Figure 2.6 demonstrates this. The table (a) at the top of the figure represents possible data values in the components below. During exploration of alternative circuit designs, a change to a single data value input might have a result on all of the values in this table. However, it would be difficult to grasp the effects without laboriously examining the values in each of the cells. Mapping the data values to familiar visual representations within the user interface (b) allows the use of visual variables to indicate change (in this case, diameter of the circles) with the effect that an engineer can see the results of a proposed change in all parts of the circuit design very quickly. Also, minor changes are made more evident because the human visual processing system is able to discern minor changes in area quite easily. Such a small change might be overlooked if it were represented in tabular form. Thus, it can be said that the externalization of the data using visual attributes enhanced cognition. But the important contribution that MINNIE made was for direct, human interaction. This supports the tasks of understanding data, exploring data and using data for problem solving. All contemporary Information Visualization systems share these attributes. Before describing how designers create novel visualizations and identifying areas for improving this process, it is necessary to discuss contemporary definitions which attempt to codify the attributes of visualizations and the tasks they support.

2.4. Definitions

Contemporary conceptualizations of Information Visualization posit that human interaction with visual abstractions of information can enhance cognition. The examples above demonstrate how information can be presented in a manner that makes it easier to understand and use, whether in print, or as a computer software system for Information Visualization. Card, et al., (1999) have offered a definition of the domain of Information Visualization. Their theoretical work has two key contributions: they describe the boundaries of the domain, and they describe a reference framework for Information Visualization systems. Their reference framework is described later, in section 2.5. For now, we focus on their definition.

The boundaries of the domain are described by Card, et al., situating Information Visualization within the body of external cognition tools and distinguishing it from other types of visualizations. They propose a hierarchy of visual representations (Table 2.3). External cognition is the overarching idea which frames this hierarchy and is described by Card, et al., as the 'role of the external world in thought and reasoning'. The arithmetical example presented earlier is a very simple illustration of what is meant by using a tool in the external world to enhance cognition. For Information Visualization, the amplification

TABLE 2.3: VISUALIZATION DEFINITIONS (CARD, ET AL., 1999). (ITALICS IN ORIGINAL.)

CONCEPT	DESCRIPTION
EXTERNAL COGNITION	Use of the *external world* to accomplish cognition.
INFORMATION DESIGN	Design of *external representations* to amplify cognition.
DATA GRAPHICS	Use of *abstract, non-representational* visual representations of data to amplify cognition.
VISUALIZATION	Use of *computer based, interactive* visual representations of data to amplify cognition.
SCIENTIFIC	Use of interactive visual representations of *scientific data*, typically *physically based*, to amplify cognition.
INFORMATION	Use of interactive visual representations of *abstract, non-physically-based data*, to amplify cognition.

of cognition consists of six factors. Increasing memory and processing resources are achieved by making data items visible in the display so that the user does not need to keep track of items mentally. Because data items are visible, important data can be visually enhanced, thereby reducing the search. The visibility of items makes it possible to aggregate similar items with similar visual attributes, enhancing the recognition of patterns. Perceptual inferences can be supported by making obvious any problems arising in the data. In the example above, users of MINNIE were able to perceptually determine when values for electronic components would be inappropriate in the circuit. Similarly, a large number of items or events can be perceptually monitored simultaneously if the visualization highlights only the important components within the display. Finally, visualizations amplify cognition by making data manipulable. This permits users to explore the data interactively, which is not possible with static diagrams. Interaction is important because it allows for knowledge crystallization, which will be described below.

Information design consists broadly of the external, graphical presentation of information to amplify cognition. This encompasses any tool which visually abstracts data. Card, et al., show how external visual aids such as, nomographs, navigation charts, and explanatory diagrams can also be useful. A more thorough exploration of this category is presented by Tufte (2001). Data Graphics are characterised as visual representations of information which map the information to visual abstractions, but not in the context of a computing system. The work of Playfair and Snow are data graphics. They map data to abstract visual representations but have no interactive capabilities.

Visualizations, in general, are described as interactive computer systems which present data graphics. Information Visualization systems are one class of these. They are computer-based, an attribute which separates Information Visualization from other types of external cognitive aids, such as charts and graphs. Because of this, human interaction with a system can change the visual representation of data, enhancing cognition. These are the two defining characteristics of visualization systems, which lead to enhanced cognition.

Information Visualization requires human interaction because interaction can change the representation of data. This allows users to explore the space of many parameter values in the context of problem-solving and is not possible with data graphics. The MINNIE system provides a good example. The function of interaction is that it aids knowledge crystallization.

This is a defining characteristic of Information Visualization. Card, et al. describe this as a way of 'getting insight about data relative to some task'. Spence refers to this as the 'acquisition of insight'. Shneiderman (1996) has described it as the 'A-ha! Moment' when the user attains an understanding about the data as a direct result of exploring and interacting with a visualization system. Knowledge crystallization is offered as an explanation of what occurs when a user interacts with data to achieve a specific goal. Drawing upon the work of Russell at al. (1993), the activity is conceived by Card et al. as an iterative process which consists of information foraging, searching for a mental schema to represent the information, instantiating the mental schema, problem-solving using this instantiation, making decisions on how to act based on the results, then using this knowledge to perform a task. The example provided is the problem of purchasing a laptop computer by collecting information about laptop attributes, determining a schema for making comparisons, adding data values to this schema, evaluating the options, then deciding on and executing a purchase. The benefit that Information Visualization tools provide is that they make this process more efficient, primarily by collecting data in one place and highlighting patterns within it. An ideal visualization tool for this task would make available all of the relevant data in a system with a visual abstraction of laptop attributes which would facilitate identification of opportunities and making comparisons. It would allow interactive explorations of these attributes so that the user could make a purchasing decision.

A second defining characteristic is *visual representation.* At least in terms of design methodology, the definition in Table 2.3 is somewhat problematic. The reason for this is that the boundary between scientific visualizations and Information Visualizations is vague and hinges upon the notion of whether the data is about an abstract or a real-world entity. This distinction is important for Card, et al., because it is used to differentiate scientific visualization from Information Visualization. Information Visualization is said to involve representations of abstract data which do not inherently manifest as physical objects in the world. By contrast, a scientific visualization system might represent, for example, the fluid dynamics of airflow over a wing by using a visual depiction of an aircraft wing and coloured mappings of the data which represent the airflow vortices that move across the wing. But in fact, all software visualization systems display a visual representation of a dataset drawn from a data subject. Whether the data subject is a trade deficit or an airfoil, the semantic content of the data subject is irrelevant from a systems point of view. Visualization systems must access a raw data file, a mathematical model, a relational table, a database, or another kind of data structure which holds data about the data subject. Tory and Möller (2004) agree that creating a distinction between the two domains is problematic, noting that, 'The various definitions of scientific and information visualization often contradict each other or contradict experts' intuition about what belongs in each category'. Because the dataset which the visualization system uses is not the crux of this definition, it must lie elsewhere. Most likely, it lies in the visual abstraction of the data.

What is confused in the definition from Table 2.3 is that it is not the *data subject* which is abstract, but the *visual representation* which is abstract. Indeed, much of the Information Visualization literature is written from a perspective in which this is the logical assumption. If this is the case, we must draw at least one important conclusion. Except for categorizing systems based on the type of data subject that they represent, the definition holds little other value from the perspective of visualization design. The point of this is not to split hairs about scientific versus Information Visualization nor the accuracy of the definition above, but rather that the creation of a visual abstraction is a key visualization challenge, from a design methodology standpoint. Notwithstanding the identification of effective visual mappings as with Bertin, and Cleveland and McGill, nor the automation of those mappings, as with the work of MacKinlay and others, the generation of the visual representation, *abstract or otherwise*, is poorly represented in visualization design methodologies.

Additionally, user interaction is similarly poorly represented. Of course, a visual abstraction of airflow such as an interactive histogram can facilitate more useful comparisons about variables than displaying those variables as animated, coloured streams travelling over a wing. Consider, however, that the user task has nothing to do with making comparisons about the variables in the airflow over a wing or the national debt, and rather, is concerned with understanding change-over-time in such systems. In this case, it is perhaps more useful to see that change-over-time represented as an animation. Further, suppose the user needs to identify optimum parameters as the wing ages. It may be important for the user to interact by changing the parameters that are displayed to highlight areas of interest (joints, rivets), to reduce visual clutter (hide the surface skin parameters), or to match an existing mental schema (this is a particular type of wing). To aid memory, perhaps the length of time displayed might be reduced. From this example, it is clear that whether this animation is a visual abstraction of the data or a picture of the data subject, the user task is highly implicated in the knowledge crystallisation. Therefore, user representation should be an aim of the visualization design process. The coordination of visual representation with a user interaction that is task-specific is challenging indeed. Section 2.5 will show how different authors have tried to structure these requirements as reference models.

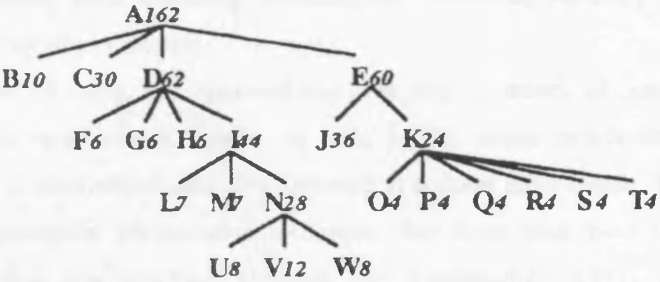
These definitions raise questions about how designers should approach the problem of creating a system with effective visual representations, and which supports knowledge crystallization through user interaction. How can a designer create a system that will aid information foraging? How will a designer know that a proposed visualization will instantiate the user's mental schema or support decision-making? In a more general sense, a question arises as to whether there is a principle method which will lead to successful, innovative visualizations?

2.4.1. Problems of innovation in IV

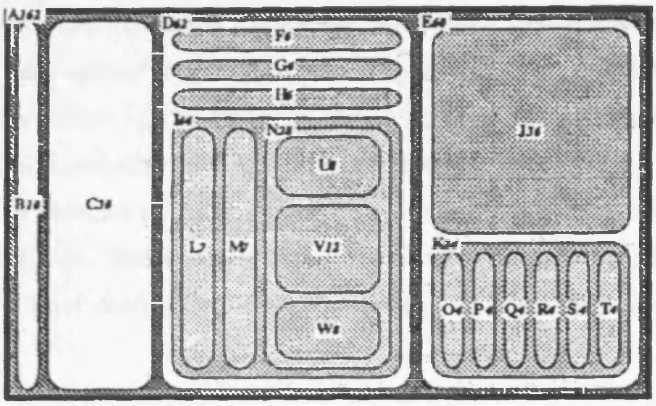
The above definitions of visualization underscore the importance of *visual representation* and *human interaction* as key characteristics of visualizations. Some examples of contemporary visualizations will serve to illustrate these characteristics and will show that it is often unclear how novel ideas were generated. The literature tends to focus on the characteristics themselves and not on the design process. A thorough overview of contemporary Information Visualization systems is provided in Card, et al., (1999), and Bederson and Shneiderman (2003), but discussions of the design process in these references and the literature at large are generally rare.

2.4.2. Example: novel representations

Visualizations use the space onscreen to manage layouts of data. The layouts can be in the form of 1D, 2D, 3D or n-D data representations, which must all be constrained to the two-dimensional space of the screen. The treemap (Johnson and Shneiderman, 1991) is a technique for efficiently representing node-and-link hierarchies which takes advantage of



(a)



(b)

Fig. 2.7: A traditional node-and-leaf representation of a hierarchy tree (a) and the tree-map visualization (b).

the maximum possible two-dimensional screen space. The rationale for this abstract representation is that traditional node-link diagrams are not space efficient. Figure 2.7 shows a traditional (a) and a treemap (b) representation of a node-and-link hierarchy. Nodes in (a) are represented as boxes in the treemap (b). Links in (a) are represented as containers in (b). In the treemap, the box labelled K24 (lower right) contains the child nodes O24, P24, Q24, R24, S24, and T24. The entire two-dimensional area of a screen can be employed to represent the hierarchy. As the number of nodes increases, the wasted screen space in the traditional diagram would soon make representation impossible. The representational threshold of a treemap is substantially higher. A typical display offering well over a million pixels of display space would be able to display several thousand nodes whilst maintaining visibility of and ability to select the smallest items. Advances in treemap designs have used zooming and alternative layouts to increase this display capacity. User interactions with treemaps entails selecting individual nodes, or collections of nodes to display data, searching, selecting and displaying variables of interest, and zooming in on regions of interest.

The results of using this approach are that large numbers of data items can be displayed in a very compact space; as long as the visual presentation entails the representation of hierarchical data, this approach is domain independent. Tree-maps have been a very successful visualization technique, they have been used in industry and academia to show file structures (Johnson and Shneiderman, 1991), news[1], product attributes[2], stocks and securities (Csallner, et al., 2003), biological data (McConnell, 2002), photo collections (Bederson, 2001), and many other data domains. But although tree-maps have been applied to a wide variety of applications, the literature usually does not provide a clear picture of why they were chosen for a particular domain.

In terms of understanding how novel representations are created for visualizations, one might turn to accounts of the design process itself. Yet such accounts are not often given in the literature. Shneiderman offers a rare glimpse into visualization design creativity with a brief description of how he arrived at a tree-map representation for hierarchies:

> *Tree structured node-link diagrams grew too large to be useful, so I explored ways to show a tree in a space-constrained layout. I rejected strategies that left blank spaces or those that dealt with only fixed levels or fixed branching factors. Showing file size by area coding seemed appealing, but various rectangular, triangular, and circular strategies all had problems. Then while puzzling about this in the faculty lounge, I had the Aha! experience of splitting the screen into rectangles in alternating horizontal and vertical directions as you traverse down the levels.* (Shneiderman, 2006b)

Analysing this description reveals some key elements: a problem-solving approach of accepting and rejecting alternatives, decomposition of the problem into smaller chunks, an incubation period and an 'A-ha' moment of inspiration. These attributes of Shneiderman's design process can be interpreted in terms of design and creativity theory.

Stumpf (2001) presents a detailed description of design practice identifying four paradigms of design activity. These paradigms are: Rational Problem-Solving, Social Process, Hypothesis Testing, and Experiential Learning. Shneiderman's description of his design activity most closely fits with the Rational Problem-Solving design paradigm, which views design problems as ill-structured but essentially decomposable into well-defined sub-components and problems. An 'individual information processor' engages

[1] http://www.marumushi.com/apps/newsmap
[2] http://www.hivegroup.com/products.html

with the problem in a design cycle of analyse-generate-test-evaluate. This interpretation is further supported by Shneiderman's own reflection upon his activity:

> *I was doing much more of a problem-solving challenge, more like what Hadamard described about Poincaré... preparation, incubation, illumination, verification. This seems to fit when there is a well defined problem that has a clear solution.* (Personal communication, 2006)

However, visualizations are usually designed as a team effort, as is clear from the preponderance of multi-author papers in the literature. In spite of the fact that one person may have the inspiration for a visual design, the input of many people is necessary to produce a finished result. So, although it may appear that the novel visual representation is the result of an inventor who arrives at a solution through a creative problem-solving process, in terms of visualization software, a more precise description is that there is probably a collaborative and creative effort on the part of many participants. In terms of Stumpf's analysis of design activity, visualization design methodology is probably more appropriately situated as a Social Process of design.

The incubation period and 'A-ha!' moment have been recognized as key phenomena that are often attendant to creative design work, but their precise functions are not understood. Csikszentmihalyi (1996) provides a thorough analysis of creativity, which accounts for the psychology of discovery and invention. The 'A-ha' moment is described as a situation during which the creator experiences an epiphany about the creative problem, often after a period of incubation and reflection. But in his analysis of individuals engaged in creative work, Csikszentmihalyi notes there is little consensus about the mechanisms of the incubation period, or 'idle time' leading up to an epiphany. Psychoanalytic accounts suggest that the subconscious mind takes up the creative work, '...and there, out of reach of the censorship of awareness, the abstract scientific problem has a chance to reveal itself for what it is—an attempt to come to terms with a very personal conflict'. Cognitive accounts offer a similar explanation which suggests no directionality of subconscious thought. Because rationality does not censor thought associations in the subconscious, new ideas become possible. It is also proposed that because intentionality does not work subconsciously, many ideas are free to form associations which may not otherwise be possible, in a process akin to 'parallel processing'. These conflicting interpretations shed little light on the benefits of incubation and offer little explanation of the role of creativity in design activities. Although Shneiderman's account indicates that incubation and epiphany were significant, it is unclear how these lead to successful visualizations or whether they are necessary.

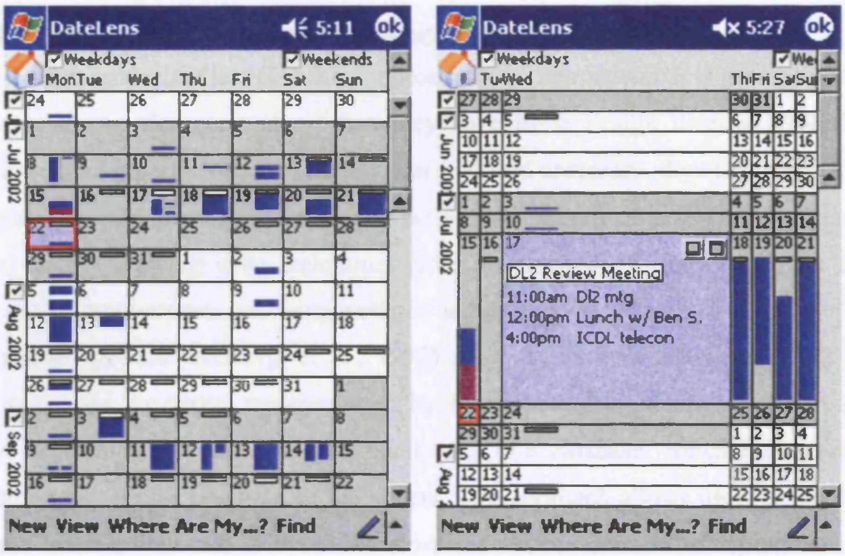


Fig. 2.8: Use of a fish-eye distortion technique to present calendar information (Bederson, et al., 2003).

Another novel representation which is often referred to in the Information Visualization literature is the technique of 'Fisheye Distortion'. This approach applies a sort of magnifying lens to a collection of data items, so that important items in the centre of attention are magnified whilst items of lesser interest are made smaller. This, like tree-maps, has the effect of enlarging the usable display space. An example in Figure 2.8 shows this approach applied to the constrained space of a calendar application on a portable digital assistant. A date such as 'July 17' can be interactively magnified to reveal appointments or reduced to emphasize the events planned for the month of July. This technique has been applied to a large number of data domains, including hierarchies (Lamping, et al., 1995), menus (Bederson, 2000), calendars (Bederson, et al., 2003), and even web browsers (Baudisch, et al., 2004). Furnas (1981) does not describe what led him to use fisheye distortion applied to program code. The later explanation (1986) is that his team were researching people's mental representations of large data structures. (Furnas, 1981, in Card, et al., 1999). But like Shneiderman's account, this provides little insight into the activity of the visualization design process, nor does it inform a methodology which can account for creativity. This sparse description is typical of accounts of design in the visualization literature, where the emphasis is on techniques and technology, rather than the generative process. Given these sparse descriptions, it is difficult to know how future designers are supposed to create novel representations or even whether these kinds of representations will be useful for solving unique visualization problems.

2.4.3. Examples: novel interactions

The picture is somewhat less difficult for determining interactions. It is generally the case that visualization designers report that they have made design decisions based upon intuitions about possible tasks which the user will find necessary, drawing upon expertise from past experiences. Some designers use a more empirical approach to interaction design, largely drawing upon techniques from the domain of HCI. These are usually composed of requirements gathering methods and evaluation techniques.

Dynamic queries (Ahlberg, et al., 1992) is an interaction technique which allows users to change the visual representation by manipulating graphical widgets, typically sliders. Adjustments to these widgets elicit calls to a database, which returns relevant data, dependant on the positions of the sliders. This approach allows users to act directly upon the visualization and to avoid the slow and tedious process of typing a database query data into fields and waiting for results. The results are interactively displayed, animated, in real time. Feedback is instantaneous, and the interaction allows the user to explore the ranges of parameters in the dataset. The technical details of formulating a valid query are dealt with behind the scenes. The user does not need to know anything about them.

There are many examples of the dynamic query technique which represent data in a variety of visual representations and data subject domains, including real estate (Williamson and Shneiderman, 1992), film (Ahlberg, et al., 1992), the stock market[3], manufacturing tolerances (Tweedie, et al., 1994) and others. The example in Figure 2.9 is the HomeFinder, from the domain of real estate. The sliders on the right side of the display can be used to adjust parameters of interest. For example, suppose a user wants to locate and buy a home. The HomeFinder interface presents a scatter-plot (or 'star-field') visual representation of houses on the market for a particular area. Potential candidates can be selected by filtering the display area based upon criteria of interest. For example, a house costing between $16,000 and $38,000, containing two to four bedrooms, and located within a certain distance from two points can be selected by varying the extents of the sliders. Candidates which meet the criteria are displayed as points on a map and those which do not are removed, in real-time. This allows for easy and rapid selection of appropriate candidates.

As with the design of visual representations, the expert designers of the HomeFinder made assumptions based upon their experiences. Several key interaction concepts

[3] http://finance.google.com

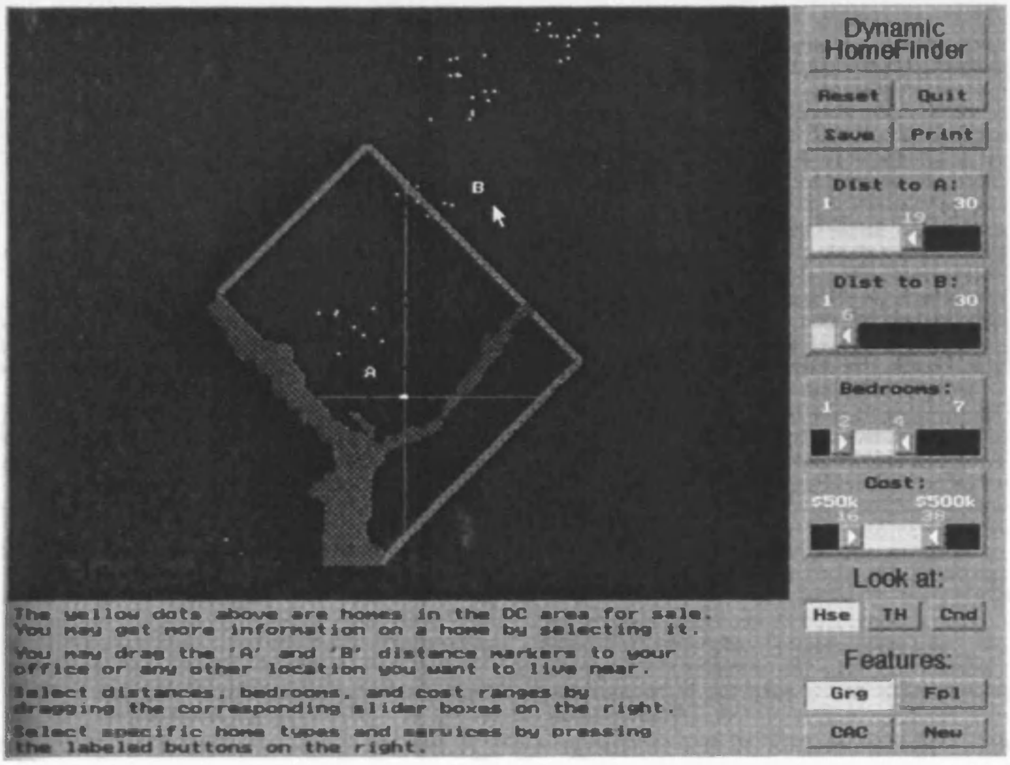


Fig. 2.9: Use of dynamic queries to support user interaction in searching for a home (Williamson and Shneiderman, 1992).

informed their interaction design decisions: rapid, incremental and reversible actions, selection by pointing versus typing (direct manipulation), immediate and continuous feedback. This experience was also used to create a similar dynamic query tool for finding films (the FilmFinder). However, in this case, the designers report that they used informal interviews with shop assistants and film buffs to inform their interaction design decisions. It is possible that the techniques the designers used represent an *ex-post facto* rationalization of a much more haphazard design process, but this cannot be determined. As presented, the description of the design process for these tools also closely matches the Rational Problem-solving design paradigm described by Stumpf.

The influence explorer in Figure 2.10 (Tweedie, et al., 1994) is a tool for selecting light bulbs matching optimum parameters in the manufacturing process. It is based upon mathematical models and pre-calculated data. The output is an abstract view of results based upon the mathematical models. Widgets in the display allow the user to adjust tolerances and performance requirements, which are represented as a scatter-plot in the display window. This technique, called the *prosection matrix*, 'slices through the parameter space' displaying the data values for hundreds of simulations. The user can

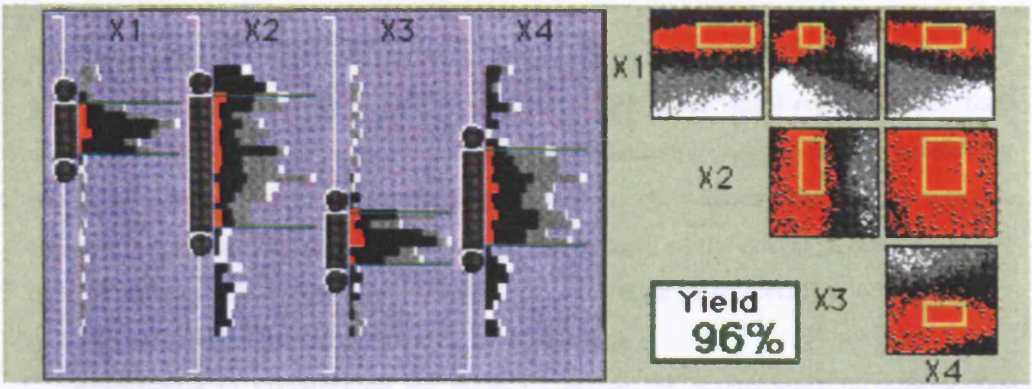


Fig. 2.10: Dynamic interaction with mathematical models to explore manufacturing tolerances (X1 – X4). Optimum yields are displayed within the yellow boxes. (Tweedie, et al., 1994).

vary parameters interactively to display results which highlight optimum manufacturing yields, themselves highlighted within yellow boxes.

The designers created this tool based upon their own intuitions but also reported employing user evaluations to determine refinements in the interaction design. Small, formative evaluation studies during different stages of the design were helpful in making design decisions and determining appropriate interactions. In particular, direct interaction with the user interface was determined to be very important. Users wanted to drag and interact with objects within the display and not with sliders. This approach is useful for evaluating designs in preliminary stages, but it is still the case that significant work must be expended in programming the system before a useful evaluation can be made. Design techniques which could eliminate or reduce this effort would be beneficial.

Though they are reported to be successful design methods, the use of subject-matter expertise and techniques of human-computer interaction do not fully explain how to approach novel visualization design problems; as we shall see, they are not accounted for in the literature.

2.5. Engineering Visualizations

Visualizations help people to understand complex data because they overcome limitations in human reasoning, and support knowledge crystallization. By taking advantage of innate human visual processing acuity, and providing a means of human interaction, they can present large quantities of information in a way that is cognitively manageable. These characteristics have made visualizations useful for researchers in a variety of disciplines with the result that many researchers desire to create visualization systems to support problem-solving in their domains. Because visualization systems allow for complex interaction with large amounts of information, using a novel visual representation, they are difficult to create. There are currently few methodologies that comprehensively describe procedures for creating Information Visualization tools. In particular, techniques for creating interactions and visual representations early in the design process have not been discussed. This leaves people who seek to create new visualizations with few resources to draw upon in order to build new systems.

Researchers in visualization are currently laying the groundwork to remedy this situation, as is shown in design examples (above), Taxonomies, Guidelines, and Reference Models. These areas of research represent attempts to codify visualization knowledge in a meaningful way, and, implicitly, to assist others in the creation of visualization systems. They do this by giving solutions (Examples), categorizing and listing artefacts (Taxonomies), recommending best practices (Guidelines), and describing how visualization systems work, as a whole (Reference Models). Though each offers an increasing level of depth and robustness, none constitutes a methodology. In terms of creating new visualization systems they describe what and when, rather than how.

Examples comprise the collection of Information Visualization systems that have been reported in the research literature. These are typically the systems that demonstrate new technical or algorithmic solutions (e.g. van Wijk and Nuij, 2003), new visual representations (e.g. Johnson and Shneiderman, 1991), and novel interactions (e.g. Ahlberg, et al., 1992). Currently, reports about these systems make up the bulk of knowledge in the domain and reported work comes largely from this area. These examples can serve as a source of design ideas and inspirations for people who seek to design new visualizations, but as they cover very diverse topics, they may not be collected in a useful repository that is pertinent, and may not be relevant to specific design problems.

Taxonomies are attempts to categorize the attributes of visualization systems, applying regular names to visual elements and interactions. They describe ways of classifying information visualization, its characteristics and salient concepts. This body of research involves developing frameworks for organizing and understanding ideas in information visualization or offering categorizations of existing tools. Examples of taxonomies are those proposed by Shneiderman (1996), Chi (2000), and Tory and Möller (2004). Of these, Shneiderman's taxonomy is particularly important because it has been

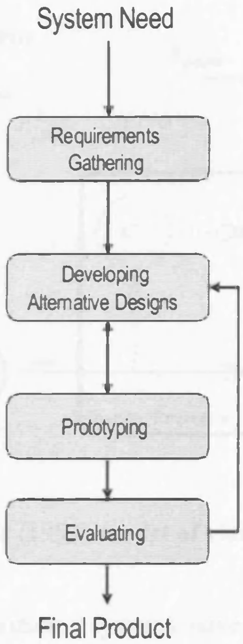


Fig. 2.11: A generic software development lifecycle (adapted from Preece, et al., 1994).

development feed into the next (Dix, et al., 1998). Another approach, the 'Spiral Model' emphasizes an iterative approach to software design, wherein business needs, customer needs, and engineering requirements are continually reassessed as a project progresses (Pfleeger, 1997). Emerging from these experiences, HCI-centred development models have emphasized user requirements, cognitive and task modelling and interface evaluation. Preece (1994) describes a generic model for user-centred design of software systems. This model (Figure 2.11) incorporates an initial phase of user and data requirements gathering, followed by design activities, wherein particulars of the system architecture and user interfaces are created. These can then be used to describe a specification for a prototype, which can be of either a low fidelity to the design concepts (e.g. a paper prototype) or high fidelity, in the form of a functional working prototype. Evaluation of the prototype by informal assessment or usability testing can then suggest design improvements in an iterative and continuous cycle. When the functionality is judged to be adequate, a software release can then be issued. Drawing upon these experiences, a few researchers in the visualization community have begun to propose Reference Models of visualization systems. These Reference Models, which are attempts

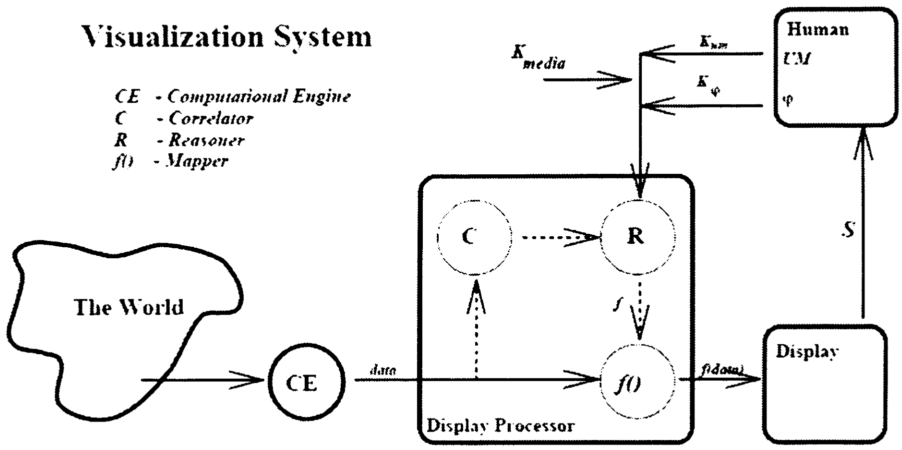


Fig. 2.12: Csinger's (1992) model of visualization systems.

to codify the components of visualization systems, have differences in emphasis, but their basic components are similar: they all entail an approach which involves manipulation of the visual representation of data by human interaction.

Csinger (1992) describes a model for visualization systems drawing upon research in psychophysics, automatic display generation and multi-dimensional data visualization. Csinger's model (Figure 2.12) is a general, high-level abstraction of the major components of a broad range of visualization systems. This model is based upon the work of Ware (2000), Roth and Mattis (1990), and Bertin (1967), who articulated the capabilities and limitations of human visual processing, as it relates to abstractions of data. In Csinger's model, real data in the world such as weather-related data are interpreted by a Computational Engine (i.e. a computer) which performs some computation on them. The output is sent to a Display Processor which reduces or alters the dimensions of the data algorithmically in order to match the capacities of human perception and sends them to a Display. This allows a Human User to view and manipulate the Display Processor interactively.

Similarly, Robertson and DeFerrari describe a model which entails the input of data into a visualization system from many possible sources (Figure 2.13). The data are then transformed according to a set of visual attributes and rendered to the screen. At several points in this process, the human user can intervene to modify either the data or the representations that are encoded and displayed.

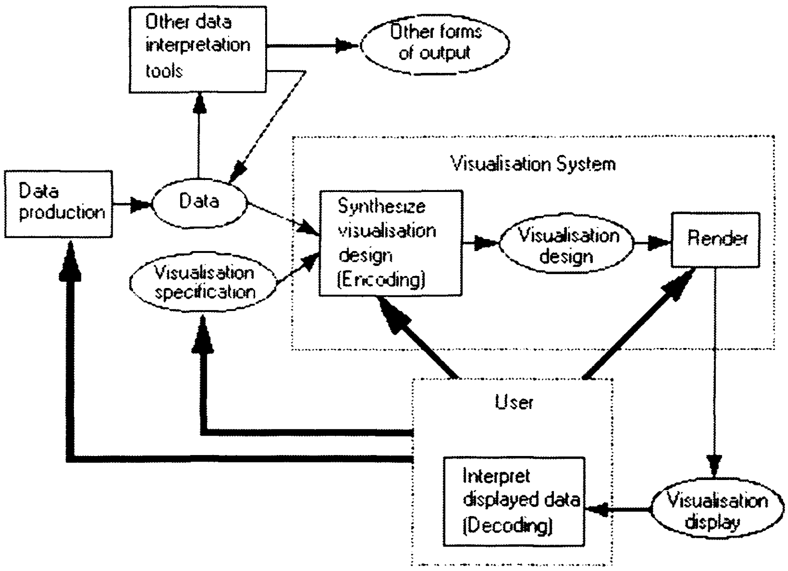


Fig. 2.13: Robertson's and DeFerrari's (1994) model of visualization systems.

Most recently, Card et al., (1999) have proposed a basic visualization Reference Model which attempts to capture the activities involved in Information Visualization design. The model (Figure 2.14) describes the activities that must be completed to create new visualization systems. In this approach, Raw Data undergo *raw data transforms* into structured data as data tables, which can be more easily manipulated and so that its features can be identified. This structured data undergoes additional *data transforms* so that salient derived results can be calculated. These attributes, such as means, frequencies, and other meta-data, describe the data extents and characteristics. These derived attributes then undergo *visual mapping transforms* wherein the structures inherent in the data can be mapped to abstract visual structures. When graphical views are calculated, the visual structures can be represented by view transformations on the screen, such as changes in shape, colour, size, location, etc. These views can then be altered by human interaction with the system. Such interaction changes characteristics of the transformations and mappings so that the visual representation can then be changed to allow exploration of the data.

These reference models attempt to capture the salient features of Information Visualization systems. Each description varies slightly in both terminology and approach, but all of them describe some means of altering the visual representation of a collection of

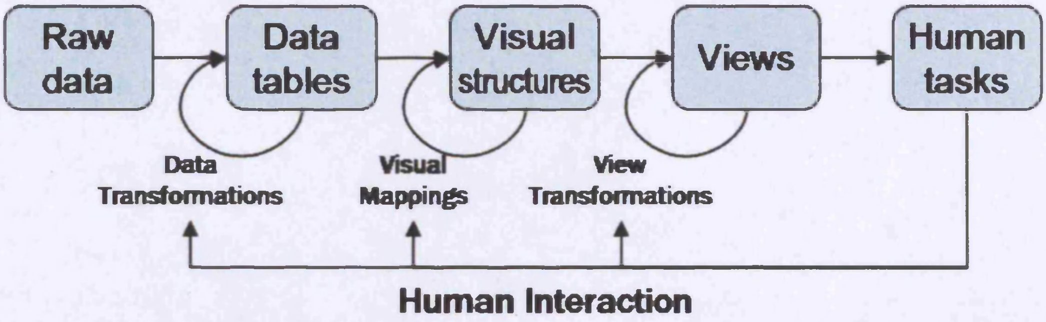


Fig. 2.14: A Visualization Reference Model (adapted from Card, et al. 1999).

data to be more easily manipulated by the user. Yet, the description of how to use the reference model is essentially data-oriented:

> *The reference model of Information Visualization developed in this chapter approximates the basic steps for visualizing information. The first step is to translate Raw Data to a Data Table, which can then be mapped fairly directly to a Visual Structure. View transformations are used to increase the amount of information that can be visualized. Human interaction with these Visual Structures and the parameters of the mappings create an information workspace for visual sense making.* (Card, et al, 1999)

Although reference models make visualization design methods easier to understand and undertake by breaking up the process into constituent parts, they do not provide guidance in how to create the new visual transformations or interaction activities that comprise a novel visualization. Moreover, they do not address the creativity and problem-solving challenges which occur in the design process.

In addition to these reference models, there are a number of systems which have attempted to automate the creation of visualizations and thus, to make concrete some of the procedural steps of visualization design. Notably Mackinlay (1986), Lange (1995), Zhou and Feiner (1998), and Salisbury (2001) have described systems to create visualizations automatically. These systems vary in implementation but all involve automated parsing of a data set, assignment of visual attributes to the data, and provision for manipulation of the representation of these attributes. For example, based on a system of ranking, the software described by Mackinlay could map the position of a quantitative variable to its most effective representation on the screen. Other variables might be represented by colour, texture, shape, etc., depending on their importance. The layout of these elements would be determined algorithmically. Other systems of this type or which offer libraries of techniques are Prefuse (Heer, et al., 2005), The InfoViz Toolkit (Fekete, 2004), and the XML visualization toolkit (Baumgartner and Börner, 2003).

Because they have a limited repertoire of visual attributes and presentations, automatic visualization systems have a limited set of possible visual representations. For example, the system described by Salisbury in the domain of urban planning offers a specific set of visual abstractions in the form of charts, plots, tables, 2D maps, 3D maps, surface maps and others. When no visualization can be mapped to a single representation, the system presents multiple views. Regardless of the presentation method, however, the visualization system must draw upon a pre-determined set of visual representations. This rules out or tends to limit opportunities for creative explorations of novel and abstract representations.

2.6. Discussion

These four categories of research, Examples, Taxonomies, Guidelines, and Reference Models represent the current state of knowledge in the design of Information Visualization systems. They are useful, in that they are beginning to define the boundaries of the discipline. However, in practical terms, when designers seek to create new visualizations there are shortcomings. For people who are not intimately familiar with the visualization knowledge domain, the diversity and discontinuity of design knowledge presents a high barrier to understanding how to create new tools. None of these research areas offers a comprehensive and thorough description of how to approach novel visualization design problems. People can use Examples as an inspiration for their own design solutions, adopting a 'case-based' design approach, and often do. Spence (2001) refers to these as 'point solutions'. Reports of new visualization systems regularly cite the preceding tools which have inspired the reported work. However, examples offer limited help in surmounting the challenges of new designs for visual representation and interaction. They merely present the 'old favourites'. Taxonomies are only useful in terms of describing the attributes of systems already extant and provide little assistance for designers. Guidelines offer useful suggestions and recommendations of best practice, but they are not unified and they often offer conflicting recommendations. Reference models, which offer the most robust methodological guidance, describe the components of a visualization system which should exist and how those parts should relate to each other. Much of the guidance offered by reference models in terms of actually designing a system is implicit rather than explicit. For example, it is obvious from the visualization reference model (Card, et al., 1999) that visual mappings between data and on-screen visual structures need to be made during visualization design. But, as with the tree-map and the HomeFinder, little guidance about how to do this is offered in the accompanying text. It is also clear that human interaction needs to figure, in some prominent way, in the design process. Implicitly, the process of knowledge crystallization should be supported.

But although their reference model accounts for human tasks, it is left to the designer to interpret which tasks they might perform, how they might perform them, and what components should be made available so that users can achieve their goals. It is up to the designer to fill in the gaps in this high-level model, and indeed, in all of the reference models, even if that designer is inexperienced or unfamiliar with the visualization domain.

Finally, and crucially, the experience and expertise of visualization designers is unaccounted for in all of these areas of research. It is implicit, rather than explicit. Visualization knowledge is captured by successful examples and point solutions. Such knowledge was necessary to generate taxonomies, guidelines and reference models. It is also identified as important by the experts themselves. Experienced designers know the properties of visual representations. Spence (2001) notes:

> *...in the great majority of situations the design of a new visualization tool is a craft activity, the success of which depends upon the designer's understanding of the task for which the tools is intended, as well as the designer's possession of many and varied skills ranging from visual design to algorithm design.*

Yet the expertise which is apparently necessary to address design problems and generate creative solutions is little described in accounts of the visualization domain. Is this knowledge essential, merely preferable or indeed necessary at all? Moreover, as the domain of Information Visualization practitioners is relatively small, how can this expertise be shared with a larger community, particularly with non-experts, so that visualizations can be beneficial to more people? Rather than relying upon a confusing array of disparate knowledge sources, is there a useful, principled approach which they can use to create successful visualizations?

Understanding how to create visualizations with novel representations and interactions remains a problem. How can this activity be described and supported? What steps are necessary? In addition to a need for a knowledge base, which the current literature provides, there is a need to use design techniques for creativity. Spence presents 'point solutions' as a palette of useful techniques which can be extended in new visualization designs. Information Visualization literature serves as a collection of examples which may provide inspiration, but which do not act as a comprehensive guide to solving problems associated with novel visual presentations or user interactions which will enhance knowledge crystallization.

Architecture has used design by example for hundreds of years, but software is more changeable and changes faster. Moreover, design activity draws substantially upon knowledge of previous solutions and these are only likely to be known by people who are already experts in the field. Those who are reporting in the literature tend to focus on a description of the new visualization they have produced and not to describe the entire design process used to generate the new visualization.

This survey has begun to answer the first research question in this thesis: *What are the shortcomings of existing methodologies of Information Visualization design?* In part, the answer is that there is poor representation of the user, and that existing design knowledge is disparate and conflicting. It could be argued that the design of such systems is so complex that no single methodology could adequately capture all of the necessary design tasks. Indeed, the research until now has concentrated separately on the four separate areas of inquiry discussed above. A question arises, then, as to how these different areas of research are helpful for both the *users* and the *designers* of Information Visualization systems. To address the methodological effectiveness of the current knowledge for both groups, two empirical studies were performed. The first experiment was conducted to evaluate the effectiveness for users of the implementation of a notable methodological *Guideline*. The second experiment was conducted to evaluate the effectiveness for designers of a set of structured *Design Patterns* for visualization. The nature of these experiments and the conclusions that were drawn are reported in the next chapter.

2.7. Summary

This chapter has described the foundations of information graphics, how they enhance human cognition and problem-solving and how this has led to the development of Information Visualization systems with the advent of modern computing. Several examples of well-known visualization systems were described. It was also shown that there are four primary sources of design knowledge for visualizations, but that there are shortcomings in this knowledge. In particular, the creative and problem-solving aspects of design have not been captured.

Chapter 3: Review and Evaluation of Two Visualization Design Techniques

3.1. Introduction

The previous chapter presented examples of Information Visualization software and described how they take advantage of human visual processing faculties to support complex tasks of understanding information. It reviewed existing approaches to describing the design of visualization systems, and identified how these do not fully capture the visualization design process. This chapter reports research conducted to investigate this assertion; it consists of two studies which were carried out with visualization users and visualization designers. These studies were performed to evaluate a visualization design guideline and a set of visualization design patterns. The results of these studies are presented, followed by an argument for further investigation of visualization design activity, which is reported in Chapter 4.

3.2. Evaluating a Visualization Guideline

The previous chapter noted benefits and deficiencies in the main areas of research: examples, taxonomies, guidelines, and reference models. A survey of the literature revealed that many authors have cited one guideline in particular, Shneiderman's (1996) 'Visual Information-seeking Mantra' (or simply 'the Mantra'), when reporting their work. The paper offers, in fact, two contributions for understanding Information Visualization methodology: the 'Visual Information-seeking Mantra' and the 'task-by-data-type taxonomy' (TTT). This paper has been cited by numerous authors engaged in the development of novel visualization systems and therefore appears to offer useful guidance to them. Of the 610 peer-reviewed InfoViz conference papers from 1995–2005, the paper describing the Mantra ranked in the top 10% of most frequently referenced works. Notably, those who work primarily outside the Information Visualization community also cite this work when they describe new systems they are creating, which suggests that the Mantra has substantial value for many practitioners, even those who may not be intimately familiar with ongoing work within academia. To further this research, a study was performed to gain knowledge of this guideline.

While the task-by-data-type taxonomy (TTT) suggests useful relationships between data types and user tasks in the context of Information Visualization, the TTT is not unprecedented in the literature. There are efforts by other authors such as Tory and Möller (2004), Wiss and Carr (1999), and Chi (2000) who have attempted to describe taxonomies for the purpose of visualization design. The TTT is interesting in that it maps high-level tasks to data types while offering guidance to practitioners based upon Shneiderman's extensive experience in both usability and in designing Information Visualization software. The Mantra, which recommends 'overview first, zoom and filter, then details-on-demand' is a summary of the most important of seven tasks proposed in the task-by-data type taxonomy. The ethic that informs the Mantra is that improved usability is an important goal and that improving support for the tasks leads to a 'better' visualization. The Mantra can be regarded as a maxim regarding the key principles of the TTT. However, there is some confusion among authors about the Mantra and the TTT. Authors are more likely to cite and use the Mantra, rather than all seven tasks described in the TTT, and they often make little distinction between the taxonomy and the Mantra. This research considers all seven parts of the taxonomy and refers to them collectively as 'the Mantra'.

Before investigating what makes it an important methodological contribution, why other researchers and practitioners frequently cite it, and why it is an important subject for research, it is first necessary to clarify the several user tasks described by the Mantra. The Mantra's suggestion for the design of visualizations is to provide 'overview first, zoom and filter, then details-on-demand'. This catchphrase is often cited by other researchers. The further user tasks not captured by the Mantra *per se*, but proposed by the TTT are 'relate', 'history', and 'extract'.

Overview provides a general context for understanding the dataset; it paints a 'picture' of the whole data entity represented by the Information Visualization. Patterns and themes in the data that may be helpful can often be seen only from a vantage point that comprises the whole view. From this perspective, major components and their relationships to one another are made evident. The overall shape of the data can provide assistance in understanding the information that is encoded. Also, significant features can be discerned and selected for further examination. Such features might not be readily viewable from another part of the data representation or might be obscured from other viewpoints. Revealing these features at the beginning of user interaction can aid the user in filtering extraneous information so that they can complete their tasks more efficiently.

Zoom and filter both involve reducing the complexity of the data representation by removing extraneous information from view and allowing for further data organization. 'Zooming' refers to user-directed adjustment of the size and position of data elements on the screen. 'Zooming-in' enlarges smaller data elements of interest and usually simultaneously removes from view or reduces the size of other data elements that are not of interest. 'Zooming-out' effects the opposite result. Significantly, while the results of both adjustments are symmetrical, i.e. zooming-in and zooming-out are procedurally and visually symmetrical; they have quite different implications for cognition.

Zooming can be regarded as filtering by navigation and change of representational vantage point. Zooming facilitates two *different* cognitive tasks, depending on whether it is zooming-in or zooming-out. In the case of zooming-in, it removes extraneous information from the visual field, to help organize the information into meaningful patterns for interpretation and decision-making. Zooming-out reveals hidden information, usually contextual information that is already known, but which cannot be recalled. This allows the users to rediscover their location within the information space, so that newly learned details about the data representation, usually discovered though zooming-in, can be integrated into a larger understanding. As noted by Card, et al. (1999), the significance of this is described by Resnikoff's Principle of Selective Omission, which posits that organisms need information from the sensory organs to be aggregated into manageable inputs by simplification and organization.

Unfortunately, 'zooming' is often used as a generic, shorthand expression for either 'zooming-in' or 'zooming-out'. This dilution of precision in the meaning of the term can lead to confusion precisely because the cognitive activities that it facilitates are so different. The term 'zooming' is often employed by users to refer generically to scalar changes in representations of elements on the screen, rather than changes in vantage point. Anecdotal examples such as those offered by Bederson (2000) are found in conversations about dynamically changing toolbars which can be expanded or contracted to reveal more finely grained levels of adjustment, or a scalar change of menu items that are represented with a fisheye-type distortion to accommodate limited screen real-estate. This vagueness of meaning has probably evolved because of a lack of colloquial terminology and to distinguish between scalar changes of space (i.e. vantage point) and scalar changes of discrete screen objects, such as text or icons.

Filtering accomplishes much the same reduction of complexity in the display as zooming-in, but without changing the data representation or the user's view. With filtering, the adjustment of widgets in an interface allows for control of which data points are visible. The user can thereby selectively hide or reveal data of interest so that the information can be simplified to aid cognition. However, if there are long delays between adjusting a widget and seeing the results in the display, the efficacy of this method is hindered because there is no apparent cause-and-effect relationship. Thus, the best implementations of filtering are those that update the display immediately as widgets are adjusted. These *dynamic filters* allow users quickly to see how a changed variable affects the data representation. If these widgets adjust the parameters of a database query in order to return results, they are referred to by Ahlberg, et al. (1992) as *dynamic queries*. Card, et al., (1999) provide a thorough review of the numerous examples of the application of dynamic queries.

Details-on-demand refers to providing information in the context and at the time that it is needed. In a typical visualization, many data points are visible in the overview, often from multiple vantage points. Depending on the visualization, the number of represented data items can number from dozens to millions. Limitations of screen real estate and visual complexity make it difficult to provide supplementary information that a data point represents, as the provision of in-depth detail about all of the item attributes may be impractical. The details-on-demand technique provides this additional information on an item-by-item basis, without requiring a change of view. This can be useful for relating a single item to the rest of the data set or for quickly resolving particular issues, such as identifying a specific data element amongst many or relating attributes of two or more data points. Providing these details by a simple action, such as a mouse-over or mouse-selection (i.e. the 'on-demand' feature) allows this information to be revealed without changing the representational context in which the data artefact is situated.

The **Relate** task allows the user to view relationships among data items. Selection of a particular data item can reveal, by changes in representation, items that are related by similarity. For example, selecting a node of a hierarchy could highlight all of the children associated with that node. Supporting discovery of relationships is particularly important where comparisons need to be made among the characteristics of different data objects in the display.

History is part of the collection of tasks in the taxonomy, but it is more accurately described as a set of features. Users should be able to return easily to a previous state in the process of exploring the data (e.g. the 'Back' button). Very often, comparing the current state of representation to a previous state can yield a better understanding of the data. In addition, if users make a mistake, they should be able to recover from it easily. An optimal interface permits these activities by providing an accessible history of the commands issued or a widget that returns the interface to a previous state. In addition, history supports the ability to replay a sequence of changes and assist the user in progressively refining data exploration.

It is often important to **Extract** data to continue work in another context (e.g. importing data to and from a spreadsheet). In the process of using Information Visualization tools, users are frequently engaged in lengthy and complex operations. Information and knowledge that they discover may be important for several different tasks or ongoing work projects. Accordingly, they should be able to extract important findings for use in other computing systems. Extraction can also provide a means of saving work, thereby preventing the need to repeat data manipulations if mistakes are made or data are lost.

3.2.1. Observations

Understanding these techniques for task support, the question arises as to which visualization design problems they can help the most. Indeed, other guidelines may conflict with the Mantra's suggestions. It is not clear whether each technique should be designed into a given visualization program, whether some are more appropriate than others in certain situations, or whether compromises should be made when constraints in the system or the supported task require them. Nor is it clear whether these seven tasks are useful for all of the different data types. Many authors have recognized these shortcomings, even as they praise the Mantra's utility.

At the time of this research, there were 53 peer-reviewed papers that cite the Mantra. They were found in many different publications, including conference proceedings, peer-reviewed journals and symposia, master's theses and doctoral dissertations. Thus, the population of authors is very diverse, as are their skills and familiarity with methods of Information Visualization design. This diversity is significant. Since the Mantra represents summary knowledge gained by experience, occasional empirical evidence, and practice in designing visualizations, it can be considered a heuristic or guideline. As such, it offers benefits to novice designers by highlighting important concepts and to experts by further defining the domain of Information Visualization methodology. This wide appeal and the relative scarcity of methodological knowledge may account for the frequency of the Mantra's citation. Roughly, these citations can be placed into five different categories: implementations (34), methods (7), evaluations (6), taxonomies (4), and other (1). [Appendix A contains a list of the publications cited below, arranged by these categories.]

Implementation papers describe novel Information Visualization systems where the Mantra or task-by-data-type taxonomy contributed to the design method used by the authors. In a typical example describing a document analysis visualization tool, the authors describe how each aspect of the Mantra informs their design approach, writing that it is 'a central principle for Information Visualization' (Costabile and Semeraro, 1999). In another example, the design for a software visualization tool, the authors write, 'Our analysis is based on Shneiderman, who presents seven high level tasks that an Information Visualization application should support' (Maletic, et al., 2001). Another group writes, 'We designed our interface to support the visualization tasks described by Shneiderman' before going on to describe how each part of the Mantra was realized in their system (Tory, et al., 2004b). Still other authors use techniques of the Mantra to propose design implementations (Attfield, et al., 2004). A small number of papers within this category cite the Mantra primarily in the context of describing previous research that has informed Information Visualization design, often describing its importance as a methodological guide (for example, Tory et al., (2004a) and North and Shneiderman (2000). Overall, these implementation papers all rely on the Mantra as a design justification, though they rarely describe why the Mantra, in particular, was selected as a methodological guide. Presumably, the experience of the author and the scarcity of lucid methodological guides motivate authors to build upon the clear and simple recommendations made by the Mantra. Those few papers that do not explicitly state that the Mantra informs implementations still recognize its significance in the evolution of Information Visualization systems.

Methodology papers describe methodological approaches to the design of Information Visualization software or describe new models of interaction. Laying the groundwork for their approach, Amar and Stasko (2004) write that, 'Shneiderman's mantra of "Overview first, zoom and filter, details-on-demand" nicely summarizes the design philosophy of modern Information Visualization systems'. Hetzler et al. (1998) recognize the Mantra as important, though they suggest, 'no single paradigm or visual method is sufficient for many analytical tasks'. In an unusual example that describes a method for development of a task model, Becks and Seeling (2001) combine Shneiderman's work with the task models of Wehrend & Lewis (1990) and Belkin et al. (1994) to create their own task model for analysis of collections of documents within the domain of knowledge management. While the authors do not describe a methodology for Information Visualization, their paper is notable because they describe Shneiderman's work as a 'domain-independent model[s] for visual retrieval and analysis tasks'. The high-level nature of the Mantra is of particular utility because its coarse granularity allows them to develop a more specific domain-dependent model. This approach, leveraging the Mantra toward further ends, is typical for those authors who describe new methods.

Evaluation papers use the Mantra as a metric by which to measure the effectiveness of Information Visualization implementations or relate other authors' use of it. Citing Shneiderman's contributions, Miller, et al. (1997), note that the identification of data types is important for evaluation, because such identification makes it easier to compare the similarities and differences of a variety of different visualization types. Wiss, et al. (1998), use the Mantra as a specific measure against which three different 3D visualizations of hierarchies are measured. In their evaluation design of these hierarchy browsers, they write, 'Our task analysis is based on Shneiderman, who presents seven high level tasks that information visualization *should* support' (emphasis added). Interestingly, these authors are perhaps unaware of Shneiderman's caveat that his recommendations were not meant to be prescriptive. They construct a matrix by which to compare the tasks against the three visualizations, evaluating each one based on whether it conforms to the tasks described by the Mantra. They conclude that, aside from considerations of data type, the design of a particular visualization may not always be able to support all of the tasks, suggesting that several different designs might have to be implemented in the same application. This conclusion echoes the justification described by the authors of the Snap-Together Visualization environment (North and Shneiderman, 2000). Among evaluation papers, a common complaint is that beyond usability studies, there are few established metrics by which to measure the effectiveness of various visualizations.

Taxonomy papers describe ways of classifying Information Visualization, its characteristics and salient concepts. These papers involve developing frameworks for organizing and understanding ideas in Information Visualization or offer categorizations of existing tools. Because Shneiderman's paper is taxonomical, other authors who list the artefacts of Information Visualization in taxonomies refer to this work. Chi's (2000) taxonomy references Shneiderman's paper as one of the previous contributions in this area. Describing their taxonomy, Tory and Möller (2004) problematize classification based on data type alone. They suggest a system that divides visualization into Discrete or Continuous models, dispensing entirely with the distinctions between 'scientific' and 'information' visualizations. Their proposal represents a substantial departure from descriptions proposed by Card, et al. (1999) and therefore, highlights potential shortcomings of the Mantra that warrant further examination.

One type of citation that does not fall into any of the above categories is the general discussion of Information Visualization as a discipline. For example, Chen (2002) cites Shneiderman in his 2002 editorial column for the journal, *Information Visualization,* though he does not specifically address it in the text. Such references show that Shneiderman's taxonomy is seen as a useful contribution to visualization in general. It is likely that similar examples exist that were not uncovered during our review.

What is interesting in most of these cases, but particularly the implementations, is that while the authors cite the Mantra as an important starting point for designing their tools, many don't actually specify how they use it. There is rarely a relationship described between the specific tasks or data types Shneiderman details and the particular characteristics of the visualization system under discussion. Most often, the Mantra is merely cited as a general, guiding principle for Information Visualization design. Indeed, this is what its author intended. However, if it is the case that the Mantra is used as a 'guiding principle' in implementation, it is reasonable to question which aspects of it are particularly relevant for users and can therefore yield an improvement in the final Information Visualization design.

Shneiderman has described the Mantra as 'descriptive and explanatory' rather than prescriptive (see Card, et al., 1999). This caveat notwithstanding, it has been widely cited by researchers developing novel Information Visualization tools as a justification for their methodological approaches. In effect, the Mantra has become a prescriptive principle for many information visualization designers. Although many authors cite the Mantra, there are no reasonably obvious studies that have validated Shneiderman's recommendations. It is not certain that visualization systems which adhere to the Mantra are more effective for end-users or whether end-users can recognize when its techniques have been implemented. These observations formed the basis for an empirical study.

3.2.2. Objectives

There were two aims of this experiment, which was oriented towards users of visualizations. The first aim was to determine whether differences in the implementation of the Mantra would be detectable by users of two different Information Visualization systems. This would be made evident by differences in subjective usability ratings for the visualizations; the 'better' interfaces should be more usable. The second aim was to determine whether, without prompting, users would voluntarily identify characteristics of the Mantra, such as overview, zoom, and filter in the two systems and to gather their subjective opinions about the two visualizations. It was hypothesized that participants would identify important features readily and that if the Mantra supports important user tasks, participants would discuss this without prompting. This would show that visualizations that use the Mantra are 'better'.

3.2.3. Two views on the same data: the Glass Engine and the Glass Eye

The Mantra's suggestion for the design of visualizations is to provide 'overview first, zoom and filter, then details-on-demand', followed by 'relate', 'history', and 'extract'. It might be expected that noticeable differences in usability would be evident among interfaces that adhere closely to the Mantra and those that do not. Accordingly, this study used two different visualization tools for exploring the same data set, namely the œuvre of the music composer, Philip Glass. Since the Mantra proposes techniques that should yield improved visualizations, it could be expected that so long as a given data set

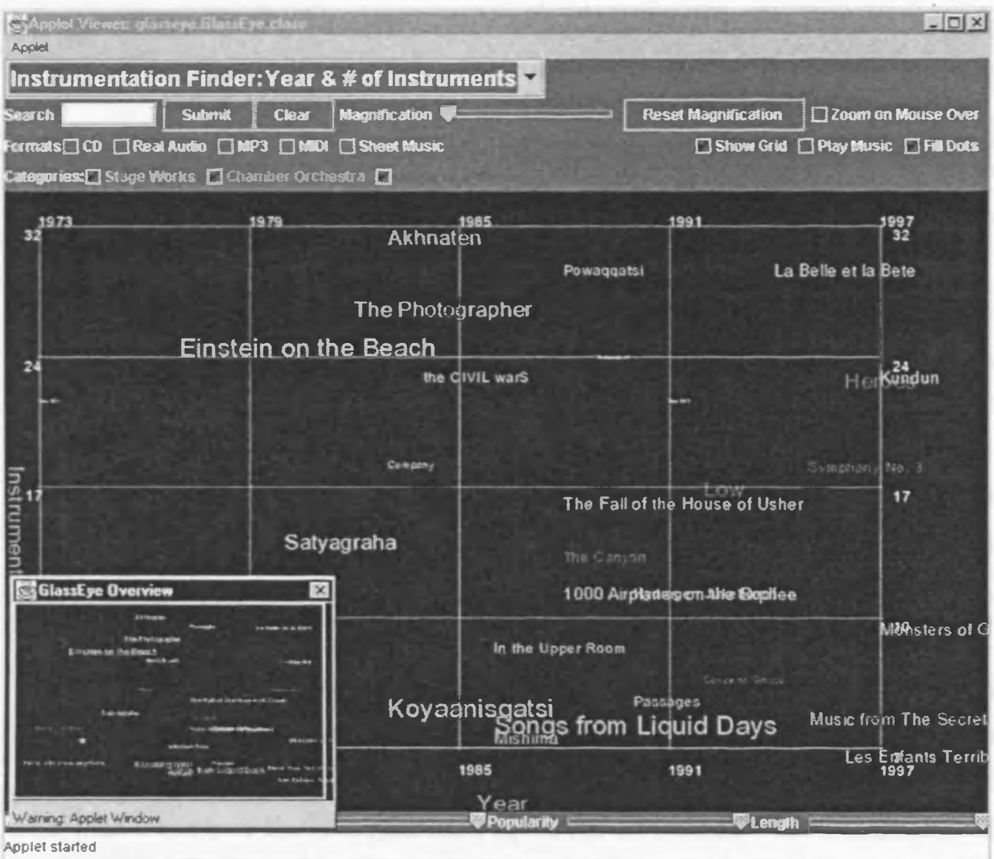


Fig. 3.1: The Glass Eye visualization.

TABLE 3.1: TECHNIQUES OF THE MANTRA MANIFESTED IN THE GLASS EYE VISUALIZATION.

MANTRA TASK	MANIFESTATION IN THE GLASS EYE VISUALIZATION
Overview	The view can be changed so that all data points are visible at once. An Overview Navigation Box provides a view of the entire data space, even when the main view is zoomed-in to a particular item.
Zoom	It is possible to move in the Z dimension using the two mouse buttons. Movements in space are reflected in the display.
Filter	Data items can be added or removed from view by selecting/deselecting check boxes.
Details-on-demand	Selection of an item with the mouse displays pictures and music, for some items.
Relate	Relationships of date, duration, musical category, and musical instrument can be revealed by using radio buttons to colour-code items. Relationships revealed by physical position can also be altered by a drop-down menu.
History	The display can be returned to its starting state by selecting 'Reset Magnification'. This is a limited way of reverting to a previous state.
Extract	None.

remains essentially the same, differences in the visualization designs would affect how users engage with the tools and therefore act as evidence of the Mantra's effectiveness as a design method. It was hypothesized that differences in usability would be evident between the visualizations, depending on the degree to which they supported different recommendations of the Mantra. The results were measured by gathering users' opinions about the usability of the systems, using the System Usability Scale and by a qualitative post-study interview. The two interfaces are described now in detail, followed by a description of the study.

The Glass Eye visualization tool (Figure 3.1) was not intended as a full-fledged production interface. The tool was created as a prototype visualization using the Java 1.2.1 Runtime Environment (JRE) and Piccolo, a zooming user-interface toolkit developed at the University of Maryland. All of the data that are used in the interface are hard-wired into the code and stored locally, along with a small number of audio files. The Glass Eye provides a unique visualization of this data, presenting a scatter-plot of musical works in a 2D representation of a 3D space sometimes called '2.5-D' (Herman, 1989). Navigation in this space is accomplished with a mouse and keyboard.

Individual works are presented in three dimensions, with characteristics of the data such as name, date of composition, and duration mapped to pre-determined positions in the 3D space, which are user-selectable. Some works are also represented by photographs taken from artwork or live performances. Playback of a particular work is accomplished by navigating near to the point in 3D space occupied by the work and selecting the Title

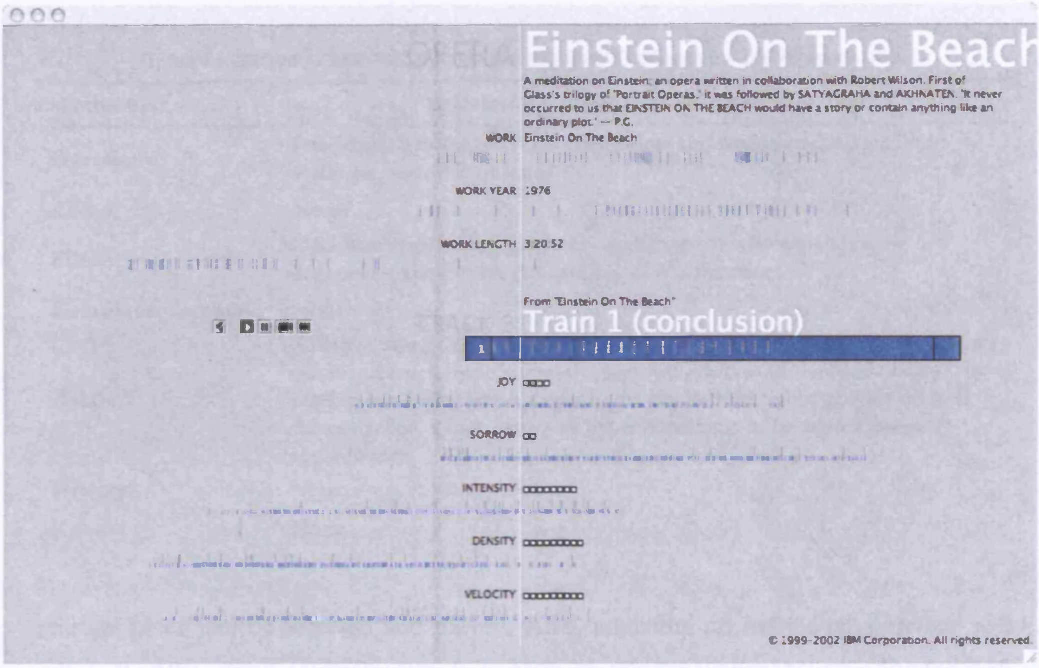


Fig. 3.2: The Glass Engine Visualization.

of the work with the mouse. In terms of the Mantra, the Glass Eye manifests several of the different recommendations, as described in Table 3.1.

The Glass Engine[4] is a production-quality visualization implemented as a browser-based, client-server application. The client portion provides the front-end user interaction and assembles queries to a local database, based on user input. Music is streamed on demand, via the internet, to the client. Although it presents the same data as the Glass Eye, visually, and in interaction, the interface is substantially different. The Glass Engine presents Glass's musical work as a 2D series of nine horizontal slide-bars, or *parallel-coordinates*, which are mapped to attributes of each work, such as Work Name, Work Year, and Work Length. Subjective attributes of Joy, Sorrow, Intensity, Density, and Velocity are also displayed for each item. Individual items are represented by small vertical blue lines within the slide-bars. Each Work has a corresponding marker on all of the nine slide-bars. The playing of a specific musical Work is accomplished by adjusting any of the horizontal bars, so that its specific blue marker is aligned with a white, vertical line at the midpoint of the display. For example, in Figure 3.2 a particular song, 'Train 1 (conclusion)', has been selected from the Opera titled *Einstein on the Beach*. Blue markers for Work, Work Year, and Work Length, as well its subjective attributes, are all aligned to (and hidden by) the central white line. By adjusting any of the slide-bars, any

[4] http://www.philipglass.com

TABLE 3.2: TECHNIQUES OF THE MANTRA MANIFESTED IN THE GLASS ENGINE VISUALIZATION.

MANTRA TASK	MANIFESTATION IN THE GLASS ENGINE VISUALIZATION
Overview	It is possible to see all of the data points simultaneously. All attribute scales are visible at all times.
Zoom	None.
Filter	Data items can be added or removed from view by expanding or contracting the extents (i.e. widths) of the slide-bars.
Details-on-demand	None.
Relate	Relationships of date, duration, musical category, and musical instrument can be explored by adjusting the physical position of data slide-bars, relative to a centreline. Specific items can be selected by placing them at the centreline, which causes all other data items to be repositioned accordingly.
History	None.
Extract	None.

musical piece can be selected and played. Also, adjusting an individual slide-bar will cause all of the others simultaneously to reorient, so that the vertical blue marker for each attribute of the Work which is being selected will be aligned to the white centreline. In this way, the slide bars are linked, but move independently. In terms of the Mantra, the Glass Engine implements several different recommendations (Table 3.2).

A comparison of the two interfaces in Table 3.3 shows the high-level differences in their implementations. The Glass Eye visualization provides more examples of the techniques suggested by the Mantra, lacking only a method of exporting data. By contrast, the Glass Engine supports only three of the seven tasks. Therefore, since the intention of the Mantra is to improve usability of visualizations, it might be expected that these differences would mean that the Glass Eye is more usable than the Glass Engine.

TABLE 3.3: DIFFERENCES BETWEEN TWO VISUALIZATIONS IN THEIR USE OF THE MANTRA. CROSSES (+) INDICATE MANTRA TECHNIQUES THAT ARE USED

TASK	GLASS ENGINE	GLASS EYE
Overview	+	+
Zoom	-	+
Filter	+	+
Details-on-demand	-	+
Relate	+	+
History	-	+
Extract	-	-

3.2.4. Method and administration

This was conducted as a within-groups study, and the order of presentation was randomized. To assess the study aims, ten graduate students from University College London Interaction Centre (UCLIC) were videotaped using both visualizations to explore the music collection. Written consent was obtained. The participants received a brief training session in both systems, wherein the overall characteristics and modes of navigation were highlighted. The important features of each tool were described and assistance in using the controls was provided, as appropriate. Because visualization software is not always meant to be task-oriented, and therefore should allow data exploration, users were allowed to explore freely and to locate and play compositions. Specific tasks were not defined. A think-aloud protocol (Ericsson and Simon, 1993) was used to encourage participants to describe their experience of using the two systems whilst they worked with them. After they indicated that they had adequately explored both tools, the participants completed a questionnaire, the System Usability Scale (SUS) (Brooke, 1996), as a measure of usability (Appendix B, Table 1). This was followed by a verbal debriefing session, using a set of questions following a semi-structured interview approach. Participants were interviewed using a guided conversation, with prepared questions (Appendix B, Table 2) to direct the discussion. Owing to time constraints, the conversations were coded using open-coding to identify major categories, but selective coding and axial coding were not performed.

3.2.5. Results of the study

On average, participants used each of the two interfaces for approximately 10 minutes. The interviews lasted roughly 15 minutes. Data from the SUS survey and the interviews indicated that users expressed a slight preference for the Glass Engine, deeming it more usable and more aesthetically pleasing.

Calculation of the SUS scores as a measure of opinion about usability was performed according to the SUS tabulation guidelines. Figure 3.3 is a box-plot for the SUS scores, which showed a slight preference for the Glass Engine, with an overall usability score of 59/100 (SD=16.42), compared to a score of 52.75/100 (SD=15.25) for the Glass Eye. This corresponds to participants' verbal expressions of a slight preference for the Glass Engine. Except for two outliers responding about the Glass Engine, the SUS scores varied less for the Glass Engine than for the Glass Eye visualization.

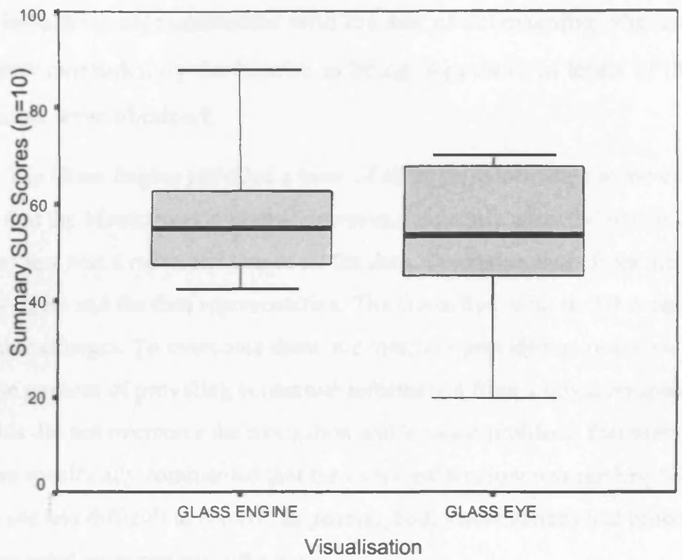


Fig. 3.3: Box-plot of SUS scores for the two visualizations showing a slight preference for the Glass Engine (n=10).

A review of the qualitative comments made during the post-test interviews revealed overarching themes about the interfaces, their characteristics, and behaviours. Generally, users verbally expressed a preference for the minimal design of the Engine, though remarkably they often did not recognize the data points, represented by small vertical blue lines, for what they were – that is, representations of individual Works in the database. Also, this minimalism obscured actionable items. For example, the adjustability of the extents of the horizontal sliders was not readily apparent. They found them unusual and difficult to control. Though it was deemed less aesthetically pleasing, some participants reported that the Glass Eye was easier to use because of the recognizability of its familiar control widgets. However, they reported that they did not find that its mapping of the data in a 3D representation provided any benefit in terms of aesthetic preference or understanding. Users found it difficult to understand what the spatial orientation of the information represented in the Glass Eye and found the interface very difficult to navigate.

Most users expressed a preference for the visual characteristics of the Glass Engine. The minimalist design and immediate feedback were aesthetically pleasing. Paradoxically however, the austere design contributed to confusion about how to interact with the system. Users more easily recognized the 2.5D metaphor presented by the Glass Eye and though navigation was more difficult to perform, they understood better how their interactions resulted in the selection of items of interest and the playback of songs.

The interviews were also conducted with the aim of determining whether users would identify tasks recommended by the Mantra as being important. In terms of the Mantra, the following findings were obtained:

Overview: The Glass Engine provided a view of all of the information in the database, which is a feature that the Mantra would imply. However, frequently users did not clearly understand that the overview was a representation of all the data. Confusion arose from the combination of control widgets and the data representation. The Glass Eye, with its 3D zooming, presented navigational challenges. To overcome them, the interface provided an overview window solely for the purpose of providing contextual information from a broad perspective. However, this did not overcome the navigation and location problems that users experienced. Several users specifically commented that the overview window was not helpful because it was hard to see and difficult to control. In general, both visualizations had problems in the way they presented an overview of the data.

Zoom and filter: The Glass Engine did not facilitate zooming. When using the Glass Eye, users reported that zooming was very difficult to use and the results of zooming were not positive. Six of ten participants explicitly stated that the zooming was difficult for them. It presented problems for navigation tasks and hindered the finding of information. Although it did allow for exploration of the space, this exploration frequently resulted in participants becoming 'lost' in the information space rather than contributing to the users' understand of the information. The filtering capability of the slide-bars of the Glass Engine was very difficult for users to understand, even when given guidance, mostly due to the subtlety of the blue bars representing individual works. It was also not immediately apparent that the unusual control widgets could be used for both navigation and filtering. The filtering capabilities of the Glass Eye, which were much more simply implemented as check box and slider controls, were said to be easier to use, and their effect was much more obvious. It is possible that users more easily recognized filters that were implemented as familiar control widgets.

Details-on-demand: The Glass Engine did not provide details-on-demand in the sense envisaged by the Mantra. The Glass Eye, which did, received positive remarks about this feature. Both interfaces could be said to have provided details-on-demand in the form of music. However, users much preferred textual information to be 'on-demand', particularly for the Glass Eye. They also reported that music-on-demand was often unwanted.

Relate: Users of the Glass Engine remarked that it would be difficult to make relationship comparisons between a song being currently played and other works in the oeuvre, because detailed information about other works is mostly hidden from view while a particular selection is being played. The Glass Engine excelled, however, in being able to delineate relationships among works when users moved different slide-bars, although it took a long time for users to understand how to control these widgets and to understand how the relationships were being represented. The zooming behaviour of the Glass Eye inhibited users from being able to make comparisons among works even though an overview window was provided, ostensibly to aid this activity. Several users commented on this. The colour-coding of different characteristics did aid in relating similar works, but the utility of this was seen as limited and arbitrary.

History: The parallel-coordinates view of the Glass Engine initially presented problems for many users. Users remarked that a history feature would have been helpful to allow them to retrace their steps when they were confused by the changes in the data representation. The Glass Engine did not provide a feature to support this, although it would likely have proved useful for those times when users 'felt lost'. This frequently happened during exploration with the Glass Eye. Participants often resorted to using the 'Reset Magnification' button to set the viewpoint back to its initial state, in what could be called a limited realization of the History task.

Extract: Neither interface offered the opportunity to export information. For the task of music browsing, this did not present any problems. This is significant because it reveals that as long as a user task is supported, particular recommendations of the Mantra may not need to be implemented. In this case, extracting data was irrelevant to users' requirements, so proving that functionality was not essential.

3.2.6. Discussion of findings

Generalizing from these findings, it is apparent that specific techniques of the Mantra are not necessarily applicable to all interfaces and that the interpretation of each of its different techniques is somewhat subjective. While the authors of these two systems did not intentionally follow the Mantra's recommendations, evidence of several of the techniques can be discerned in the system designs, as reported in Table 3.3. Yet it is unclear from this study whether better and more complete implementation of those techniques would yield more usable information visualizations, as envisaged by Shneiderman.

It was also not clear from this study whether the Mantra had a positive effect on usability. Participants expressed an opinion that the Glass Engine was more usable, as shown by its higher SUS scores. The Glass Eye used more of the Mantra's techniques than the Glass Engine, contrary to the hypothesis that use of the Mantra leads to better usability. However, it was not possible to determine whether characteristics of the Mantra were responsible for preference of the Glass Engine's usability or whether other factors such as aesthetic preference were implicated. The post-test interviews did reveal user preferences for the two interfaces and for specific ways that parts of the Mantra were employed, such as overview, zooming, and filtering (e.g. users preferred the filtering approach of the Glass Eye to the Glass Engine). But it was not possible to correlate this with the way that the two interfaces put these techniques into practice. For example, although visualizations are intended to allow data exploration that is not task-specific, the lack of a specific task to complete may have affected participants' perceived impressions of system usability. The vagueness of the task concepts that the Mantra recommends and the variability with which they can be put into a visualization may also have affected user assessments of usability.

While the study showed that participants used some of the capabilities that the Mantra recommends, questions remained. What are the bounds of an *overview* and how should it be navigated? When is *zooming* helpful to users and when is it a hindrance? Must it always be implemented? What is the most useful implementation of *filtering*? How can its effects be made most obvious? Though they might have benefited from it, users did not explicitly express a need for *history*; to what degree is it essential? Is it always important to provide a facility to *extract*? Since the study did not entail putting into practice any of the components of the Mantra, but rather sought their evidence in already existing visualizations, it did not clearly reveal how the different techniques should be applied to design problems.

Moreover, this study did not reveal what it is about the Mantra that would cause designers to value it as a useful method for creating visualizations. It was deemed necessary, therefore, to study a different visualization design method — one without the problems associated with design guidelines. Whilst the participants of the first study were end-users, it was decided that the second study should focus on designers. The intent was to investigate a more thoroughly described method which might more effectively and comprehensively address visualization design problems than a single guideline alone.

3.3. Evaluating Visualization Design Patterns

To build upon knowledge gained in the first study, a very different approach to Information Visualization design was examined. The second study explored Design Patterns for Information Visualization. The following section describes them in detail and presents the quantitative study which evaluated them among a population of designers, with an emphasis on results that shed light on the visualization design 'knowledge gap' which was revealed by these two experiments.

3.3.1. Design patterns: an overview

As noted previously, the current visualization methodology literature is comprised mainly of visualization Examples, Taxonomies, Guidelines, and Reference Models. As demonstrated by the first experiment, the use of guidelines as a design method has many limitations. To improve upon guidelines, Wilkins (2003) proposes visualization *Design Patterns* as a structured set of knowledge about solutions for visualization design problems. Acknowledging the problems with guidelines, Wilkins observes that the value-systems inherent in the different methodological techniques differ because they place emphasis on different stages of development. That is, some models are system or data oriented, whereas others are aimed at improving user experience. To overcome some of the limitations posed by these methods, Wilkins proposes a pattern-based method for design of Information Visualization software. Drawing upon other design disciplines, this pattern-based method attempts to codify and order existing knowledge so that known solutions can be communicated and used for new design problems.

It is not surprising that for any design discipline there are problems within a domain that tend to arise, time and again. Although solutions may be known, such solutions are not helpful unless that knowledge is coherently structured and communicated to colleagues. Without structure, design knowledge may be disparate and disjointed, and relationships among solutions may not be apparent. Without good communication, designers who are unaware of solutions may be forced to 'reinvent the wheel', wasting unnecessary effort on a problem that has already been solved by others. Some means of capturing and codifying solutions to design problems would be useful for those engaged in practice.

To address this challenge in the field of architecture, Alexander, et al. (1977), identified a collection of *design patterns*. Drawing from many years of experience as architects, they catalogued a set of over 250 solutions known to work for specific problems in building and town planning, which were presented in a structured and easily usable format (Alexander, 1979). The patterns present the solutions to design problems that tend to recur within a certain context. In this way, the design patterns are an attempt to consolidate useful knowledge about building and space design and to support the creation of new buildings and environments. Each pattern is presented in a clearly written and ordered manner, with several pages of supporting text and an illustration. An example can help to illustrate how patterns are structured and what they offer to the designer.

Figure 3.4 presents an abridged version of the 'Alcoves' pattern presented as a table, for clarity. Like all of the patterns, it contains specific elements: a Title, Context, Problem description, related Forces, a Solution, some visual Examples, and a list of Related Patterns. This pattern offers a specific solution to an architectural design problem, that of providing a semi-private space for social interaction that is not cut-off from the rest of a room. It codifies design knowledge in a structured manner and communicates it meaningfully. This structure allows other patterns to be proposed and defined similarly. Moreover, groups of Related Patterns can be combined to address problems that arise in a particular scenario. Alexander described these combinations of patterns as a *pattern language*.

In the intervening decades since they were first developed in the domain of architecture, computer scientists and software engineers have adopted the idea of patterns as a useful way of structuring knowledge about the design of computer software. Most notably, Gamma et al. (1995) have developed a set of software patterns that attempt to bring some order to knowledge about design of software systems. These patterns have been taken up by others, who have used them to create their own solutions to software engineering problems and who have extended them to address new problems and solutions.

As user-interface design is intimately tied to the design of software systems, some in the HCI community have also adopted design patterns as a means of addressing recurring design problems. One such series of design patterns for HCI has been proposed by Borchers (2000a, 2000b). Building from a formal syntactic model of interdisciplinary design patterns, he describes how design patterns can be applied to specific usability engineering problems. He presents an example of the use of a pattern language to address a range of details involving the user experience of an interactive exhibit. More recently, Dearden and Finlay (2006) have shown that patterns can be used in participatory design, as a technical lexicon, as organizational memory (knowledge repository), as lingua franca, and as a design rationale.

Borchers also posits that patterns may be more effective for user experience design than for design of the underlying software. In the same way that the products of architecture are meant to be used and experienced by people who inhabit them, user interfaces are meant to be experienced by the people who interact with computers. Borchers points out that most people are not likely to alter or even see the code of the software that they use. Thus, the use of patterns in a participatory design process which

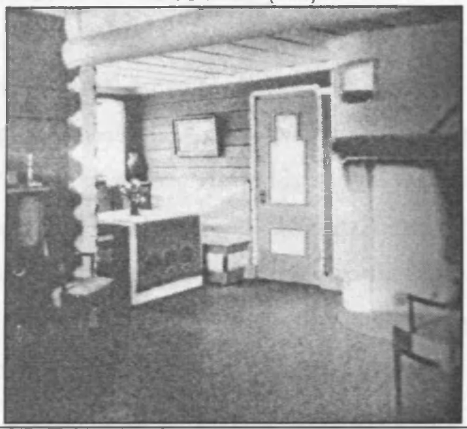
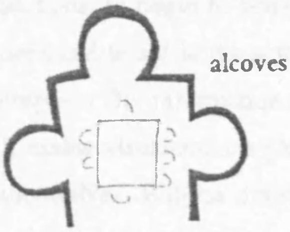
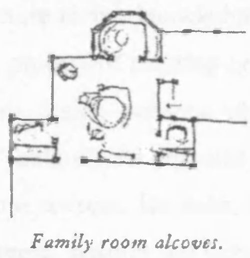
Title	ALCOVES (179) 
Context	This problem is most acutely felt in the common rooms of a house—the kitchen, the family room, the living room. In fact, it is so critical there, that the house can drive the family apart when it remains unsolved.
Problem	No homogeneous room, of homogeneous height, can serve a group of people well. To give a group a chance to be together, as a group, a room must also give them a chance to be alone, in one's and two's in the same space.
Forces	To solve the problem, there must be some way in which the members of the family can be together, even when they are doing different things. This means that the family room needs a number of small spaces where people can do different things. The spaces need to be far enough away from the main room so that any clutter that develops in them does not encroach on the communal uses of the main room.
Solution	Make small places at the edge of any common room, usually no more than 6 feet wide and 3 to 6 feet deep and possibly much smaller. These alcoves should be large enough for two people to sit, chat, or play and sometimes large enough to contain a desk or table.
Examples	  *Family room alcoves.*
Related Patterns	CEILING HEIGHT VARIETY (190), HALF-OPEN WALL (193), COLUMN PLACE (226), WINDOW PLACE (180), BUILT-IN SEATS (202), THICKENING THE OUTER WALLS (211) INDOOR SPACE (191).

Fig. 3.4: Example of the 'Alcoves' Pattern (excerpted and adapted from Alexander, 1977).

actively involves end-users is a more effective place for the use of patterns than the software coding process, and is more akin to the way that Alexander envisaged patterns being used. This view is also articulated by Gabriel (1996), who notes that patterns, as Alexander described them, were meant to accommodate and facilitate the changes that building occupants make to the structures they inhabit as their needs evolve over time.

Similarly, software patterns may be more effective if they invite active involvement of end-users. Pattern oriented approaches, which advocate user-centred methods such as participatory design, may be particularly well-suited for HCI.

This is the line of reasoning that Wilkins (2003) puts forth in support of the development of design patterns for visualization. He holds that visualizations, which involve a highly-interactive user interface which presents different views of data, should be able to take advantage of interaction-oriented patterns presented for HCI and software engineering. Thus, Wilkins proposes a pattern-supported methodology that uses visualization patterns in a participatory design process for the creation of Information Visualization software.

3.3.2. A pattern-supported approach to visualization design

Wilkins observes that the domain of visualization has encountered design challenges similar to those described by authors in architecture and other kinds of software engineering. In general, knowledge of design solutions has been captured in the literature mostly by new examples of visualization systems in the literature and by guidelines. Design guidelines offer only limited help, based on the experiences of practitioners in solving the practical problems of creating visualizations. Wilkins notes that the many design heuristics and guidelines which have been presented for visualizations suffer from inconsistency and contradiction, as previously discussed.

To overcome the limitations, to begin to bring some structure to the knowledge of visualization design solutions and to aid in the software design process of creating novel visualizations, Wilkins proposes 36 Information Visualization design patterns which should support successful, usable visualizations (Appendix B, Table 4). To organize the structure of the patterns themselves, Wilkins draws from several sources. He notes that various authors have structured patterns differently. For example, despite the notable contribution of Borchers' (2000a) patterns for interaction design and the work of Fincher et al. (2003) on a Pattern Language Markup Language (PLML), there is not yet a standard format for HCI design patterns. Drawing comparisons between HCI approaches offered by Tidwell (1999) and Griffiths et al. (1999), Wilkins proposes a pattern structure that is a synthesis of the various approaches from Alexander (1979), HCI, and software engineering (Gamma, et al. 1995).

To better clarify what a visualization design pattern is, it is useful to present one example from this set in its entirety (Figure 3.5). Like the design patterns proposed by

Alexander and Borchers, each of the 36 patterns contains key elements which provide uniformity and consistent structure across the entire set. The key elements are:

- a pattern ***Title*** which describes the pattern at a high level and makes it possible to refer to it easily;
- a description of the design ***Context*** in which the pattern is relevant;
- a description of the ***Problem*** which the pattern is meant to address;
- external ***Forces*** which may be relevant to the problem;
- a ***Solution*** to the problem described in detail, which has been demonstrated to work;
- ***Examples*** of successful use of the pattern including screen-shots, where possible;
- a list of ***Related patterns*** which could be used as alternatives to solve the problem or which could be useful for constructing a pattern language.

The last item in this list provides the possibility of extending an individual pattern by combining it with others to build a complex interface. Thus, like patterns for architecture, the visualization patterns can be related in a *pattern language* to form solutions to interaction or navigation problems (Appendix C, Figure 1). Wilkins notes that the term 'language' may be misleading as it '...implies a syntax, grammar, etc., which are not present in pattern languages'. Indeed, the 'pattern language' described formally by Borchers (2000b) is merely a graph where each pattern is a node and the relationships among the patterns are edges. The usual application of patterns and the intended application of visualization patterns is in the form of a simple hierarchy; Wilkins suggests that a better term for this might be a 'pattern hierarchy', rather than pattern language. Unfortunately, the term 'pattern language' has diffused through the literature, so to address this lack of precision by changing the nomenclature would likely compound the problem.

As noted previously, unstructured heuristics have limited utility and can be contradictory; also they are difficult to use. However, this does not mean that the knowledge they capture is unimportant. Rather, knowledge and experience can be made more useful if it is organized in a manner that is practical and appropriate. With visualization patterns, what is practical and appropriate is that which leads to a visualization tool which enhances understanding of the data and system usability. An

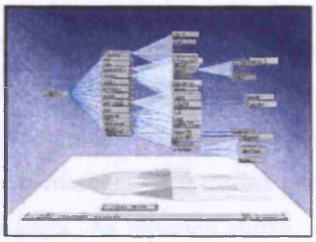
Title	**3D Representation**
Context	There are a number of data items, from one or more datasets, that need to be viewed by the user.
Problem	How to represent the data in visual form?
Forces	• The number of data items may be high. • The structure of the data or the source from which it was gathered requires the use of the third dimension. • The dimensionality of the data is such that it does warrant the use of the third dimension. • Lack of screen space.
Solution	**Use a 3D representation.** **Using three dimensions** provides access to a much large space in which to place the data items. However, 3D representations may require substantial additional work due to the problems of occlusion, the need for depth cues, the need for a simple 3D navigational model, etc. These considerations must be taken into account before the visualisation designer decided to use a 3D representation.
Examples	• Arc Map (Cox et al. 1996) • Cone Tree (Robertson et al. 1993)  • Narcissus (Hendley et al. 1995) • Numerous other examples.
Related Patterns	Visualisation patterns, GUI design patterns.

Fig. 3.5: An example visualization design pattern: '3D Representation' (Wilkins, 2003).

ethic of usability, therefore, should drive the design process which the patterns are intended to support (i.e. the patterns should be usable), and the software that the patterns are intended help design (i.e. the resulting visualization should be usable).

Arguing that visualizations are a form of Human-Computer Interaction, Wilkins compiled a list of usability heuristics for visualizations (Appendix C, Table 1) which were drawn from empirical evidence and personal experiences offered by Brath, Carr, Eick, Foley and Van Dam, Rheingans and Landreth, and Shneiderman, from which the examples above were drawn. Wilkins classified these heuristics based upon the frequency that they occur in the literature and the degree of effect on usability. For example, all of the sources argued that visualizations should be 'task specific', whereas there was less agreement among authors as to whether 'occlusion is undesirable'. The visualization patterns were developed as a way to codify these heuristics meaningfully, to eliminate contradictions, and to overcome the problems of guidelines presented out-of-context.

To test these heuristics and the design patterns that capture them, Wilkins created several different visualizations in the domain of Military Command and Control and evaluated them with simple and complex tasks. The study measured reaction time, accuracy and subjective opinions of usability. Wilkins' research showed that very similar visualizations did not yield substantially different usability, perceived or actual. More importantly, the research showed that while for similar designs the heuristic factors did not appear to predict accurately whether a design would be more usable, for significantly different designs or those where the heuristics have been violated, the heuristic evaluations that inform the design patterns appeared to predict which designs would be more usable. The findings from Wilkins' research appear to indicate that the proposed patterns support creation of visualizations with good usability.

3.3.3. The study

Because patterns present successful designs in a structured format, Wilkins argues that they are more useful than guidelines such as the Mantra, which are problematic because they can be 'difficult to select, interpret and apply,...may be too simplistic, and they may even contradict one another'. A second experiment was therefore conducted to investigate the use of visualization patterns by designers. As reported above, Wilkins' study of the design patterns showed that their use in the design process appeared to result in more usable visualizations. If employment of the design patterns yields usable visualizations, then the patterns themselves should be easily and consistently identifiable to designers so that they can aid the design process. This experiment aimed to assess this at the most basic level. The goal was to determine whether the patterns were consistently recognizable by designers.

From an HCI perspective, an ideal design team would be composed of experts from several different areas, such as software development, perceptual psychology, human-computer interaction, and data mining. Acknowledging that these experts may not always be available, Wilkins posits that the patterns should be usable by 'human designers'. Dearden, et al. (2002), suggest that because design patterns are intended to support a generative process, they should also be useful to designers in the course of the participatory design process. Indeed, this should be possible even at the 'end' of a design process, when a software release has been made. The implication is that to be useful, the patterns must at least be recognizable and consistently meaningful to designers without requiring a great deal of training before their use.

3.3.4. Objectives

Whereas the first study was focused on effects on users, this second study was conducted to address whether human designers would recognize the use of Wilkins' visualization patterns in completed iterations of visualization software. The contention for this work was that if the visualization patterns were meant to be used as a significant part of a methodology for actual design activities, they also should be recognizable by designers in fully functional prototypes of the software. It was deemed appropriate to build upon the previous experiment by using the Glass Eye and the Glass Engine as visualization design iterations against which the visualization patterns could be assessed.

3.3.5. Method and administration

The study involved exposing 20 designers to the existing Glass Eye and Glass Engine visualizations and to Wilkins' 36 Information Visualization design patterns. The goal was to determine whether the designers would be able to identify the presence or absence of each pattern in the two visualizations. The hypothesis for the experiment was that there would be similar responses among participants as to whether patterns were 'present' or 'absent' in the two visualizations. Participants would identify the presence or absence of each pattern and the consistency of their answers would demonstrate a consistent understanding of the patterns in actual visualizations. The rationale for having participants assess two different interfaces was to evaluate consistency in responses across two different visual representations of the same data. This was intended to reduce the likelihood that respondents' answers would have been similar due to chance or because of visual similarities between the Glass Engine and Glass Eye visualizations.

Wilkins argues that the patterns should be useful and accessible for 'human designers'. It should not require a great deal of special training or knowledge for designers to use them. In its most general interpretation, this could be taken to include anyone who is involved in user-interface design, though the patterns would ideally be most useful to those who are designing visualizations. As the research did not require special knowledge of either of the two visualizations or of the patterns, 20 participants who self-identified as being designers generally knowledgeable about creating user interfaces were recruited from the University College London Interaction Centre (UCLIC).

Before carrying out the study, two experts in Information Visualization assessed the patterns to judge whether they were extant in the Glass Eye and the Glass Engine. This was to establish a baseline from which to measure how far participants would deviate in their responses from the presumably 'correct' interpretation made by the experts, as to whether the patterns were present or absent in the visualizations. If the patterns were meant to be usable by human designers, it was anticipated that there should be little variation among the participants and little difference from the experts.

The measures were:

a) the participants' self-evaluated skill as measured by the design experience survey (Appendix B, Table 3);
b) their responses to whether they were able to discern the use of each pattern in the two visualizations.

The first metric on the design experience survey was the number of years of software design experience (0–5+). The second two metrics were subjective responses using a Likert scale. These were: self-ascribed familiarity with user-interface design ('1' for 'not very familiar' and '5' for 'very familiar'), and self-rating as a 'Designer', as the participants understood that word ('1' for 'beginner' and '5' for 'expert').

The study was conducted as a between-subjects design. The visualizations were installed on adjacent computers in a usability lab at UCLIC. All participants viewed only one of the two visualizations. At the beginning of the study, participants completed the design experience survey to indicate their level of design skill (measure (a)). After introduction and description of the experiment, subjects were briefed about the analysis they were to perform and were given a set of 36 laminated cards, each presenting one of the visualization design patterns (see Appendix B, Table 4). The cards were randomized between tests to eliminate any bias arising from the sequence of presentation. The facilitator refrained from providing information about the meanings of the patterns.

For each pattern, participants evaluated the visualization and verbally responded either 'present' or 'absent'. Their answers (measure (b)) were recorded by the facilitator on an answer sheet. This forced-choice approach was intended to yield an opinion on every pattern for both systems, even when participants had difficulty deciding.

3.3.6. Results of the study

After administration, the study data were tabulated and analysed. For each of the self-assessment questions, the mean, median and standard deviation was calculated. The forced-choice responses were added and ranked using a binomial distribution. The details are described below.

One goal of the experiment was to study 'human designers', as opposed to people unfamiliar with design. This was measured with three metrics based on the responses to the self-assessment questionnaire. The results are shown in Table 3.4. The participants had an average of 3.2 years of design experience, with a median of 3.5 years and a standard deviation of 1.85. By the subjective measures, they considered themselves to be relatively familiar with user interface design, with an average response of 3.90 and a standard deviation of 0.85. In self-rating as a designer, the participants judged themselves at just over the midpoint of the experience scale, with an average of 2.95 and a standard deviation of 1.099.

Another objective of the study was to determine whether the responses given by 'human designers' would be similar to those given by subject-matter experts in visualization. Table 3.5 shows the responses given by participants on whether patterns were present or absent. The figures in columns two and three indicate how many of the designers determined that a pattern was present or absent. This can be compared to the evaluation made by the self-styled experts (SMEs), in column four. 'P' indicates that a pattern was determined to be present in the visualization. 'A' indicates that the pattern was absent, according to the SMEs. For example, in the Glass Engine visualization, 10 participants agreed with the SMEs that the 'Filter' pattern was present and one participant disagreed, indicating that 90% of the respondents agreed with the expert opinion. To simplify interpretation, items with a score of 50% or lower are highlighted in red, i.e. less than half of respondents agreed with the experts' opinion.

A third goal of the experiment was to determine whether there would be consistency in the responses of the designers to whether each of the patterns was 'present' or 'absent' in a visualization. To reiterate, the question of interest was not whether each pattern was

TABLE 3.4: SELF-ASSESSMENT RESULTS FROM QUANTITATIVE STUDY 2 (N=20).

SELF-ASSESSMENT QUESTION	AVG	MEDIAN	SD
1. Number of years of software design experience (years)	3.20	3.5	1.85245
2. Rate your familiarity with user interface design (0=not familiar, 5=very familiar)	3.90	4.0	0.85224
3. How would you describe yourself as a Designer, as you understand that word?" (Beginner=0, Expert = 5)	2.95	3.0	1.09904

TABLE 3.5: DATA RESULTS SHOWING % OF RESPONDENTS AGREEING WITH VISUALIZATION EXPERTS. (N=20) A=PATTERN WAS PRESENT, P=PATTERN WAS ABSENT. ITEMS WITH <50% AGREEMENT ARE SHOWN IN RED.

PATTERN	GLASS ENGINE				GLASS EYE*			
	PRESENT	ABSENT	EXPERT	% AGREE	PRESENT	ABSENT	EXPERT	% AGREE
2D representation	11	0	P	100.00%	6	2	A	75.00%
Dynamic queries	11	0	P	100.00%	8	1	P	88.89%
Single direct selection	11	0	P	100.00%	9	0	P	100.00%
Bounding box + keyboard	0	11	A	100.00%	1	7	A	87.50%
Click-n-drag	11	0	P	100.00%	5	4	P	55.56%
3D navigational model	0	11	A	100.00%	5	3	P	62.50%
NAFS model	0	11	A	100.00%	6	3	P	66.67%
Visualization	10	1	P	90.91%	6	2	P	75.00%
3D representation	1	10	A	90.91%	5	3	P	62.50%
Filter	10	1	P	90.91%	8	0	P	100.00%
Selection	10	1	P	90.91%	7	1	P	87.50%
Smooth transitions	9	2	P	81.82%	8	1	P	88.89%
Interaction	9	2	P	81.82%	7	1	P	87.50%
Navigation	9	2	P	81.82%	9	0	P	100.00%
Spatial navigation	9	2	P	81.82%	8	0	P	100.00%
Direct manipulation	9	2	P	81.82%	6	2	P	75.00%
Bounding box	2	9	A	81.82%	3	6	A	33.33%
Single direct selection + keyboard	2	9	A	81.82%	2	6	A	25.00%
2D navigational model	9	2	P	81.82%	6	3	A	33.33%
Redundant encoding	3	8	A	72.73%	7	2	P	77.78%
Multiple direct selection	3	8	A	72.73%	2	6	A	75.00%
Reduction filter	8	3	P	72.73%	7	2	P	77.78%
Non-familiar organizational device	7	4	P	63.64%	8	0	A	0.00%
Small multiples	7	4	P	63.64%	4	4	A	50.00%
Legends	7	4	P	63.64%	8	1	P	88.89%
Context maintained filter	7	4	P	63.64%	5	3	A	37.50%
Familiar organizational device	5	6	A	54.55%	7	2	P	77.78%
Datatips	6	5	P	54.55%	2	6	A	75.00%
Teleportation	5	6	A	54.55%	6	2	P	75.00%
Reference context	5	6	P	45.45%	8	0	P	100.00%
Visual separation	6	5	A	45.45%	8	1	P	88.89%
Level of detail	6	5	A	45.45%	8	0	P	100.00%
Appropriate visual objects	7	4	A	36.36%	5	3	P	62.50%
Details on demand	4	7	P	36.36%	5	3	P	62.50%
Overview and detail	8	3	A	27.27%	8	1	P	88.89%
Navigation box	8	3	A	27.27%	8	0	P	100.00%

* There were items for which one participant did not respond.

in fact present or absent, but rather, whether the designers' responses tended to be similar to each other. A high degree of consistency would appear to indicate that the participants understood the meaning of a particular pattern and recognized its presence or absence. Inconsistent responses would appear to indicate uncertainty among respondents about the meaning of a pattern because of disagreement about its presence or absence. Put simply,

TABLE 3.6: DATA RESULTS (N=20), RANKED BY BINOMIAL DISTRIBUTION.
THERE WAS GREATER DISAGREEMENT ABOUT DESIGN PATTERNS IN RED, TOWARD THE TOP OF THE TABLE.

GLASS ENGINE		GLASS EYE	
PATTERN	BINOMIAL DIST.	PATTERN	BINOMIAL DIST.
Familiar organizational device	0.2256	Small multiples	0.2734
Reference context	0.2256	Click-n-drag	0.2461
Datatips	0.2256	Appropriate visual objects	0.2188
Visual separation	0.2256	3D representation	0.2188
Level of detail	0.2256	Details on demand	0.2188
Teleportation	0.2256	Context maintained filter	0.2188
Appropriate visual objects	0.1611	3D navigational model	0.2188
Non-familiar organizational device	0.1611	Bounding box	0.1641
Small multiples	0.1611	2D navigational model	0.1641
Legends	0.1611	NAFS Model	0.1641
Details on demand	0.1611	Visualization	0.1094
Context maintained filter	0.1611	2D representation	0.1094
Redundant encoding	0.0806	Datatips	0.1094
Overview and detail	0.0806	Direct manipulation	0.1094
Multiple direct selection	0.0806	Multiple direct selection	0.1094
Reduction filter	0.0806	Single direct selection + keyboard	0.1094
Navigation box	0.0806	Teleportation	0.1094
Smooth transitions	0.0269	Familiar organizational device	0.0703
Interaction	0.0269	Redundant encoding	0.0703
Navigation	0.0269	Reduction filter	0.0703
Spatial navigation	0.0269	Interaction	0.0313
Direct manipulation	0.0269	Selection	0.0313
Bounding box	0.0269	Bounding box + keyboard	0.0313
Single direct selection+keyboard	0.0269	Smooth transitions	0.0176
2D navigational model	0.0269	Legends	0.0176
Visualization	0.0054	Visual separation	0.0176
3D representation	0.0054	Overview and detail	0.0176
Filter	0.0054	Dynamic queries	0.0176
Selection	0.0054	Non-familiar organizational device	0.0039
2D representation	0.0005	Reference context	0.0039
Dynamic queries	0.0005	Filter	0.0039
Single direct selection	0.0005	Level of detail	0.0039
Bounding box + keyboard	0.0005	Spatial navigation	0.0039
Click-n-drag	0.0005	Navigation box	0.0039
3D navigational model	0.0005	Navigation	0.0020
NAFS model	0.0005	Single direct selection	0.0020

more randomly distributed responses would indicate less agreement about the meaning of a pattern. The *null hypothesis* would be that participants' responses were due to chance and no different from a random distribution. To refute this, the probability that respondents would have all chosen the same answer due to chance can be tested. The probabilities of the given responses for each pattern were calculated using a *binomial distribution*. Binomial distribution was used because the participants' responses were dichotomous (either 'present' or 'absent'), mutually exclusive, independent, and randomised.

For this experiment, it was important to determine whether, for each pattern, any similarity of participants' responses was due to chance. A numerical analysis of this is presented in Table 3.6. For each visualization, the binomial distribution of the participants' responses for each pattern is presented. The responses have been ranked. Because the focus of the research was to determine which patterns may be most confusing, those exhibiting the least consistency are presented at the top of the table. For a pattern such as 'Familiar organisational device', participants examining the Glass Engine responded as if each participant had simply tossed a fair coin. Similarly, for the Glass Eye visualization, the responses given for the Small multiples pattern were not different from chance. Patterns exhibiting responses with more consistency are at the bottom of the table. The binomial distribution shows that it is much less likely that the participants' responses in regard to these patterns were due to chance; there is more agreement about these patterns. For example, participants were very consistent in their responses regarding the NAFS Model: all 11 participants gave the same response. It is highly unlikely that this would have occurred by chance.

3.3.7. Discussion

According to the self-assessment, all participants believed themselves to be designers of moderate skill and experience. The twenty participants shared a median of 3.5 years of experience with software design, and ranked themselves toward the 'Very Familiar' end of the scale with a median response of 4. Overall, they ranked themselves in the middle regarding whether they considered themselves to be skilled designers, with a median response of 3. These responses indicate that the participants believed themselves to be designers with at least a moderate level of skill and a good understanding of user-interfaces. It can be said that they qualify as the 'human designers' for whom the visualization patterns were devised and were therefore a good representation of the population for whom the patterns were intended.

Interpreting the results from Table 3.5, it appears that generally respondents were in agreement with the subject-matter experts about the presence or absence of specific patterns. For most of the items, in both visualizations, the majority of participants (> 50%) agreed with subject-matter experts. Also, there were no patterns for which there was disagreement with experts, across either visualization. For one notable outlier pattern, 'Non-familiar organizational device', everyone disagreed (0% agreement) with the expert opinion that it was absent from the Glass Eye visualization. This is perhaps due to the specific characteristics of the visualization, as most respondents agreed (63%) with the expert opinion that it was present in the Glass Engine.

It might be expected that if all of the patterns are meaningful and unambiguous, there also would be agreement among designers about many of the patterns and that this consensus would hold, regardless of which visualization was being examined. Table 3.6 shows probability results for each pattern and visualization. For the visualization patterns at the top of the table, the responses given by the group of participants were close to random. There was no consistency in whether the patterns were 'present' or 'absent' in the visualization. For the visualization patterns at the bottom of the table, there was more agreement. For example, answers given by participants were essentially random, regarding whether the 'Familiar organization device' was present or absent in the Glass Engine visualization. By contrast, almost all respondents gave the same answer, regarding the 'NAFS model' design pattern. This would tend to indicate a lack of agreement about the meaning of the 'Familiar organizational device' pattern.

A question naturally arises regarding which patterns were consistently the easiest or most difficult to understand, regardless of the visualization. Which patterns did designers agree about most consistently among themselves? Which ones were the most ambiguous across visualizations and therefore might be more difficult for the visualization designer to identify and use? To arrive at an answer, it is not sufficient to simply add the probability results from the binomial distributions according to the additive law of probability, because this would not control for a situation where a pattern was very consistently identified in one visualization but not in the other. A good example of this is the Click-n-drag pattern. Whilst participants consistently recognized this pattern for the Glass Eye visualization, their responses for the Glass Engine visualization were similar to chance. This would seem to indicate that recognition of that pattern is highly dependent on the visualization which is being assessed. Grouping the results helps to overcome this.

To better reveal which patterns might be easily recognizable, and therefore more useful for designers, the patterns can be ranked into six groups based on their binomial probability scores (Table 3.7). These are:

- Group 1. Consistent: these are patterns for which there was consistent agreement ($p \le 0.054$), for both visualizations.
- Group 2. Good: these are patterns for which there was consistent agreement on one visualization but less consistency ($p \le 0.080$) on the other.
- Group 3. Moderate: these are patterns for which there was moderate agreement ($0.054 \le p \le 0.080$) on both visualizations.

Group 4. Poor: these are patterns for which there was moderate agreement on one visualization but much less consistency, approaching chance, ($0.080 \leq p \leq 0.273$) about the other.

Group 5. Inconsistent: these are patterns for which there was high ambiguity ($0.080 \leq p \leq 0.273$) for *both* visualizations.

Group 6. Biased: these are patterns which had opposite responses for the two visualizations.

For patterns in Group 1, the consistency of responses was very unlikely to be due to chance. 'Filter', 'Dynamic queries', and 'Single direct selection' all exhibited high agreement among respondents for both visualizations. These patterns (highlighted green) can be ranked as Consistent. The next group of nine patterns were slightly less consistently reported across interfaces, though there were slight variations. They may have ranked high in one or the other of the two visualizations, but not in both. The responses for these six patterns (Group 2) were slightly less consistent ($p \leq 0.054$) and ranked in the middle range of the binomial distribution. Notably, these concepts in the Consistent and Good groups tend to refer to concepts that are relatively concrete and which have obvious visual manifestations on the screen.

Group 3 patterns were less often consistently identified across the visualizations and were ranked as 'Moderate'. For both visualizations, the probabilities for these patterns were an order of magnitude higher than those in Groups 1 and 2. It is interesting to note that, with the exception of 'Redundant encoding', these patterns all refer to interaction behaviours and not to physical artefacts that would appear in a visualization.

Groups 4 and 5 account for patterns which appeared to generate a high level of uncertainty, regardless of the interface in question. In some cases these were patterns representing more abstract concepts. For example, 'Appropriate visual objects', had high levels of ambiguity ($p \geq 0.0806$) for both visualizations. Perhaps the subjective interpretation of 'appropriate' contributed to the ambiguity of this pattern.

The last category contains patterns for which probabilities were either very high or very low, depending on the interface. Participants' responses either agreed strongly or were essentially random, but their responses were very dependent on the visualization that was being viewed. These items are in Group 6 (Bias Group). Because of the strongly skewed responses, it is difficult to determine whether the respondents' answers were attributable to the patterns or to the particular characteristics of the visualization that was being assessed.

Certain patterns (identified in boldface in Table 3.7) were ones for which less than 50% of the participants agreed with the subject-matter experts as reported in Table 3.5.

TABLE 3.7: A COMPARISON OF PATTERNS, RANKED BY CONSISTENCY OF RESPONSES ACROSS VISUALIZATIONS. PATTERNS IN BOLD ARE ITEMS FOR WHICH LESS THAN 50% OF RESPONDENTS AGREED WITH EXPERTS.

VISUALIZATION PATTERN	GLASS ENGINE (N=11)			GLASS EYE (N=9)			BOTH VISUALIZATIONS					BIASED
	P≤0.0054	P≤0.0806	P≤0.2734	P≤0.0054	P≤0.0806	P≤0.2734	GROUP 1: CONSISTENT	GROUP 2: GOOD	GROUP 3: MODERATE	GROUP 4: POOR	GROUP 5: INCONSISTENT	
Filter	•			•			•					
Dynamic queries	•			•			•					
Single direct selection	•			•			•					
Overview and detail (<50% agreed)		•		•				•				
Navigation box (<50% agreed)		•		•				•				
Smooth transitions		•		•				•				
Navigation		•		•				•				
Spatial navigation		•		•				•				
Visualization	•				•			•				
Selection	•				•			•				
Bounding box + keyboard	•				•			•				
2D representation	•				•			•				
Redundant encoding		•			•				•			
Multiple direct selection		•			•				•			
Reduction filter		•			•				•			
Direct manipulation		•			•				•			
Interaction		•			•				•			
Single direct selection + keyboard (<50% agreed)		•			•				•			
Bounding box (<50% agreed)		•				•				•		
2D navigational model		•				•				•		
Familiar organizational device			•		•					•		
Datatips			•		•					•		
Teleportation			•		•					•		
Appropriate visual objects			•			•					•	
Small multiples			•			•					•	
Details on demand (<50% agreed)			•			•					•	
Context maintained filter (<50% agreed)			•			•					•	
Visual separation (<50% agreed)			•	•								x
Level of detail (<50% agreed)			•	•								x
Reference context (<50% agreed)			•	•								x
Legends			•	•								x
Non-familiar organizational device (<50% agreed)			•	•								x
3D representation	•					•						x
Click-n-drag	•					•						x
3D navigational model	•					•						x
NAFS model	•					•						x

As might be expected, these tend to be patterns in Groups 5 and 6, the groups associated with a high degree of ambiguity. There are two notable exceptions where respondents tended to agree among themselves, indicating a common conceptualization, but disagreed strongly with the experts. For these patterns, 'Overview and detail' and 'Navigation box' only 27% agreed with the experts that they were absent. Notably, this disagreement was limited a single visualization (Glass Engine), which would tend to indicate that something about that visualization made those patterns difficult to identify.

These interpretations of the data must be tempered by possible problems with the administration and analysis of the results. It appears that patterns in Groups 1 and 2 were easily identifiable by the participants and would therefore prove useful in a visualization design scenario, whereas the more ambiguous patterns in Groups 4 and 5 might prove difficult to understand and interpret. Also, it is tempting to draw a conclusion about the abstractness of a pattern and the ambiguity of responses. Perhaps the more abstract patterns were more difficult for participants to conceptualize. However, other factors may contribute to a misinterpretation of the data.

Most significantly, there could be problems with the quality of the patterns themselves due to such failings as the use of poor examples, lack of effective illustrations, or simply poor writing. It could also be argued that the level of disagreement is due to participants misunderstanding the patterns, failing to notice the presence or absence of a pattern, or fatigue. Furthermore, although the results presented in Table 3.5 show a general tendency to concur with experts, it is possible that the participants were simply less skilled than the expert designers at recognising more ambiguous patterns. Also, although the use of two different visualizations was meant to control for the bias that a particular visual representation might create, the large number of items in the Bias Group, (25% of the total number of patterns), suggests that two visualizations alone may not have been a sufficient control.

Interpreting more broadly, the design patterns provide case-based examples of known solutions, and offer a method for creating usable interfaces, but they are limited. Many of the patterns present interaction techniques, but because they are presented in a static format (i.e. as a screen-shot or descriptive text), it can be difficult to interpret the sort of interaction described by the pattern. Users in the study found this to be the case for 'Details-on-demand' and 'Context-maintained filter'. More problematically, although they offer examples of solutions, use of design patterns alone does not directly support the creativity that must occur during in the design process.

3.3.8. Limitations of pattern-based design

Other authors have begun to explore patterns for development of Information Visualizations and have proposed their own. Drawing upon and extending Wilkins and Alexander, researchers at the University of Oregon, Department of Geography have published five new patterns of their own.[5] This would seem to indicate that the patterns are compelling enough to motivate others to add to and extend them. However, it does not indicate how effective they are for creating new visualizations, and the use of patterns does not address the problems of enhancing the creative aspects of the design process.

Finlay et al. (2002) also describe frequently overlooked characteristics of pattern-oriented design methods. Observing the useful role of patterns in participatory design, they stress that Alexander's intention for patterns to be useful to *users* and not to design professionals. Pattern languages were proposed as a means to help users solve their own design problems and thereby gain a sense ownership and satisfaction from doing so: '...the emphasis is on the pattern language as a catalyst for discussion [among users] and not as something that constrains design activity in a particular direction'. They also note that Alexander's patterns were intended to go beyond merely facilitating communication among professionals. They were meant to engage participants in the design problems they face and thereby enable them to shape their own environments. It is surprising that many authors in software engineering and HCI have overlooked this fact, as it was one of Alexander's fundamental objectives.

Alexander's intention for developing design patterns was not merely a technique to solve a design problem. His motivation arose from a holistic worldview that emphasized the need for people to be actively involved in the design and habitation of spaces in which they live. His emphasis was on improving the quality of life and happiness of people in their daily lives, a focus that is evident throughout his work. This design philosophy, verging on the metaphysical, shaped the design of the patterns and prescribes the manner in which they should be used, which he refers to as 'the Way'. A transcendent quality is also captured by his title *The Timeless Way of Building*. To consider the patterns without acknowledging the ethic that informs them is to overlook one of their key aspects.

Gabriel (1996) recognizes this, observing that the software engineering community has largely ignored the motivation behind the patterns. Yet apparently, this need not be a problem. It is evident that many people in the software engineering community have failed to incorporate the metaphysic of Alexander's philosophy into their design process,

[5] http://geography.uoregon.edu/datagraphics/patterns/index.htm

have failed to incorporate end-users as the 'inhabitants' of their software systems, but still find the patterns useful. Most notably, the work of Gamma, et al., is frequently cited by the community as an important contribution. This contradiction is perhaps explained by the structure that is imposed by the patterns, which organizes knowledge into a coherent, useful and manageable form. Wilkins argues that this structure is useful for the design of Information Visualization software because it organizes the disparate visualization techniques and heterogeneous heuristics, promotes reuse of proved solutions and provides a common language among designers. Specifically, he proposes that patterns are beneficial in the design stage at the point where visual mappings are applied to the data.

While a repository of known solutions is a useful contribution to the body of knowledge about the design of visualizations, unfortunately it does not help designers who are involved in the process of creating ideas about how to represent data in a new and compelling way. As with other visualization design approaches, Wilkins' method for applying design patterns is rather procedural and while it does not preclude creativity, it does not thoroughly articulate how creativity and problem-solving figures in the proposed method. He does suggest that visualization heuristics, combined with the design patterns, can inspire creativity and novelty of visualization designs:

> Mixing the visualization heuristics and the visualization design patterns allows the designer to be both creative and at the same time use techniques that have proven to be effective. (Wilkins, 2003)

However, the effectiveness of these claims is not reported. Also unreported are the reasons why the use of design patterns and heuristics was helpful for exploring design alternatives.

3.4. Conclusions

3.4.1. What the quantitative research yielded

These experiments were conducted to further inform the inquiry into Research Question 1 regarding the shortcomings of existing methodologies of Information Visualization design. As quantitative measures, these studies offered a small window of insight into the design process for visualization, which is a rich and complex human activity. Both were conducted to explore the results of applying design techniques to visualization problems. The first study demonstrated that the use of the Mantra did not lead to better usability and that it was unclear how best to apply its concepts. The second study showed that although they may be inspirational, there is confusion about some of the design pattern concepts, which may make their use difficult in real-world design situations. In both, the responses of participants showed that the common understandings of the meanings of terms such as 'overview', 'details-on-demand', 'filter', etc. may vary widely. As shown by these experiments, the vagueness of the terminology surrounding some of the different visualization techniques is substantial and has significant implications. Particularly for the use of design patterns, where the name of the pattern is meant to summarize what are often very complex topics, ambiguity of terminology may prove problematic for designers.

The conclusion resulting from these two studies is that neither of the two design methods offers sufficient assistance to the designer, as the concepts are difficult to interpret and may not adequately capture the user experience. For a domain in which the user experience is recognized as very important (as is made obvious by the inclusion of user input in all of the visualization reference models described in Chapter 2), this appears to be a significant area for improvement. A more user-oriented design approach should be considered.

3.4.2. What the quantitative research did not yield

Although this research illuminated a very narrow field of enquiry about the specific hypotheses which were being tested, it is difficult to understand the richness of the design process with only these kinds of studies. While the data from both studies showed individual and aggregate responses resulting from specific design methods, questions about the design process itself remain. A comprehensive analysis based on testing dozens of hypotheses would be necessary to even begin to build a robust understanding of the visualization design process through quantitative experimentation.

More importantly, this research did not indicate how these two methods can best be used to support designers in the creation of visualization tools. What was clear from them was that the methods did not address more fundamental questions about designing visualizations. The most useful finding from both studies was that the use of quantitative studies alone as a means to understanding the design problems surrounding Information Visualization is an approach with limited efficacy. A different approach was needed, one oriented towards studying the design process itself, rather than its results.

3.5. Summary

These studies focused on an examination of existing IV software. Thus, they were only able to examine the product of design activities. The results of the research did not reveal ways in which the design process might be changed to assist designers in creation of new, interactive visualizations, nor in supporting creativity and problem-solving activities in the design process. It was therefore deemed necessary to study real designers employing a more user-oriented design approach for real visualization design problems. The following chapter presents a review of techniques to support creativity and problem-solving in design activities outside the visualization domain. A case is made for adopting some of these techniques for Information Visualization design. It then argues for using qualitative research methods to evaluate this approach.

Chapter 4: A Case for Sketching-oriented Information Visualization Design

4.1. Introduction

The previous chapter showed that a more user-oriented approach to Information Visualization design is called for, since current design techniques can be difficult to interpret and do not offer designers a successful user-centred approach. Additionally, current methods offer designers little assistance in design ideation, creativity, and design problem-solving. This chapter reviews the results of a literature review of other design disciplines, as a means of answering Research Question 2. This leads to the hypothesis that techniques from those domains may also be useful for addressing visualization design problems. Finally, a case is made for a third study of the use of *sketching* and *design patterns* to enhance ideation, creativity, and problem-solving for a *bona fide* visualization design problem, using a qualitative research method.

4.2. A Three-part Rationale for Learning from Other Disciplines

Two significant shortcomings in visualization design methods have been identified by this research, and a third can be inferred from this, addressing the answer to Research Question 1. The first of the shortcomings is confusion about concepts in visualization terminology. As reported in Chapter 3, two studies were performed to evaluate visualization design guidelines and design patterns. The findings from both studies revealed that the terminology used in visualization design is often vague and poorly understood among users and designers. Although some of the visualization design patterns were consistently identified among designers, the confusion regarding the concepts could cause problems for both users and designers.

A second shortcoming was also identified. The guidance provided by examples, guidelines, taxonomies, reference models and design patterns does not explain how to build a 'good' visualization. The ethic which informs the design of visualizations is that a 'good' visualization will make data easier for the user to understand and use. It follows then, that the representation of the user is important in the design process. But the fact that the Mantra does not particularly lead to usability, as demonstrated in the first study, shows that at least one very notable guideline fails to represent the user adequately. Also, although the visualization design patterns are based upon usability heuristics, they are so confusing that they themselves can be hard to use.

Since user interaction and visual presentation are the key elements which set visualizations apart from other kinds of software (as demonstrated by their importance in all of the reference models), user-centred design techniques which support these should be fostered and described by visualization design methodologies. Work in the human-computer interaction community has shown techniques such as participatory design to be very effective for creating usable software, but they are not often discussed in IV literature and usability evaluations are relatively rarely reported.

A third shortcoming can also be identified, though it is not directly demonstrated in the experimental findings: the Mantra and visualization design patterns give little support for design creativity and problem-solving inherent in visualization design. Although the design of some visualizations can be decomposed into procedures and aided by guidelines, reference models, or design patterns, the building of visualization tools, like any design process, involves ideation, creativity and problem-solving. Whilst reference models offer the most detailed descriptions of visualization systems, they do not provide support for creating new visual representations or interactions beyond describing which aspects of the system should be present. As a starting point, design patterns are an attempt to structure design knowledge, but they merely imply ideation and problem-solving. Also, they do not address the issue of design creativity.

These three shortcomings lead to the conclusion that a visualization design methodology should be user-centred and should account for creativity and design problem-solving. This knowledge is sparsely represented in the literature. Happily, other disciplines have recognized that these activities can be substantially supported during the design process. In particular, the technique of *sketching* alternatives is promising, but a design approach which incorporates sketching has not been examined by the visualization community. These considerations provided the rationale for a third study. It was performed to evaluate a *bona fide* design problem, in context. In preparation for this study, a review of techniques from other design disciplines was conducted to answer Research Question 2, regarding the techniques which other disciplines use to support creativity and problem-solving in the design process.

4.3. Understanding Design in Other Disciplines

Before addressing how design activity in other domains is relevant to Information Visualization methodology, it is important to note how others have characterized the design process. The use of the term 'design' has become much diluted, in both common parlance and technical discourse, to such a point that it has almost lost any useful meaning. 'Design' can refer to an activity or an object, a process, or an outcome, whilst a 'designer' may have been trained in such wide ranging disciplines as architecture, marketing, or industrial engineering. Opinions about the precise meaning of 'design' vary for almost every author who attempts to address the issue. It is useful then, to examine several different opinions in order to arrive at a working definition for the purpose of discussion.

Jones (1992) recognizes the dilemma inherent in trying to define design and notes that its definition has changed as design disciplines have matured and new methods have emerged. For example, as architecture, industrial design, and engineering have evolved, critical discourse and new methodologies have grappled with whether design should be defined by the tools and techniques that are used or by the outcomes of the process. Jones cites several authors whose definitions are useful for illustrating the diversity of opinions on the matter:

Finding the right physical components of a physical structure (Alexander, 1963)... Relating product with situation to give satisfaction (Gregory, 1966)... The imaginative jump from present facts to future possibilities (Page, 1966)... The performing of a very complicated act of faith (Jones, 1966)...

This variety is interesting, in that these descriptions are all potentially applicable to any number of design situations or disciplines. They are rather generic in character and tend to avoid specific details about particular methods. As Jones notes, this range of opinions is illuminating and is, perhaps, a key to understanding. He suggests that the place to look for answers is in the environment where the designing is taking place, in the context in which design exists. That is, to ask: 'What is the effect of the design process on the environment in which it is situated?' For Jones, the effect is the impact that design has in changing society and the world. He summarizes by proposing that 'the effect of designing is to initiate change in man-made things'. Designers engage in a series of activities involving human-artefact systems which cause this. As with the previous descriptions, this definition is sufficiently broad that it encompasses a wide range of activities, from creating buildings to creating computer software. But it is problematic because it does not allow for any distinction between building an international airport or a new flight booking system. Nevertheless, this definition is a step in the right direction, in that it recognizes the importance of the context in which designed entities exist, yet requires some articulation to be useful for addressing questions of design methods.

One refinement is articulated by Buxton (2007), who describes design activity for developing new technologies, in particular, the creation of new human-machine systems.

Buxton offers that, '...design is a distinct discipline. It involves unique skills that are critical to the molding of these emerging technologies into a form that serves our society, and reflects its values'. Buxton's definition emphasizes the effect of design on society, but adds the important notion that it is a creative process which requires special skills. The 'molding' involves consideration of design constraints, trade-offs, and opportunities. This activity requires a visual creativity that designers must learn. Drawing from Goel's (1995) research which suggests that designers have a different cognitive approach to problems, Buxton argues that the archetypal activity of design is *sketching*. He contends that sketching is a fundamental part of any design process and that designers use sketching in specific ways to help them with the creative activities in which they engage themselves.

These opinions from traditional design disciplines characterize design as a creative process with a social impact. Experts differ on the exact nature of design, but there is a consensus that designers must take into consideration the effect of their work upon users. Sketching is an important part of design activity, primarily because it supports cognition in ways which lead to creativity. In addition to the practical experience of designers over the years which has shown that sketching is useful – indeed essential – to design activity, there is also experimental evidence to support this assertion, as shall be discussed below. This suggests that sketching may have value in visualization design activities.

4.4. Sketching in the Design Process

Sketching is such a successful method for creativity and problem-solving that it has traditionally been central to the design-oriented disciplines of architecture, engineering, and visual communication. It is also a significant part of the pedagogy of these disciplines (Laseau 2005). Its role is underscored by Robbins (1994) who describes the relevance of sketching and drawing in architectural practice, noting that 'As an agenda and a mnemonic, a form of dialogue as well as a visual guideline, the drawing serves as both the subject of conversation and the object of our endeavours'.

In its most general form, sketching is a way of playing with ideas. It is a visual-motor activity which involves visually rendering the world of ideas that are in the mind of the person doing the sketching. It is a way of externalizing the internal (Fällman, 2003). Whatever the medium, sketching involves an activity of creating and eliminating, modifying and compromising, and weighing possibilities. It permits creative exploration of ideas that may not be fully formed, but where there may be a specific goal in mind. This is corroborated by Lawson's (2005) description of design as 'a spectrum of design activities dealing with both precise and vague ideas,...systematic and chaotic thinking,...imaginative thought and mechanical calculation'. A question arises as to exactly how sketching does this. Whilst seasoned designers such as Buxton, Lawson, Jones, and others offer their opinions based on extensive design experience, design researchers require more concrete evidence for the value and mechanisms of sketching. Current research has provided some answers in three ways: by empirically demonstrating the value of sketching, articulating the cognitive support it provides, and describing how this is useful for fostering creativity.

4.4.1. Design value of sketching

One way of demonstrating value is by asserting value metrics and measuring the success of sketching in achieving those metrics. Schütze, et al., (2003) report the results of a study which showed that sketching during design activity produces significant benefits. Their experiment evaluated the use of sketching by industrial designers to solve the problem of creating a backyard barbecue grill. Participants were asked to create a grill meeting specific constraints. The resulting designs were evaluated against metrics produced by a panel of expert designers. Participants were either prohibited the use of sketching, permitted sketching for part of the design time, or allowed free use of sketching until they arrived at a solution. The results showed that groups using sketching produced designs of significantly better functional quality. Participants who used sketching reported experiencing less difficulty in the design process, and that sketching acted as an aid to memory during problem solving.

Van der Lugt (2002) conducted an experiment which analysed the functions of sketching, as opposed to written notes, during meetings for generating design ideas. Employing a technique referred to as 'brainsketching', participants generated individual sketches and presented them for group evaluation. These were used as a source of inspiration for additional iterations of sketching and evaluation activity. Sketching was found to be valuable because it supports individual re-interpretive cycles of idea generation, and enhances individual and group access to earlier ideas.

Heiser et al., (2004) also studied the use of sketching to enhance the activity of collaborative design groups. They conducted an experiment in which participants designed route plans (i.e. maps from point 'a' to point 'b') both face-to-face and remotely. Participants in the study who used collaborative sketching produced more efficient routes in less time than groups who were not able to do so. Their work also showed that by providing an external representation with which team members could interact, sketching enhanced collaborative activities. Sketches established a focus of attention for design groups and eased communication.

Landay (1996) observes that empirical research has shown sketching to improve user interface design in two important areas: design evaluation and formation of ideas. It aids evaluation by preventing designers and teams from focusing on trivial issues such as fonts and alignment and allowing them to concentrate on the larger conceptual issues at hand. It encourages ideas by supporting lateral thinking and 'allows the *designer* to focus on the proper design issues' [italics in original].

The work of these researchers suggests the value of sketching by demonstrating empirically that sketching produces measurable improvements in design activity. The studies above used both statistical significance and the opinions of participants as value measures. However, this body of research does not address which cognitive mechanisms may be implicated in design creativity. These questions have been taken up by others.

4.4.2. Sketching, cognition, and creativity

Design activity is often characterized as a process of creating a new form to meet specific needs. This is captured by the title of Alexander's (1964) work on the design of architecture: *Notes on the Synthesis of Form*. Sketching is a valuable means of externalizing, manipulating and synthesizing new forms during the creative design process. According to many authors, sketching is essential to design creativity. Although the subject of creativity in general is outside the scope of this discussion, it is useful to examine the cognitive mechanisms which enhance creativity and which benefit from sketching.

Many design decisions are based upon intuition. Designers such as Buxton and Lawson have a gut instinct based on extensive experience which leads them to conclude that sketching is important for creativity. But rather than simply asserting that sketching is important, demonstrations of the cognitive benefits of sketching and how these lead to design creativity can provide a concrete basis for the argument that sketching should be incorporated into design activities for visualizations. This is better than simply asserting that 'sketching is important' because it identifies specific links between sketching and creativity.

Creativity is dependent upon cognition, because creativity does not happen outside thinking and reasoning processes. Though its sources may be mysterious, it results from the activity of a thinking, human agent (Boden, 2004). Thus, some activities which support thinking can be said to support creativity. It is reasonable then, to wonder what kinds of cognitive support are useful for creative thinking and what role sketching plays. But identifying the relationships among sketching, cognition, and creativity appears to be difficult. Because creativity and cognition are intertwined, the literature which describes these two concepts can be similarly convoluted and authors do not always make clear arguments demonstrating the links from sketching to cognition and from cognition to creativity.

A different way to consider the problem is by an analysis of the emergent themes in the research. In terms of creativity and cognition, two concepts recur in the literature: lateral transformations and cycles of generation and interpretation (or dialectic). The notion of 'lateral thinking' or 'lateral transformations' refers to related-but-distinctly-different drawings which are produced during sketching activity. A 'dialectic' refers to iterative cycles of idea generation and consideration of alternatives, using sketching. A designer can use sketching to bring forth into the world an idea to be considered, shared, embellished or perhaps discarded. These two concepts are seen as cognitive benefits which sketching provides and which foster activities generative of creativity.

The notion of sketching as a cognitive aid has been suggested by Plimmer and Apperly (2001), who observe that sketching is useful 'as a cognitive support tool during the design process' which aids memory, makes mental images concrete, and enables the designer to 'describe the overall concept and then reorganize, refine and explore the details'. This allows an unstructured problem to be slowly modified and resolved into a final design, and 'makes good use of our innate visual intelligence'. Fällman (2003) also describes sketching as having important cognitive effects because it parallels designers' thinking processes. He sees sketching as 'not primarily a tool, technique, or skill that is available to designers, but rather as the way in which designers think'. Because of this, Fällman (like Buxton), considers sketching to be archetypal of design.

Tversky et al. (2003), also argue that there is a cognitive basis for the use of sketching in the design process, drawing a direct connection to creativity. They characterize sketching as a means of constructive perception: 'Rather than inducing uncertainty or confusion, ambiguity in design sketches is a source of creativity, as it allows re-perceiving and re-interpreting figures and groupings of figures'.

Other researchers have accumulated evidence of the cognitive mechanisms of sketching during design activity and have focused on the way that designers think about problems. Goel (1992) identifies important distinctions between ill-structured problems and well-structured problems, and argues that 'some ill-structured problems require "ill-structured" representations to prevent premature crystallization of ideas and facilitate the generation and exploration of alternate solutions'. Further, he suggests that the uses of ill-structured (i.e. sketched) and well-structured representations are related to different cognitive functions. To make this argument, he describes seven properties of symbol systems for representing well-structured problems, and uses the game of chess as an example of how well-structured problems require accurate symbol systems in order to make them well-structured. By contrast, ill-structured problem spaces not only do not require well-structured symbol systems, but to facilitate cognition about design problems, they require the sort of looseness that sketching provides. Goel bases his assessment upon two protocol studies of design problem-solving among industrial and graphic designers, which compare freehand sketching versus a computer-based paint program with pre-defined symbols. The studies showed that freehand sketching generated more lateral transformations than the well-structured representations offered by the paint program. Furthermore, he implicates the ambiguity and density of symbol systems as the key properties of representations which are associated with successful problem-solving. More ambiguous and dense representations are seen to lend themselves to ill-structured problem spaces.

While Goel observes the importance of lateral thinking in generating design ideas, he does not draw direct conclusions about creativity. Suwa and Tversky (1997) address this issue head-on. They report the results of experiments on sketching activity, which evaluated the complexity and robustness of designed components among novice and experienced designers. Their experiment used a protocol analysis of architects' retrospective reports of their videotaped activities during the design of an art museum. Experienced practitioners used sketches to create rich sets of laterally-related ideas which they refer to as 'dependency chunks', which allowed them to more successfully explore and consider related design alternatives. They report that because of this, sketching enhanced the subjects' creative thinking. This lateral thinking is similar to the lateral transformations referred to by Goel and Landay.

Drawing upon their own work and others, Suwa and Tversky also emphasize the importance of the sketch as an *external* representation. By physically manifesting a mental idea or image, external representations such as sketches provide three cognitive benefits that enhance creativity: memory, calculation, and modelling. First, sketching aids both long-term and short-term memory. External, sketched representations reduce the load on working memory by '...providing external tokens for the elements that must otherwise be kept in mind'. (Suwa and Tversky, 1997). This allows the designer to use working memory for kinds of mental activity other than simply bearing in mind the elements under consideration. Drawing on the work of Goldschmidt (1994), Suwa and Tversky observe that externalizations also provide support cognition because they '...remind the user of conceptual knowledge necessary for problem-solving and of other similar situations that may promote creativity'.

A second benefit provided by sketches is for '...visuo-spatial and metaphoric calculation, inference, and insight'. The visual, spatial nature of sketches and diagrams makes them effective for '...stimulating visual and spatial associations'. External representation of ideas allows them to be compared to one another based on their physical manifestations. For example, a sketch of a room full of furniture can help a designer to make rough calculations about the size of the room, the spatial arrangement of the objects, the numbers of objects which the room can support, etc. Other judgements such as the visual gestalts of 'proximity, grouping, and common fate' are made possible by visual inspection and comparison. This can lead to insights which are based upon such attributes.

Finally, sketching supports modelling by forcing the designer to make specific decisions about the 'organization, specificity, and coherence' of design ideas, 'which, in turn, by inspection, may lead to new discoveries'. This is similar to modelling things in the real world. But rather than building a physical model, the designer can use sketching to try out ideas quickly and easily by putting them together. In order to make design ideas act in unison, trade-offs and considerations must be made about what is possible and what is not possible. Grappling with these compromises leads to creative ideation.

In addition to these specific cognitive supports outlined by Suwa and Tversky, a recurring theme is that designers use sketching during iterative cycles of generation and interpretation. Goldschmidt encapsulates this idea by referring to it as a dialectic of sketching, involving a discourse between the designer and the sketch, using two types of reasoning. She draws her conclusions based upon protocol studies of architects. Describing their sketching activity, she makes a distinction between 'seeing-as' and 'seeing-that'. 'Seeing-as' refers to analogical or metaphorical thinking about the sketches, and deriving new meaning from the sketched entities. 'Seeing-that' refers to developing an understanding of the design consequences of proposed sketched ideas. Goldschmidt suggests that this type of sketch-based reasoning occurs in rapid-oscillation and is an important component of design activity which leads to creativity. That sketches are, by their nature, vague plays a role here. They are 'sketchy' and allow designers who are 'seeing-that' to reinterpret individual figures and groups. During this activity, preconceived notions of a design problem as represented on paper might be changed after a designer has inspected and reviewed the sketched design. Recombination can facilitate detection of new features, new inferences and insights (Suwa, et al., 2001). In group design, the unstructured nature of the sketch opens opportunities for others to participate and to modify design ideas.

Though he does not address the effect of sketching *per se*, Schön's (1991) work has also examined the dialectical nature of the design process. Schön describes this as 'seeing-drawing-seeing', a cognitive act giving rise to creativity, which characterizes a process of 'reflection-in-action'. By this, he means that in considering problems during the course of their work, architects, engineers, designers, and other professionals engage in design as a 'reflective conversation with the situation'. In this way, design reflection is a cognitive activity which gives rise to creativity. Schön uses a case study to illustrate the properties of this activity. In the study, an apprentice architect engages with the problem of creating buildings for a kindergarten. She uses drawings to help her think about solutions to the problem, as a way of externalizing ideas for consideration. Rather than analysing the effects of drawing, Schön's analysis centres on the reflective thinking of the architect. The drawings produced are seen as part of a larger dialectical activity in which she considers the design problems, trade-offs, and possible alternatives. This results in creative decision-making. It is the process of drawing that reveals the parts of the proposed design which are possible and the parts which are not. For example, the pitch of a hill on the building site has implications for both the arrangement of kindergarten rooms and the amount of light which they will receive. Her sketching of the possible building layout helps her to consider what will be the best compromise among the various contending requirements. Similarly, Robbins (1994) and Lawson refer to architects having conversations with themselves about design problems, using sketches and drawings as a means of supporting this interpretive dialogue.

4.4.3. Situating sketching in the design process

Buxton (2007) also describes the ways that sketching is integral to design activity, situating his arguments in the context of the design process as a whole. Drawing upon Laseau (1980), he asserts that the design process is one in which there are overlapping objectives of exploring and elaborating ideas, whilst gradually narrowing the possibilities through decision-making. These are represented as overlapping funnels in the design process in Figure 4.1. In this model, Elaboration is a process of embellishment that emphasizes increasing alternatives; Reduction through decision-making refers to paring down options with the objective of arriving at a final design. From Buxton's perspective, this dynamic is present during any design process and is most active in the earliest phases, where sketching the greatest promise for generating alternatives and supporting creativity. Similarly, Robbins suggests that sketching offers a 'significant degree of freedom' in that it can allow architects to let their imaginations 'roam over an almost infinite set of possibilities'. It can also offer cost savings because designing with sketches and drawings may reduce initial development costs.

This is similar to Goel's (1992) account of design activity, which he characterizes as having four phases. According to this conceptualisation, design involves a process of problem structuring, preliminary design, refinement, and detailing. Preliminary design involves ill-defined problem-solving activities which are based on creating and exploring alternatives. Goel asserts that this is where lateral transformations are important, because they widen the problem space and promote thinking about alternatives. Though this may seem obvious, he suggests that these phases are non-trivial, because they are a claim about the design problem space which may not be found in other kinds of problem spaces. In (Goel, 1995), he argues that it is useful for the problem-solving activities in early phases precisely because it allows ambiguity. He posits that sketching entails 'a process of creative, ill-structured problem solving in which generating and exploring alternatives is facilitated through a coarseness of detail, a low commitment to ideas, and a large number of lateral translations'. The ambiguity of sketching facilitates these problem-solving activities and moderates the tendency to crystallize ideas too quickly. We can draw comparisons to Laseau's model of the early phases of design, which emphasizes generating a widening set of possibilities, represented by the diverging lines in Figure 4.1.

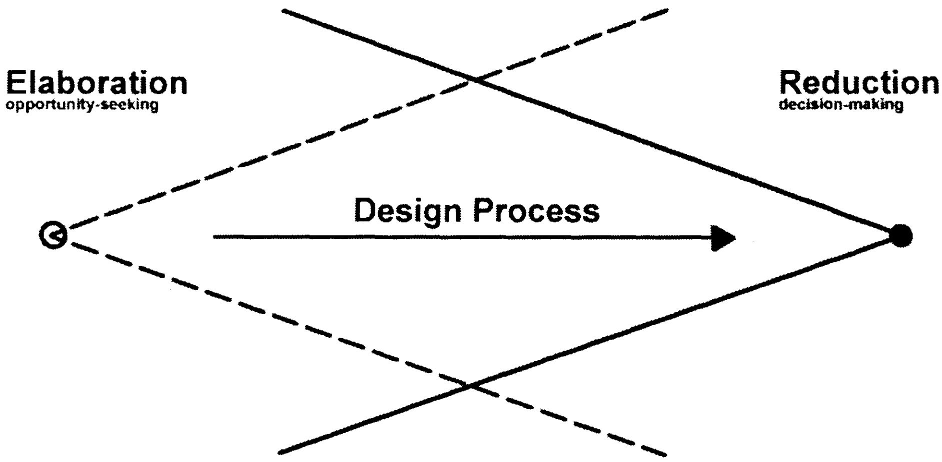


Fig. 4.1: Overlapping funnels in the design process (Buxton, 2007. Adapted from Laseau, 1980).

Recognizing the benefits identified by other design disciplines, the human-computer interaction (HCI) community has long acknowledged the role of sketching in the design process and has situated it in the prototyping process with other techniques such as low-fidelity prototyping and modelling (Fällman 2003). Fällman gives examples of the prototyping continuum in HCI, which include brainstorming, sketching, paper prototypes, wireframes, and modelling. He observes that because of the need to deal with 'interactivity, temporality, tangibility, immersion, sound, and haptics', HCI tends to refer to sketching generically as 'prototyping' and uses a wider array of tools to aid the process, including cardboard, models, wire-frames of interfaces, and even programming environments.

However, Buxton argues that sketching is different from prototyping because it allows greater freedom of exploration. While prototyping is important, Buxton observes that it is crucial to recognize the open-ended and exploratory creativity that sketching enables. He illustrates this argument by comparing sketching to the activities that prototyping involves. Table 4.1 shows the significant ways that the two diverge. In this conceptualization, Buxton highlights a dichotomy between the exploratory nature of sketching and the limiting or constraining nature of prototyping. Examples in the Sketch column, which occur at the early phases of design, are oriented toward maximizing design possibilities, whereas the features which describe a prototype tend to narrow down and crystallize ideas into a specific solution.

SKETCH		PROTOTYPE
Invite	→	Attend
Suggest	→	Describe
Explore	→	Retire
Question	→	Answer
Propose	→	Test
Provoke	→	Resolve

Buxton argues that sketching is critical for supporting early creative ideation in any design process and that it fruitfully contributes to the later refinement processes involved in prototyping as well. Sketching supports design ideation because it invites exploration of design alternatives, allows designers to quickly pose new questions, and provokes creativity. By contrast, prototyping activities seek to reduce alternatives because the outcome of design work is to realise ideas in a working software system or engineered object. Moreover, quickly rendered sketches cost almost nothing to produce, whereas a prototype requires more time, effort, and usually, more money to build.

4.4.4. Relevance to Information Visualization

If sketching can facilitate ideation, creativity and problem-solving in design processes for generic human-machine systems, it then follows that the design of Information Visualization systems design should also benefit from the active cultivation of sketching. Indeed, some people regularly involved in designing visualizations use sketching, as will be shown below. However, few authors describe the role of sketching in facilitating the design of visualization systems. This presents an interesting paradox. Since such systems are highly visual in nature and often employ specialized visual representations (e.g. fisheye distortions, tree-maps, etc.), it could be expected that a technique for supporting visual creativity, design ideation, and problem-solving would be well understood and articulated in the literature. But this is not so. Rather, most of the methodological discourse (particularly the guidelines and reference models) has centred on descriptions of how to create visualizations by altering data representations according to prescribed visual mappings. Whilst this is perhaps to be expected for an emerging discipline, such approaches do not completely describe the complex design activities involved in creating visualization software. This methodological 'knowledge gap' exists precisely where better understanding could yield more fruitful and effective visualizations. A more complete description would include creativity enhancing and problem-solving techniques which could help visualization designers.

To begin to address the shortcomings identified here, we argue that sketching should be integrated into methods for Information Visualization design. The visualization community must formally acknowledge that a creative process occurs, that it is useful to visualization designers and that it is an important subject for study. Furthermore, as design patterns have proved useful to the software engineering and design communities by structuring knowledge usefully, we propose that they can be used in a sketching-oriented visualization design process. It should be possible to adopt and adapt existing visualization design reference models, enhancing them with techniques that support creativity and problem-solving in design work.

This is not to suggest that sketching does not already occur informally among engineering teams. Recognizing the importance of sketching in the early stages of design, several authors have described software tools to aid the process. Buxton, et al., (1983), describes a generic system for graphical design of user interfaces. Apperly (2001), Lin et al. (2000), and Landay and Myers (1995) describe several software systems that support sketching. Gross and Do (1996) demonstrate an electronic cocktail napkin for capturing early design sketches in a collaborative design environment. Notably, Graham (2000) has described the use of initial sketches made by developers as a means to confirm that early ideas for a visualization would meet the expectations of users. Apperly (2001), and Wong, et al., (2006), have also described the use of sketching in their own visualization projects. But whilst it may be that people often engage in sketching during visualization design, relatively few authors discuss it specifically as an important part of the development process, nor do they describe its benefits. Further research in this area is warranted because existing frameworks do not acknowledge the importance of sketching.

4.5. A Case Study Evaluation of Sketching-oriented Visualization Design

The two experiments described in Chapter 3 were undertaken to explore current knowledge of two visualization design methods: guidelines and design patterns. The results of these experiments revealed that existing visualization methods have failed to capture the importance of creativity, design problem-solving, and user representation early in the visualization design process, answering Research Question 1. Furthermore, it was found that the quantitative research approach taken was inadequate to describe human creative behaviour in design teams. These experiments did not evaluate the use of sketching. The review of design techniques for enhancing creativity, design ideation, and problem-solving addressed Research Question 2. To answer Research Questions 3, 4, and

5, a third study was performed using a qualitative research method to assess sketching and design patterns as part of a visualization design methodology. The technique used, Action Research (AR), employs a methodical approach to phenomenological inquiry. It derives validity by requiring a structured theoretical framework on which to base the research. The details of the Action Research method will be discussed in the next chapter. At this point, however, we focus on the framework.

The theoretical research framework was created to address Research Question 3. It incorporated the findings about visualization design knowledge from the first two studies with the previously described sketching techniques to support design ideation, creativity, and problem-solving. It also used visualization design patterns as a source of design knowledge. This four-part, sketching-oriented visualization design framework is presented in Figure 4.2. For the sake of brevity, we refer to this sketching-oriented approach as 'SoViz'. It is important to emphasize that SoViz has two roles in this thesis. It was used as a theoretical framework for the Action Research process, and at the same time it is a practical method for designing visualizations.

The SoViz approach is derived from Preece's model of interaction design (Chapter 2). It depicts a participatory design method which actively and overtly incorporates sketching into visualization design, acknowledges its importance, and recognizes that it supports creative visual representations. To address user representation, the SoViz framework involves active participation of a researcher with the design team in a participatory and iterative process. Because the design of visualizations requires both expertise in the subject domain (i.e. the domain for which the visualization is being created) and in visualization software design, the design team must include a variety of different stakeholders engaging end-users, software engineers, and a visualization specialist in an iterative design process. From the perspective of Stumpf's (2001) design paradigms overview, this can be seen as a social process design approach, wherein visualization designers are engaged in a team effort, within a problem-solving context, though the problem may be ill- or partially-defined. This is augmented by design knowledge provided by a visualization design expert, visualization design patterns, existing knowledge in the form of examples, or a combination of these resources.

The provision of visualization design knowledge is a key attribute of the SoViz approach, which distinguishes it from other development methods. When available, a visualization expert can provide guidance in the form of experiential knowledge to support the social process of design. Design patterns are used to provide the design team

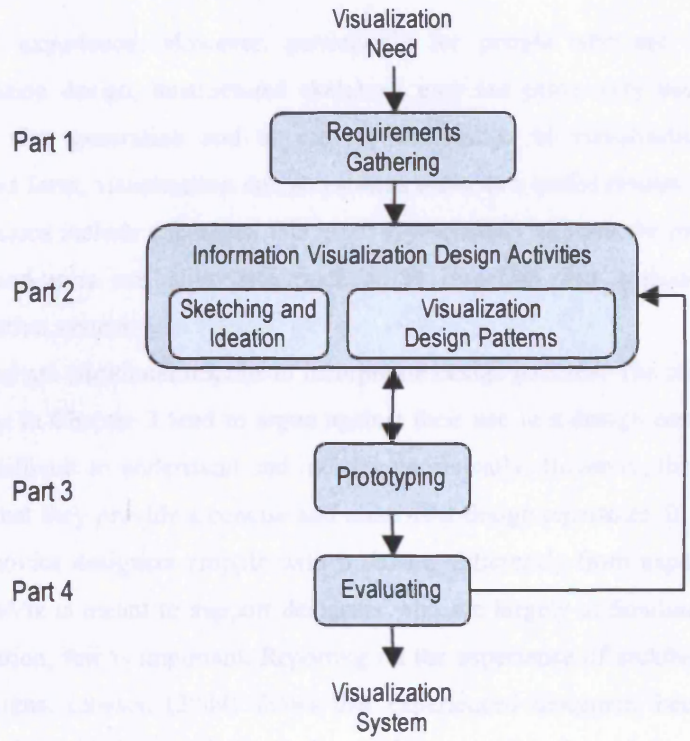


Fig. 4.2: A research framework for sketching-oriented Information Visualization design (SoViz). Compare to Preece et al., Chapter 2.

with knowledge about effective visualization techniques. Significantly, ideation, creativity, and problem-solving, which have been identified as key deficiencies of current visualization knowledge, are supported through collaborative sketching.

As with any software engineering process, the first phase of SoViz entails requirements gathering activities (Part 1). The process involves gathering functional requirements, data requirements, and usability requirements. It may employ common techniques such as dataflow diagramming, entity-relationship modelling, task analysis, and interviews to capture knowledge captured for use in the process of designing new visualizations. A design team can then engage in visualization Design Activities (Part 2) involving design ideation, creativity, and problem-solving which are actively supported by sketching and which may draw upon the structured knowledge provided by visualization design patterns.

Such activity involves all members of the team in a collaborative and iterative process, incorporating the exploratory characteristics described by Buxton, Laseau, and Goel. There may be a person on the team who is knowledgeable about Information Visualization who can contribute to the direction of the activity by offering ideas from personal experience. However, particularly for people who are not familiar with visualization design, unstructured sketching may not prove very useful. Therefore, to support idea generation and to provide knowledge of visualization solutions in a structured form, visualization design patterns serve as a useful resource. As participatory design teams include end-users, this more appropriately situates the patterns in a context where end-users are able to influence the direction and realization of the final visualization system.

There are additional reasons to incorporate design patterns. The mixed findings from the study in Chapter 3 tend to argue against their use in a design context, because they can be difficult to understand and identify consistently. However, the patterns have the benefit that they provide a concise and consistent design repertoire. In addition, as noted above, novice designers grapple with problems differently from experienced designers. Since SoViz is meant to support designers who are largely unfamiliar with information visualization, this is important. Reporting on the experience of architects grappling with new designs, Lawson (2004) shows that experienced designers, because they have a history of knowledge resolving design problems tend to be solution-focused, whereas novice designers without such a history adopt problem-focused strategies. Expert designers can rely on experiential memory rather than theoretical memory. This is significant for the SoViz approach because many people seeking to design visualizations are novices in the domain and are unaware of the history of visualization design solutions. Indeed, the number of people who could be called visualization experts is relatively small, compared to other computing disciplines. Thus, visualization design patterns serve to give these novices a means of addressing the shortcomings of their knowledge about Information Visualization design by providing a design history.

Alexander proposed a similar role for design patterns in architecture. His patterns were originally intended for use by people who were not architects, but who inhabited their own environments and knew their own problems intimately, though they may have been unaware of solutions that are commonly known to architects. When architects were available, they were able to provide design guidance and assistance with the technical problems associated with architecture and construction. This user-centred approach to architecture is the same as the user-centred approach envisaged for design patterns of visualization using SoViz.

The results produced by visualization design activities led to the production of visualization prototypes (Part 3). Ideally, these are produced by software engineers who have been part of the design activities and who will therefore be familiar with the designs which have been proposed. The technologies used for the software (e.g. operating system, programming language, database, etc.) are not specified by SoViz, as these details should be selected by the software engineers. The prototype(s) produced are then subjected to evaluation by the design team. Using sketching and design patterns, the strengths and weaknesses of the prototype are identified and new design ideas are proposed for the next iteration of the prototype. Because this can be construed as additional visualization design activities, the diagram shows this is an iterative process, with evaluation leading to further design work until a decision is made by the team to release the visualization software.

It is important to note that the activities described by the SoViz approach are not intended to be a substitute for techniques described by other reference models for visualization. Rather, they are intended to supplement these design activities. In any visualization design situation, activities such as those described by the reference model of Card, et al., must still occur. In particular, the mapping of data values to visual attributes and interactions will also have to be performed. The SoViz approach is intended as a way to augment these types of visually-oriented activities with idea generation, creativity and problem-solving methods that have been used successfully by other design disciplines. We refer to this as a sketching-oriented approach precisely because it is the use of sketching which enhances Information Visualization design activity.

4.6. Summary

At the beginning of this chapter, we were presented with the problem that current visualization reference models and frameworks may describe aspects of visualization systems, but they do not explain how to put these pieces together to form a novel visual representation or interactive system. Moreover, they do not address the significance of creative, collaborative design, in which the user plays an important role. Recognizing that other design disciplines have encountered this challenge, some of the means of supporting design activities were explored, in particular the use of sketching. It was shown that other design disciplines have successfully used sketching to support ideation, creativity and problem-solving. This may hold promise for visualization design activities.

A case was then made for using *sketching* and *design patterns* as part of a method (SoViz) for generating new visualizations, particularly in the early stages of design. To evaluate the effectiveness of using the SoViz approach, a research method is needed. The following chapter presents a detailed description of the Action Research method and an explanation as to why it is appropriate for investigating Information Visualization design.

It will also discuss the use of the Grounded Theory research method to build theory about the factors involving creativity and problem-solving using SoViz.

Chapter 5: Research Methodology

5.1. Introduction

This chapter describes a twofold approach to evaluating visualization design using Action Research (AR) and Grounded Theory (GT) to address Research Questions 4 and 5, concerning the practical and theoretical outcomes of using SoViz. The research strategy will be presented, followed by a justification of the research aims, a summary of the research questions. The background of AR is reviewed, along with the rationale for its use in a case study, and an argument for the validity of this approach. The effects of using the SoViz intervention were evaluated with this method. Further theory-building analysis of the factors involved in design ideation, creativity and problem-solving in visualization design was conducted with Grounded Theory. A brief description of Grounded Theory is provided, along with the justification for its use. Finally, the authorial voice of the reporting is addressed.

5.2. Research Strategy

The objective of the case study was to explore the SoViz method, using a new visualization in a real-world situation, an activity which requires creativity, design ideation and problem-solving. The strategy for achieving this was to use two research methods: Action Research and Grounded Theory. The AR method was used to evaluate the outcome of the SoViz approach during initial design phases and to aid the organization in producing visualization software to address their needs. The Grounded Theory method was used to analyse these design activities, *post facto*, and to develop theory about the effects of using SoViz. Table 5.1 shows how this twofold research approach was applied.

TABLE 5.1: THE TWOFOLD STRATEGY FOR ACHIEVING THE RESEARCH OBJECTIVES.

RESEARCH TOPIC	RESEARCH METHOD
Case Study application of the SoViz approach to create a novel visualization design for a real problem. (Research Question 4)	Action Research
Developing theory about the outcomes of supporting ideation and problem-solving in the visualization design process. (Research Question 5)	Grounded Theory

5.3. Research Questions Addressed

Exploratory, and oriented towards theory building, this research was performed in a two-stage process of quantitative and qualitative studies. The findings of the two quantitative studies presented in Chapter 3 indicated that it would be important to investigate the design process in practice. Studying a design process in a way that is meaningful and relevant to the community requires working with people who are creating software which is being designed in a real-world situation, and not simply a fabricated visualization design problem. Therefore, a method was required that is known to be effective for studying human processes. Quantitative measures offer limited utility in this area. Qualitative research methods such as AR and GT offer an accepted alternative to this type of inquiry.

The objective of the case study was to address the practical outcomes of using SoViz to account for the representation of the user, existing design knowledge, and sketching, as aids to the creation of visualization tools. As observed in Chapter 4, design disciplines such as architecture have a long history of using sketching, combined with domain knowledge to solve novel design problems. More recently, the software engineering and HCI communities have turned to both early-phase sketching and design patterns to enhance ideation and to solve problems in software design. A handful of authors have described using sketching as an aid to visualization design. However, there are no accounts of combining these two techniques in design settings where people are seeking to produce visualizations. There were three areas of inquiry:

1. Explore the practical outcomes of using SoViz as a method to aid an organization in generating a new, unique visualization (Action Research);
2. Evaluate the effectiveness of this approach and specify the learning from this intervention (Action Research);
3. Develop theory about the effects of using sketching and design patterns to support creativity, ideation and problem-solving on building novel visualizations (Grounded Theory).

The findings of the case study address the Research Questions posed in Chapter 1, regarding the practical outcomes of using SoViz (Research Question 4) and the theoretical significance of this approach for visualization design methodology (Research Question 5).

5.4. Choice of Case Study

The next chapter describes a case study using SoViz to create a novel visualization tool within a research organization at University College London called the Beacon Project. It is important to address the reasons this particular project was selected for the research and to explain why it was a good case study for investigating the use of the SoViz approach. Foremost, the project was selected because it involved groups of people engaging in a design process in order to produce Information Visualization software. Because they had little knowledge of visualization, the participants recognized a need for additional design expertise. They also understood the potential for visualization to enhance their work, they desired to change their *ad hoc* approach to visualization design by following the structured approached proposed by SoViz. They were interested both in the success of their own project and in understanding visualization design, and were willing to participate in a collaborative research process. They were also motivated by the potential to contribute new visualizations to the body of knowledge in their respective disciplines.

The project also presented itself at an appropriate and convenient time. The UCL Beacon Project team were seeking assistance at about the time that the fieldwork was begun and were eager to participate in the research. In addition, the duration of the Beacon Project requirements and of this study were well matched.

Finally, the project was deemed highly suitable because it entailed designing a visualization as a practical solution to problems the teams encountered. This kind of real-world, applied scenario is impossible to replicate in a laboratory setting and is well-suited for exploratory, qualitative methods such as Action Research.

5.5. Action Research: An Overview

The Information Visualization design method described in the previous chapter (SoViz) was used as an Action Research framework for assessing Information Visualization design in an organizational context. The purpose of SoViz is to support design activities among groups of people who are involved in the complex process of building visualization software. Such people are usually part of a larger organization which has a vested interest in its outcome. Therefore, it was important to choose a research method which lends itself to studying complex human behaviours situated in an organizational context. One presumption of this requirement is that it is impractical to study the design process in a controlled laboratory setting because a controlled setting would artificially change the behaviours of the people involved in the study.

Action Research (AR) arose in the medical and social-science fields in the mid-twentieth century to address the problems of studying complex human systems (Baskerville, 1999). The use of controlled experiments was deemed to be ineffective for accurately describing and understanding the nature of such phenomena. At that time, controlled experiments were in the domain of positivist research, which relies on rigorous control of the research conditions, and manipulation of specific variables, followed by measurement of any changes. The expectation of this approach is that any observed changes can then be used to confirm or refute hypotheses about the subject of study, provided that the experiment has been well-designed. However, the phenomena in human social systems are complex, dynamic, and sensitive to outside influences. These factors make it difficult, even artificial to study them in the lab. The very act of creating a controlled setting artificially changes the nature of a human activity that is under scrutiny, rendering it almost impossible to correlate laboratory behaviours reliably with normal human behaviours. Moreover, outside a laboratory, it is not possible to create an environment wherein the researcher can act as a detached observer. Even the use of remote observation techniques, such as video or audio recording equipment, either changes the social environment under study, or poses problems for informed consent of the participants. This paradox is raised by even the most carefully staged remote recording approach: if the equipment is not hidden, the social environment is unavoidably altered by its presence; but if the equipment is hidden, the participants cannot give informed consent, raising ethical issues. Furthermore, use of recording devices alone simply cannot accurately capture the richness and texture of complex human interactions. Some interaction of the researcher is required.

Human social systems are not static entities. Organizations are always undergoing processes of change. In terms of Information Technology, such changes may be the introduction of software or design processes, or those resulting from the use of new information systems, such as email and productivity software. AR studies in Information Technology, such as those reported by Checkland and Holwell (1998) and Fléchais (2005), have been performed to gain knowledge about these organizational change processes.

The key assumptions of the AR approach are that the social environment under study cannot be somehow reduced for study either in a lab or by some mathematical or symbolic method, but that research conducted within an organization (i.e. in context) can yield understanding of the change process. As Baskerville (1999) observes, 'The fundamental contention of the action researcher is that complex social processes can be studied best by introducing changes into these processes and observing the effects of these changes'.

Action Research evolved to address these problems by actively engaging the researcher(s) in the human system that is being studied. Rather than seeking to remove them from the research setting as much as possible, as with the positivist paradigm, their presence is acknowledged and embraced. Rather than attempting to avoid, minimize, or ignore the effect of their presence in the organization, it is welcomed as an opportunity to increase knowledge. Ballantine (personal interview, 2005.) describes the researcher's intervention in the organization as fundamental:

> *Whatever flavour of Action Research you use, even if it's pure 1950s Tavistock Institute style or whether it's 1990s, 21st century Checkland style, it's the common thread throughout, it's where a researcher collaborates in a real life setting that's absolute - if there's no real life setting, then it's not Action Research. They must be wanting to improve their performance in some respect.*

Like other participants, the researcher becomes a subject of study. The researcher's presence necessarily adds both previous knowledge and the researcher's interpretation of the events that transpire. Just as the meanings of events that occur among all of the participants form part of the data, the researcher's interpretation becomes part of the data of the research. Also, an important part of the data is derived from the interaction between the participants and the researcher-as-participant.

5.5.1. Rationale for using Action Research

Action Research must be appropriate for the domain of research to which it is brought. Referring to the fruitful work of other researchers, Baskerville (1999) argues that organizations which are grappling with change processes involving Information Systems are appropriate candidates:

> *One clear area of importance in the ideal domain of action research is new or changed systems development methodologies. Studying new or changed methodologies implicitly involves the introduction of such changes, and is necessarily interventionist. From a social-organizational viewpoint, the study of a newly invented technique is impossible without intervening in some way to inject the new technique into the practitioner environment, i.e., "go into the world and try them out".*

The value of AR is linked to the outcomes that it brings about. The research setting must be one where the researcher's objectives and the organization's objectives are compatible and the researcher's expertise is valuable to the organization. It must be possible for the researcher to be actively involved in the research process and for this work to be embraced by the participants in the organization. The learning from the research should be able to be directly applied, rather than shelved for future reference. It generally entails linking theory and practice in such a way as to be applied immediately so that the changes it brings about can be observed. Baskerville and Wood-Harper (1996) argue that:

> *Action research is one of the few valid research approaches that we can legitimately employ to study the effects of specific alterations in systems development methodologies in human organizations.*

Beginning in the late 1970s and early 1980s, a few individuals such as Wood-Harper (1985), Checkland (1981), and Mumford and Weir (1979) began to investigate the application of AR to the study of Information Systems. Baskerville's summary of this history presents arguments in support and gives an example of the use of AR for studying information systems phenomena in organizations. The AR methodology varies slightly among researchers, but there are some consistent features among all of the approaches. These features, identified in a comprehensive study by Peters and Robinson (1984), are summarized here:

1. **An action and change orientation:** the participants (a group which includes the researcher) have a clearly defined research methodology and a focus on change in the organization;
2. **A problem focus:** there is a specific problem or need that is driving the area of study, which is often summarized in a Theoretical Problem Statement;
3. **An 'organic' process involving systematic and sometimes iterative stages:** it is understood that the change process will occur in several stages over time and that knowledge learned will feed into subsequent iterations;
4. **Collaboration among participants:** the participants and the researcher act as a group to conduct the research together and then participate collaboratively in the change process.

Proceeding from this, we believe that the case study in this research presented appropriate situations for the application of AR. These four characteristics were extant in the case study presented in the next chapter. Combined with a need to understand human phenomena (i.e. design activities) in their natural environment, these form the basis for the use of AR to evaluate the SoViz framework as an aid to visualization design practice.

The study was undertaken with the Beacon Project at University College London. The members of the Beacon Project were seeking concrete action to bring about changes to meet an organizational need: the creation of visualization software. They also agreed to use Action Research to meet this need and to use it as a problem focus. The participants in the Beacon Project understood that the research would evolve gradually, undergoing iterations of change over time. Finally, the organization recognized that software development is a collaborative process and this collaboration could include an outside researcher to participate, assist, and provide knowledge. For these reasons, Action Research was an appropriate way to study the questions raised by the SoViz approach; to understand how sketching and visualization design patterns can be used together to create novel systems in a real-world organizational context.

5.5.2. Research validity

Validity in Action Research is driven by openness and rigour (Baskerville and Wood-Harper, 1998). It is important for the participants to be explicit about the research approach from the outset and at all points of the AR process. It is also important to adhere to this process carefully in order to derive the benefits of the AR approach. The research derives its validity in part from the care with which these steps are taken.

In order to understand how the intervention has changed the organization, the intervention must be specifically described. The pre- and post-intervention states must be measured so that changes that have been brought about can be understood. Baskerville (1999) argues that a model or framework provides a way to organize the changes that will be brought about:

> *The researcher must impose a clear, mutually agreed theoretical framework on the situation, in order for explicit, general lessons to emerge from the research.*

In this research, the SoViz approach described in Chapter 4 provides that framework. For organizational teams that desire to bring about a change in the way they design visualization software by enhancing creativity, design ideation, and problem-solving, this research offers SoViz as the primary intervention. The pre-intervention state is captured in the requirements gathering *diagnostic phase*. The post-intervention results are specified at the end of the *therapeutic phase*. Baskerville (1999) describes other features which are important for ensuring the validity of the work:

- ***Unlike the positivist research paradigm, which might seek to hide the research objectives from participants to maintain validity, the theoretical framework is part of the research and known to all participants.***
  The SoViz research framework described in Chapter 4 was used for this research and agreed by participants.
- ***There is active intervention in the research setting.***
  This work was conducted on-site in the organization where visualizations were needed.
- ***There are carefully determined data collection methods such as recordings and questionnaires.***
  For this work, the intervention sessions were recorded and sketches were saved.
- ***The problem in the social setting must be resolved.***
  The participants should know more about visualization design, sketching and design patterns after the intervention. In this research, this does not mean that these should have yielded a useful visualization, although that would be a good outcome. Rather, the problem of enhancing the participants' knowledge about visualization and design of visualization with SoViz should have been resolved. Ideally, the concrete solution of creating a visualization to meet the organizational need also would be achieved.
- ***The approach should be cyclical/iterative.***
  There were several design sessions and several opportunities to review progress and to make any changes deemed appropriate.

Because this research took an approach which adheres closely to these features of validity, we believe that the findings represent a reasonably accurate reflection of the effect of using the SoViz framework for Information Visualization design.

5.5.3. Stages of the Action Research methodology

For this research, the case study was carried out according to the stages of the Action Research model. The research generally consists of a **diagnostic phase**, where the participants engage in efforts to try to understand the current state of the organization and a **therapeutic phase**, in which the members of the organization participate with the researcher in experiments to change the organization. Changes are introduced and assessment of the effects of those changes are analysed and studied. Although there are

TABLE 5.2: STAGES OF THE ACTION RESEARCH METHOD.

Diagnostic Phase	Diagnosing: assess problem situation in the organization
	Action Planning: plan an intervention using a clearly defined framework (SoViz)
Therapeutic Phase	Action Taking: perform the intervention
	Evaluation: study the intervention effects
	Specifying Learning: reporting what was learned; how the change is manifested in the organization

researchers who differ, AR used to study information technology consists of some generally recognized events. These steps are described in Table 5.2.

First in the AR process, the **Diagnostic Phase** consists of two stages. In the *Diagnosing* or problem assessment stage, the organizational needs for visualization software were identified through individual and group interviews. At a high level, it was determined that such software might potentially be appropriate to the problems at hand. In the *Action Planning* stage, participants agreed to a course of study involving Action Research and the SoViz framework. Both concepts were introduced in general terms and consent was obtained to conduct AR as a means to solving Information Visualization problems.

The **Therapeutic Phase** involves the application of AR and study of the results. In the case study, the *Action Taking* was performed over several successive design sessions, corresponding to the visualization design activities of the SoViz framework. *Evaluation* of the intervention effects used discussions with participants and analysis of the data, in the form of sketches and dialogue, which were generated during the design sessions. Further effects of using SoViz were revealed in subsequent interviews with the participants. *Specifying Learning* was achieved by the description of proposed visualization software subprojects (Appendix D), by publication of peer-reviewed articles, and within this thesis, in the discussion of findings.

At this point, it is important reiterate the distinction between SoViz as a framework for studying using the AR method, and SoViz as a practical design method for enhancing the design process. Using SoViz as a framework for AR research provides a foundation for the validity of the research and provides a structure for intervening in an organization. At the same time, however, SoViz acts as a design method which provides guidance for the activities of the visualization design team. Seen this way, SoViz is both a *research framework* and a *design method.*

This has important consequences for the reporting of this work. Both the AR method and SoViz, used as a design method, involve the evaluation of outcomes. But these two approaches to evaluation are very different. Unlike *Evaluation* in AR, evaluation in the SoViz design method (Part 4) involves an assessment of the visualization software which was produced (as illustrated in Figure 4.2). This means that, in keeping with the *Specifying Learning* step of AR, the results of the AR *Evaluation* are reported in the manner described above, and results of the evaluation of the SoViz design method are reported differently. They are the results of a software assessment, which will be reported in Chapter 7.

5.6. Grounded Theory: an Overview

To build theory about the factors involved in design ideation, creativity, and problem-solving in visualization design, additional analysis can be performed using a research method called Grounded Theory (GT) (Strauss and Corbin, 1998). This addresses Research Question 5. Whilst Action Research was selected to explore the practical outcomes of SoViz used as a method, Grounded Theory was used after the AR intervention to further analyse the factors surrounding ideation and problem-solving and to build theory about their relevance to the design of novel visualizations.

Grounded Theory was developed in the social sciences as a method for *theory development* rather than *theory testing.* It is applied in situations where a researcher seeks to evaluate and understand phenomena, with an aim toward developing knowledge about them and formulating theory. Rather than starting with a theory, *a priori*, and then testing it, the researcher develops a theory from the evaluation of data.

GT is conducted through an analytical process of decomposing data to constituent parts and identifying emergent properties using a three stage process: open coding, axial coding and selective coding. *Open coding* is characterized by identifying concepts from the data. *Axial coding* organizes those concepts according to their relationships, and is oriented towards identifying structures or processes inherent in the categories. This allows the complex relationships to be identified which constitute the phenomena under study. *Selective coding* is a process of integrating this knowledge into a consistent theoretical understanding by producing a *central category* around which the other phenomena can be related and theory described. It is the central concept of a theory about the phenomena under study. The theoretical description tells the story of the data and provides a rationale for the phenomena that were observed. In this work, the purpose of using GT was to produce such a theoretical description of the emergent factors involving ideation and problem-solving for visualizations.

5.7. About Authorial Voice

In previous chapters, this text has centred upon the background and theoretical basis for the research. It has largely adopted the ostensibly 'neutral' tone usually used by the positivist approach to reporting scientific work (Kuhn, 1970). The following chapter describes the scientific work that comprised the bulk of this research, which involved collaboration of the participants and the author in an exploratory, empirical process. It is therefore important to mention the authorial voice used to report this work and in particular, the use of the first person. I have chosen to take the first person for reporting some parts of the case study precisely because of the method of inquiry. As part of this research relies upon a qualitative methodology, one that involves a researcher participating actively in the research setting, there are good reasons to depart from the discursive boundaries that are common to doctoral theses in reporting those results.

My rationale is twofold. The use of the first person prevents unnecessary wordiness in my reporting, often caused by the use of convoluted, third-person grammatical constructions (e.g. passive voice with an absent, implied subject) to lend a supposedly 'neutral' tone to the text. This approach is common to positivist scientific reporting. More importantly, it serves as a reminder that my presence and influence cannot be removed from the research setting, as third-person reporting implies. It is my intention that this approach should underscore my active participation in the research process and to emphasize that the results described here were obtained in a collaborative and co-creative process which was the consequence of mutual participation. The interpretation of the results of this collaboration arises, in part, from my own analysis of our design activities and from the observations of my collaborators as we engaged together in the design process. To remove myself from a report of this process would obscure the nature of my involvement and undermine the results of the research. This approach is in keeping with the philosophical stance of AR. Examples of using a first-person reporting approach in AR are in Baskerville (1999) and Whyte et al., (1991).

5.8. Summary

This chapter has described presented a research strategy for investigating the design of visualization software in a real world context, using Action Research and Grounded

Theory. This sets the stage for answering Research Questions 4 and 5. The next chapter describes the case study where this research method was applied, followed by an analysis of the research results.

Chapter 6: Qualitative Research

6.1. Introduction

This chapter reports the details of a case study which employed the SoViz method (Chapter 4) as an aid to the development of novel Information Visualization software. Using the research method devised to address Research Question 3 (Chapter 5), this case study was performed to investigate Research Question 4, regarding the practical outcomes of using SoViz. This chapter is organized according to the two-phase Action Research (AR) method that was used for the case study. Section 6.2 describes the Action Research conducted with the UCL Beacon Project. It discusses the two phases of Action Research undertaken, the Diagnostic Phase and the Therapeutic Phase. Both phases consist of several parts and this section of the chapter is divided accordingly. Following this parallel structure for reporting the research helps to demonstrate the way AR was used at each step in the process. Section 6.3 presents a Grounded Theoretical (GT) analysis of the design work resulting from the AR intervention, followed by a discussion of the theoretical knowledge generated from the research. The GT analysis of the design activities makes it possible to develop theory about the use of SoViz for visualization design, addressing Research Question 5.

The key findings from the AR intervention were that visualization designers benefited from taking a participatory design approach that incorporated the use of existing design knowledge with techniques that supported design ideation, creativity, and design problem-solving. Also, the participatory approach kept end-users actively engaged in the design process and put a high importance on their input, with the result that the organization was able to change and improve its design processes, and to produce a proposal for eight new visualization projects. The key finding from the GT analysis was that sketching and design patterns support Information Visualization design activities by facilitating the elaboration and reduction of alternatives at key stages of the design process. The use of sketching enhanced activities that relied upon exploring alternatives. Combined with the use of visualization design patterns, this allowed the group to benefit from existing Information Visualization design knowledge, and to propose a set of specific design solutions.

For the data analysis and interpretation of results in this chapter, transcriptions often provided the data source. Where quotations from the transcriptions are used as examples, the data source is identified according to the following convention: (Design session : line number).

6.2. Action Research Case Study: UCL Beacon Project

6.2.1. What is the Beacon Project?

The national UK Beacon initiative comprises a group of six computational biology (CompBio) research projects funded since 2002 by the UK Department of Trade and Industry (DTI) with a budget of over £8M. One of these projects, based at UCL, seeks to create an integrated computational model of liver processes that can '...fit together the different levels at which complex biological systems function, from genes, through to cells, and up to the whole organ and organism'. (DTI, 2006). For example, one part of the project seeks to model how calcium is metabolized within groups of liver cells, which can have important implications for understanding liver pathology and diseases such as diabetes.

The benefit of having a computational model, referred to as an *in silico* model, is that an accurate one would allow 'virtual experiments' to be performed and would facilitate easy exploration of liver function. There are many experiments that life scientists would like to perform that are simply not possible due to either practical or physical constraints. Such experiments are often labour-intensive, time-consuming, and expensive. Also, experiments on *in vivo* human livers, which would yield important understanding of liver processes, are simply not practical. Thus, an *in silico* liver which accurately models such processes would be useful to biologists who are studying this area. Creating an accurate computational model which can be used for these purposes is one of the goals of the Beacon Project.

The creation of such a model involves advanced mathematical knowledge that can accurately express how real liver cells behave in very specific circumstances. As metabolic processes occur, the activity among liver cells involves dozens of different biochemical signals and changes, which fluctuate over time. To describe this behaviour, very complex mathematical models must be constructed. These are so specialized that the Beacon Project has several staff members who are mathematicians. Because the models are complex, a user interface to the computer is required, but the ones in use by the Beacon team are geared to the expertise of mathematicians and are not easily understood by biologists.

During a preliminary meeting with the project lead, a programmer, and a mathematician, one topic of discussion was the reason the project team felt they needed a visualization tool. The team were not certain that one would be helpful, but they had seen compelling examples of visualization software in other domains. Also, the mathematicians were finding it increasingly difficult to have useful conversations about their mathematical models with other members of the Beacon Project who were not mathematicians. They thought that having a way to visualize the behaviour of the models and control their behaviour interactively would help the other project members give useful feedback, particularly on how to change parameters of the models to more accurately reflect *in vivo* activity of liver cells. They had reviewed other graphics and modelling packages, such as those in MatLab, XPP, and Mathematica, but had decided that these would not offer the capabilities they were seeking. They were hopeful that with some assistance they would be able to design a visualization tool that would meet the needs of the project.

6.2.2. The participants

There were nine participants on the Beacon visualization project who contributed in some way to this research. Because of the nature of Action Research, in which each of the participants brings a unique world-view to the research setting, it is important to describe their roles on the project. For the sake of brevity, participants are referred to by the first letter of their name:

- B, this researcher, i.e. the author of this thesis. A doctoral student studying Information Visualization design.
- P, the project lead. The leader in terms of programming and system architecture; also my principal contact for the visualization project.
- O, a programmer, assists in coding the various parts of the models and the model management system.
- AF, a senior computer scientist and administrator, who is not responsible for programming but provides senior-level guidance and logistics support.
- L, a senior mathematician responsible for model development and senior-level guidance on mathematical issues.
- J, a mathematician, primarily responsible for creating the models that are used by the project.
- AW, the principal investigator, a senior biologist and the person at the most senior level of the project.
- M, a biologist, responsible for coordinating and conducting the laboratory experiments that yield the data that the model designers use.
- S, a biologist, responsible for conducting the experiments that yield the data that the model designers use.

The group can be divided into two categories: those who work on the project every day, and those who have a vested interest but are at a more senior level and who are not involved in the daily work in a direct way, though they do participate in the strategic planning. Each of these people participated in the requirements-gathering interviews, but not all took part in the design sessions. Although each person is a stakeholder, for the design phases it was decided that the most value could be provided by people who would actually use the software, i.e. those who are involved in day-to-day design and testing the software and mathematical models, or in running the biological experiments that generate raw data. Along with researcher B, these people, P, J, M, O, and S, were the primary participants of the design sessions.

6.2.3. Formal research agreement

In accordance with the openness of the AR approach, at the beginning of the first design session meetings, I described the Action Research process and ensured that the participants understood its method. The participants received a brief, typed AR summary of how AR was relevant to developing the Beacon Project visualization software, and what was expected from team members. I explained my objectives for the research, how I expected them to benefit, and emphasized that team members were free to withdraw at any time. I described how, according to the AR approach, the researcher is meant to be an active participant. We also agreed that we might determine if the problem the team were facing was not a need for Information Visualization software, and if so, whether we could consider changing or discontinuing the research. I answered questions about the AR process. After discussing these things, all participants agreed verbally that they were willing to continue with the research.

6.2.4. Diagnostic phase: Requirements-gathering

The first major part of the AR process used in this research was the Diagnostic Phase. This involved collecting information the needs for organizational change through requirements gathering interviews, formulating a theoretical problem statement, and planning the intervention. As a ground-premise, the shared understanding of the organizational problem was that the Beacon Project team were seeking to develop an Information Visualization tool using the intervention of a subject-matter expert to assist in this process. Expressed in terms of Action Research, the organizational change they desired was: to move from ignorance about visualization design to an understanding about it which could be used by the team for future work. The requirements gathering proceeded on this basis.

Diagnosing the requirements for the organization served the dual purpose of identifying and refining the organizational needs in terms of AR and the design needs for a visualization. The organizational needs centred on desires of team members for learning about visualization design. The visualization needs pertained to the types of computational biology tasks the visualization should support and what such software might look like. Some of these requirements were captured through conversations at a preliminary meeting with the project members. Additional, general opinions were gathered during a weekly project meeting where team members addressed the objectives for the visualization as a group. P, the project leader, had previously identified what he believed to be the requirements of the software and formulated them into a document. However, these ideas represented only his synthesis, based on his experiences of working on the project. The document did not capture the separate and independent opinions of individual members about how the tool could be useful. Since people tend to express their opinions differently when they are in groups, I requested a meeting with each staff member privately, expecting that each would be able to express more clearly their individual opinions and desires for the project.

After obtaining agreement from the team to go ahead with the project, we arranged for the individual interviews, which were conducted over a two week period. The interviewees were from all parts of the project. They were the principal investigators AW and AF, two mathematicians L and J, two programmers O and M, and the project coordinator P, and two of the life scientists who perform bench-biology experiments that generate new raw data. Eight one-on-one interviews took place. Each lasted approximately an hour to an hour-and-a-half, during which individual project members expressed opinions and desires for the project; these were collected on an electronic audio recorder. During the interviews, I encouraged participants to try to draw out what they thought things might look like in a visualization. This was also a useful way to document the kinds of existing visual metaphors and descriptions that people were already using. In the course of their professional/academic practice, practitioners tend to develop a common visual vocabulary for describing the processes that they are studying. Examples in architecture (Robbins, 1997), graphic (Lidwell, et al., 2003), and software design (Mullet and Sano, 1995) are well documented. I therefore encouraged the participants to sketch their ideas so that this might elicit preconceived representations that each of them held. These visual elements could then be used for the SoViz design work.

After the interviews, the recordings were transcribed and the different requirements of participants were noted. These were then aggregated into a comprehensive list and compared to a list of requirements which had been provided by the project lead, P. There were only a few duplications. With the aid of the P, these were edited into a master requirements document which was circulated by email to the project team, in preparation for the design sessions. These requirements were used to propose a theoretical problem statement as a starting point for the design activities.

6.2.5. Diagnostic phase: Theoretical problem statement

Devising a theoretical problem statement is important in Baskerville's AR method for information systems. It acts as the premise which validates the Action Research and provides structure for the intervention. It also encapsulates the research domain and gives focus to the problems that will be addressed. Thus, it is important to have it written down for everyone to see and to agree upon. To achieve this, I devised a research question for the formal research agreement. During the first design sessions, participants took some time to examine the statement and to suggest any modifications to it. I also suggested that we might decide to change the problem statement at some future date, if the research so indicated. The theoretical problem statement for the Beacon Project was:

> Subject to your agreement, in our case, the change we are hoping to achieve is: *to design an information visualisation tool for the Beacon Project that will make use of Sketching and Design Patterns as aids to the design process.* [italics in original]

All of the participants agreed that the theoretical problem statement described in the formal research agreement matched their conceptualizations of the problem at hand. They also agreed to continue to pursue the research according to the AR approach and to use SoViz as a way to try to address the design problems they faced.

6.2.6. Diagnostic phase: Action planning

After agreeing to the AR method, the group made decisions about how to proceed. We planned to use information that had been gathered in the interviews, and to hold an indeterminate number of design sessions wherein we would explore visualization design alternatives. As the research was exploratory in nature, we did not know what to expect from the design activities. Nor was it obvious how many design sessions might have been required to either determine workable solutions or to abandon the research. As it was one of our research goals, we planned to use sketches to assist in the design process and to use the visualization design patterns as a way of helping us in our decision-making process, although team members were not yet familiar with them. We planned to then discuss and evaluate these design activities later to try to understand what was learned during the process.

As with other methods of software development, the first step of the SoViz approach is the requirements gathering process. After the requirements have been collected and understood, decisions can be made about the high-level design of the software. Also, any differences among the expectations of different team members can be identified and if possible, resolved. These requirements then form the basis from which the design activities can begin.

As Action Research requires collaboration of the researcher and the participants in the research process, the group discussed how to collect data for later use. After discussion, we agreed to use audio recordings and sketches. [The sketches produced are included on the accompanying CD Appendix.] Although we did not discuss using video until later, we decided that the use of a camera would have proved cumbersome and intrusive to the design process. The information that would have been gained by using video did not appear to outweigh intrusiveness of setting up and operating the camera, nor the distraction of its presence. We agreed to use pencils and paper and an unobtrusive audio recording device. I agreed to transcribe the design sessions for later analysis and to collect and scan any sketches.

The sketching tools consisted of a large (A3) cartridge pad, as would be used for figure drawing or watercolour painting. A selection of coloured pencils, some drawing pencils of varying hardness, and an eraser were also used. The rationale for this was to try to avoid any electronic devices so that the tools themselves would not become a barrier to trying to sketch out ideas. For ease of use, the design patterns were printed in colour onto A4-sized laminated plastic card. They could be shared conveniently among team members, stacked into categories, or set aside for later use.

Using these materials, the design sessions were carried out in a small meeting room on the premises of the Beacon Project. Meetings were conducted around a group of tables which were positioned to allow everybody access to the sketching materials. The largest group participating in a single session consisted of five people.

6.2.7. Therapeutic phase: Action taking

The second major part of the AR process used in this research is the Therapeutic Phase, which entails action taking, evaluation, and specifying learning. The Action Taking was done with the SoViz approach, using sketching and the visualization design patterns in four separate design sessions, the agenda for which was agreed collaboratively. After these sessions, there followed other informal discussions and a requirement specification session which outlined the final requirements that were the result of the design sessions. These sessions are described below.

The first two design session meetings were conducted with separate stakeholder groups. The mathematicians and biologists met me separately to describe what they had in mind for the visualization tool and to begin to use the sketching tools and design patterns. At the middle of Design Session 2, the two groups joined up for a larger, collaborative design activity. The following Design Sessions 3 and 4 were conducted only with the mathematicians and programmers, O, P, J, and O, P. We recognized that this was not ideal because we did not want to exclude the biologists from the design process, but for practical reasons, it was the best compromise that we could achieve.

Following the four design sessions, one final requirements specification meeting was held to define possible visualization projects resulting from the design activities. The result of this meeting was a requirements specification document outlining eight visualization tools at a high level (Appendix D). It was recognized that the team could not undertake to engineer more than one of these visualizations, so the team selected a single visualization as the candidate for further development. The details of the development of this visualization are discussed in Chapter 7. The design sessions and requirements specification meeting are summarized below:

Design Session 1 (20/9/05, duration 02:50:17): Getting started

Attendees: O, P, J, B.

Summary: This session was attended by me, two software engineers, and a mathematician. I explained the Action Research process to them and obtained their formal consent. I had prepared an agenda describing what we were going to do during the meeting. We generated ideas using sketching and collaboration. I showed the visualization patterns to members of the team. The software engineers frequently discussed solutions to design problems. I tried to point out where their solutions were described by one of the patterns.

Outcomes: The outcomes were that we agreed to conduct a further meeting with the biologists.

Decisions: Meet again for another session.

Design Session 2 (10/10/05, duration 2:03:08): Understanding biologists' needs

Attendees: M, S, P, J, B.

Summary: During this meeting, I met first the biologists to avoid reiterating the Action Research process which I had explained in Design Session 1 to the rest of the group. I explained the AR process to the biologists and obtained their consent to continue with this research approach. Then we began to discuss the requirements document that I had put together as a result of Design Session 1.

Outcomes: Several sketches were generated with preliminary ideas for how to build a visualization system. The biologists confirmed that the best representations would be in visual artefacts that they were familiar with, e.g. tables and graphs.

Decisions: Meet again for another session. It was decided that the team would not actively try to learn all of the patterns and would instead rely on me to introduce them when they would seem appropriate to a design problem.

Design Session 3 (4/10/05, duration 1:42:48): Exploration of patterns

Attendees: O, P, J, B.

Summary: The topic of discussion was how to further build the models from subcomponents and what visual form the models would take.

Outcomes: After this meeting, a document was circulated which specified the design requirements by pattern.

Decisions: Meet again for another design session. P to list the visualization requirements captured so far for everyone to review in Design Session 4.

Design Session 4 (31/10/05, duration 1:36:34): Bringing it all together

Attendees: P, J, B.

Summary: The requirements list generated by P was reviewed. The items were categorised by difficulty of implementation. The production of these resulting visualization components was discussed. A decision was taken to compare the ideas generated so far with the visualization patterns and to note which patterns had been identified already and which had not. We also decided which patterns would not be useful for the visualization.

Decisions: B and P to meet again to discuss the final design requirements of the prototypes so that they can be put into production.

Requirements Specification Meeting (13/10/05, duration 1:30:40)

Attendees: P, B

Summary: The topic of discussion was the effect of using patterns and sketching in the design process, plans for future work on the project, the fact that P's time remaining on the project was drawing to a close. We discussed the list of design requirements by pattern from Session 3 and P agreed to produce more specific descriptions of projects.

Outcomes: Plan to possibly have one more design meeting.

Decisions: Write up the requirements as a requirements specification document so that the management can provide the labour to write the prototypes.

6.2.8. Therapeutic phase: Evaluating actions taken

As with all parts of the Action Research process, evaluation is meant to be a collaborative process to study the effects of the intervention and whether they were positive or negative. Baskerville (1999) notes that this entails: (1) determining whether the theoretical effects occurred and (2) whether the AR intervention was the sole cause of the effects.

In this area, the team's efforts were somewhat weak because the group was more focused on solving design problems than evaluating the effect of the intervention. Therefore, the evaluation was generally ad-hoc and occurred informally, throughout the AR intervention, but such discussion was generally prompted. The Beacon team evaluated the effects of SoViz by reviewing the design sketches, talking about decisions, and engaging in discussions about the design process. This occurred near the end of Design Sessions 3 and 4 and in a follow-up meeting. On several occasions, the team discussed the design patterns, the sketches, or the SoViz process and remarked upon their usefulness for making design decisions. Here is an example of team self-evaluation from Design Session 4, regarding using sketching:

Design Session 4 (4:617-620)

617 B: So let me ask you this: if you're used to thinking in that way, and not visually, then in what way – if any way – did using this drawing part, what way was that helpful in this [SoViz] process?
618 J: It was really helpful...
[Remarks omitted]
620 J: ...when you made me write it down, I was forced to draw it like this. And this idea would never have come to me, if you hadn't made me do that.

Their reflection about the design work also led team members to propose topics for future research:

Design Session 3 (3:619)

619 J: I would be interested to know whether or not you would find, if you did this systematically, any significant added value in using some more sophisticated technology than a piece of paper and a pencil; something where you could see the dynamic evolution of the drawings recorded and be able to match the timing of events on the piece of paper to the timing of events in the microphone.

The evaluation of the intervention also occurred informally in the requirements specification meeting, via personal communications, and emails.

The result of the evaluation is an interpretation that the effects specified in the theoretical problem statement were the result of the intervention. The Beacon team has learned a beneficial method which they believe will be effective for future addressing future visualization problems. This effect would not have occurred without the SoViz intervention, which was the primary causal factor. No other organisational effects were identified by the team which would have contributed to this interpretation.

6.2.9. Therapeutic phase: Specifying learning

The final stage of the Action Research methodology is *Specifying Learning.* Reporting findings in AR is often in the form of anecdotes or quotations that describe the AR interventions and their results, an approach that is adopted here. After completion of the four design sessions, the data, in the form of the sketches and the audio were analysed. To aid analysis, some of these data were presented to the Beacon team members in follow-up interviews. In many instances, verbatim transcripts of the design sessions were used to gain feedback from team members about the nature of the design process and about how sketching, the design patterns, and the activity of the researcher as a subject-matter expert (SME), were useful. The findings comprise the learning that resulted from the Action Research in the following areas: (1) the effects of sketching, (2) visualization design patterns, (3) the effects of the presence of a subject-matter expert (i.e. this researcher) in the visualization design process, (4) a list of the projects that resulted from the design process. Details of each of these and some examples from the transcripts are presented below. These data illustrate what was learned from the AR intervention and provide a partial answer to Research Question 4, regarding the practical effects of using SoViz. Further insight into this question is provided in Chapter 7, which reports the implementation of one of the proposed visualization designs.

Specifying learning: sketching

Sketching was particularly effective for helping participants to move ideas from their internal thought processes to a public space where they could be explored and modified. While many participants on the project had privately arrived at good ideas about solutions to specific problems and had perhaps encountered novel solutions in their research, these ideas tended to remain internalized. They had not yet shared them with one another on the project. This excerpt from Design Session 4 serves to illustrate this point:

Design Session 4 (4:627-633)

627 B: In other words, you take the ideas out of your head and put them somewhere so that you can look at them and act on them.
628 P: Yeah. And then you can stand on them and get higher.
629 J: And I don't think you would have come up with the idea of making that disappear when you put something into that, if I hadn't –
630 P: No. You can't see it quite so much. I don't see pictures or words in my head, I see abstract concepts.
631 B: You wouldn't have come up with the idea of making this triangle shape disappear without having it drawn on the page.
632 P: Yeah.
633 J: Yeah.

This passage shows how sketching allowed individual ideas to become part of the collaborative problem-solving process. Sketching ideas together facilitated design decisions and allowed team members to modify solutions to more tightly address project needs and user tasks, as illustrated by J's remarks in 4:629. Moreover, the participants reported that sketching enhanced their effectiveness in a way that was not possible through verbal dialogue alone. Sketching had prompted changes in thinking about and exploring design problems which would not have otherwise occurred.

Although the Action Research process was not meant to uncover evidence of the benefits of sketching as described in Chapter 4, the team members offered many subjective opinions about the benefits and drawbacks of using both tools. Generalizing, the participants' opinions concurred with the literature on how sketching supports ideation, creativity, and problem-solving. Three examples serve to show how participants felt that sketching was useful to them in these ways. In the first example, J suggests that sketching supported creating a new idea:

Design Session 4 (4:617-622):

617 B: So let me ask you this: if you're used to thinking in that way, and not visually, then in what way – if any way – did using this drawing part, what way was that helpful in this process?

618 J: It was *really* helpful. Let's take an example from today. I was going to describe this bit –

619 B: The puzzle piece.

620 J: – in words. I was thinking of it mainly algorithmically. And I was thinking of it in terms of the maths that you would need to do to wrap the RC interface into the WR [Waveform Relaxation] interface. And then when you made me write it down, I was forced to draw it like this. And this idea would never have come to me, if you hadn't made me do that.

621 B: So you actually had a new idea as a result of –

622 J: And I would not have done that if you had not said, 'No, [J], draw it'.

In the second example, the design team were able to use sketching for creative purposes. This is an example of using an abstract visual metaphor to constrain user input. One visualization idea involved allowing non-mathematicians to construct mathematical models from sub-components. However, the mathematical models in the software have numerous parameters that only interact in certain ways. The team were looking for a way to force users to construct only valid models out of constituent sub-components but needed to allow users to match interfaces among different models. Through sketching out this design problem, they were able to arrive at a jigsaw-puzzle metaphor that would constrain the user's interaction possibilities. Figure 6.1 shows this novel representation, which was created by the participants early in the course of exploring this problem.

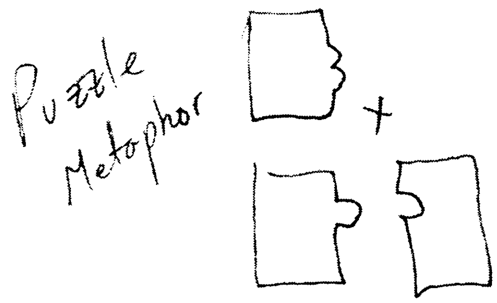


Fig. 6.1: Creating a visual metaphor for software behaviour.

In the third example, J explains that the design team were able to use sketching to solve complex problems in order to create more advanced features:

Design Session 3 (3:643):

643 J: But in terms of how this has also benefited us, I'd say that it's allowed us to develop more advanced features. Because you say, 'Oh what we had last time is this and we can build that'. And it's much easier to express an advanced functionality when you've got written down the basic functionality. Otherwise you have to – you're building some abstract concept within someone's head and you have to explain to them the bits that you've got so far before you can add on the pieces at the end. Whereas, if you can point to it on the piece of paper and say, 'This is what we've got so far. Now what we're going to add is that'. then you don't have to build such a large abstract concept in a person's head.

These examples illustrate how sketching was useful for the design team. However, sketches also presented occasional difficulties. In this example, the participants identified one of the key drawbacks to using sketching in the context of designing for dynamic interfaces: that sketches are static, whereas a visualization is, by definition, dynamic:

Design Session 3 (3:625-626):

625 J: See, I find that what we've been trying is to draw computer screens and dynamical evolution of computer screens on pieces of paper and I find that quite difficult.

626 P: Yeah, that's true. That's why we've got all the stupid things where it goes like that. And the box like that. And this thing with the person with the brain looking at the two bits of information.

Additionally, there were mixed opinions about the use of sketches as an historical record. One drawback was that although sketches were a useful reference during design time, it was sometimes difficult to interpret their meaning after extended periods of time. Compare the following opposite opinions regarding this, in excerpts from design sessions 3 and 4:

Design Session 3 (3:605-606)

605 B: You didn't talk about the drawings that we did. What did you think of those?
606 P: I think that those were amazingly helpful because they form a record of what we did.

Design Session 4 (4:280-282):

280 B: The question is, 'Do the drawings in the longer term, have some utility as well? Can we still talk about them outside that original context? Do they still carry the same meaning that they did?' Or are they merely a tool at the time and once we've done them, we've dispensed with them. And if you can't even identify what you were talking about in a drawing, then that would support the hypothesis that –
281 J: I don't know what that one was!
282 P: Oh, those are just sliders!
283 J: I know, but it's a rectangle with some lines on it.

These examples illustrate many of the benefits and some of the drawbacks of using sketching. It aided communication by facilitating the sharing of ideas, supported design ideation and creativity and aided problem-solving. By contrast, design patterns proved more difficult to use.

Specifying learning: design patterns

Though the sketching was occasionally challenging, the design team felt that design patterns were also occasionally useful. Their opinions, expressed in Design Session 2, were offered after the decision had been taken to avoid examining the patterns exhaustively until a later design session. The opinions identify two problems with the patterns: (1) it was generally agreed that they were considered to be difficult to understand, and (2) the examples used in the patterns were not familiar to the team, which made it difficult for team members to relate the solutions to past experiences. The following excerpts demonstrate these problems:

Design Session 2 (2:701-704):

701 M: The last time that I read this [design pattern] I don't understand anything.
702 B: Uh-huh.
703 M: Really.
704 B: Yeah.

Design Session 2 (2:707-711):

707 J: These are not very readable.
708 B: Okay.
709 J: To my mind they're not written in enough, like, – for many of them, they're standard things that you, they're just reminders that, "Oh, yes, somebody used that at that part of the PowerPoint interface and I've used that. You know. Oh, look, it's a list of little icons down the side that I can click on to change the mode context of the thing, for example, to pick a PowerPoint example. I find these have not got the right, Oh this really worked with the mouse in that piece of software.

710 B: So you need an abstraction that communicates the core idea?
711 J: I think that what it really needs is example, for those ones that are familiar from many of the standard pieces of software, it needs a reminder from software that we are familiar with. I am sure there are some that I'll go, Wow, that's a cool idea! and I won't have come across any bits of software that have used that thing.

Design Session 3 (3:153):

153 J: But right, now some of these, I think. Let's get a bit of paper for these [design patterns] and when one of us manages to translate it from this language to something we understand, we'll make a note.

Design Session 2 (2:738-741):

738 J: But from the point of view of your research, I think that it would be great to have some great flash cards or something with the standard patterns on, but these are not they. Because, let's just look at this one. That is picture-free and the example -
739 B: Is a citation.
740 J: - for 'Smooth Transitions' is Brath, 1999 *not:* 'You know, like in the Mac OS dock'. Which, if it said *that* -
741 B: Would be much better.

Although they were difficult to understand and use, one benefit of the patterns is that they are a consistently organized source of visualization design knowledge. For much of Design Session 3 (3:152-3:686), the team decided to use the patterns in a process of elimination, to ensure that well-known design solutions which are captured by the patterns had not been overlooked. A review of the patterns was made, and one team member noted whether the pattern had been already applied to design problems ('Thought of it'). In this way, it was decided whether a pattern was applicable or not and whether it should be used ('Plan to use it') for any of the design problems the team had encountered. The team also decided that some of the patterns were too generic to be of interest to the design problems. Table 6.1 shows the list of these decisions.

The table shows several examples of patterns which the team did not think of and decided would be useful for visualization problems. For example, J suggested that the reference context pattern would be useful for representing graph structures in the computational models. In this case, the example illustration in the pattern was useful for inspiring a solution to a problem:

Design Session 3 (3:201-223):

220 J: That's right. For us, this would be something – that's interesting.
221 P: It's a cool picture actually. You can imagine how that would be quite useful to us.
222 J: I'm thinking about it in a different way. We've been talking about our graph structures being presented on the screen and just letting a crossing-minimisation algorithm arrange them on the screen. But instead you could arrange the nodes on the graph - there will be more crossings here - but you could arrange the nodes on a graph according to some piece of information about the nodes.
223 P: You mean for presenting, for example, the topology of a system like, this.

224 J: Exactly. So, we normally draw our biological connection diagrams already do this subconsciously. I hadn't though t of this. It's a good idea. We always put the higher things up in the signal processing pathway, nearer the top of the screen.

In another example (3:337), P suggests how the SDS + keyboard pattern might be useful for the large number of parameters that the computational models use:

Design Session 3 (3:337):

337 P: We've just done SDS + keyboard and I think we said that we like it. We're probably going to want to use that for selecting parameters and stuff aren't we?

Later, P suggests that categorization of the patterns shows how they are useful for the design process. Recognizing that many user interface actions were already familiar, he suggests that the patterns are helpful because they present ideas that the team did not think of:

Design Session 3 (3:434-436):

434 P: I think the fact that there are some in the 'We didn't think of that and it's useful' column at all – even if it's just a handful in there – it proves that the patterns are useful.

435 J: Yeah.

436 P: Because some quite behaviour we've been able to come up with off the top of our heads, but not all of them. So even when we were talking about the problem quite deeply, there were still patterns which are useful to us and really provide something we didn't think of.

TABLE 6.1: THE BEACON TEAM'S DECISION CHECKLIST REGARDING THE VISUALIZATION PATTERNS. 1 = "YES" AND 0 = "NO". N = "TOO GENERIC".

	Thought of it	Plan to use it		Thought of it	Plan to use it
1. Visualization	0	N	19. Navigation	1	1
2. Appropriate visual objects	1	1	20. Level of detail	1	1
3. Familiar organizational device	1	1	21. Spatial navigation	1	N
4. Non-familiar organizational device	0	1	22. Dynamic queries	1	1
5. 2D representation	1	1	23. Direct manipulation	1	1
6. 3D representation	1	0	24. Single direct selection	1	1
7. Reference context	0	1	25. Multiple direct selection	0	1
8. Redundant encoding	0	1	26. Bounding box	0	1
9. Smooth transitions	1	1	27. Single direct selection + keyboard	0	1
10. Datatips	0	1	28. Bounding box + keyboard	0	1
11. Small multiples	0	1	29. Context maintained filter	1	1
12. Legends	0	1	30. Reduction filter	1	1
13. Visual separation	0	1	31. 2D navigational model	1	1
14. Overview and detail	1	1	32. Click-n-drag	1	1
15. Filter	1	1	33. 3D navigational model	0	0
16. Details on demand	1	1	34. NAFS model	0	0
17. Interaction	0	N	35. Teleportation	0	0
18. Selection	1	1	36. Navigation box	0	1

By the end of the design session, participants were starting to use the patterns to formulate ideas and to recognize where they have already been applied and to see the value of them in helping to solve design problems, as these two excerpts show:

Design Session 3 (3:452):

452 J: 'Teleportation' plus 'Smooth transitions' is good. That's Google Earth.

Design Session 3 (3:543-544):

543 P: But I do think that the depth to which we're beginning to understand this problem and the domain and the way the different people who will use the tool – is putting us in a good position to think of those 'Non-familiar organization devices'.

544 J: You see, I think we need one of those in order to solve the problem that I want to solve which is, to allow that structure of models which is normally encapsulated in equations, to be understood by biologists.

This first excerpt also lends support to the notion that the patterns could be built into a pattern language, one of Wilkins' objectives for them. However, because of time constraints, the team did not choose to pursue this possibility further.

These examples demonstrate that patterns were useful for inspiring design ideas and solving design problems. The team initially found them difficult to understand because they were very text-rich and contained unfamiliar examples. However, after having uncovered some of the design challenges using sketching, a group review of the patterns was helpful.

Specifying learning: effect of the subject-matter expert

As discussed in Chapter 5, both research methods used in this work embrace the researcher as an integral part of the research context. For this reason, it is important to note the effects of the researcher as a subject-matter expert in the domain of Information Visualization. A way of assessing the impact of the SME is by examining the kinds of knowledge given and the timing of the aid provided.

In this study, in addition to being co-researcher with the other participants, I provided the primary means of expertise in Information Visualization. When obvious examples of helpful techniques presented themselves, I indicated that such techniques might be useful. Three examples characterize this kind of help:

Design Session 1 (1:193-195):

193 B: Well, and one of the key things – if you're making comparisons, you want to make them visually at the same time, so maybe you want v1, v2, and v3 all visible at once.

194 P: Okay, yeah, you may want three variables visible at once, but you also have to be able to cope with the fact that you may have 99 variables.

195 B: Well, there are solutions for that problem.

Design Session 2 (2:113):

642 B: The whole idea behind a visualization is that it's interactive. It's not just an animation like you see on PowerPoint at a presentation. But it's interactive, so that you can engage with what's being shown on the screen and it will tell you something that you didn't know before.

Design Session 2 (1:635-641):

635 B: This is standard technique called 'Parallel coordinates' plot.
636 J: Super. That's what I was hoping you'd say.
637 B: Yeah.
638 P: it's kind of orthogonal to our main diagram isn't it, but.
639 B: Well, this is very useful for comparing large numbers of variables simultaneously across many parameters. I mean, this is exactly what it looks like, actually. You've drawn it beautifully. One of the examples that I've seen this done in – it's much more prosaic – is the national automobile insurance association's lists of parameters of American automobiles from 1975 to 1982 or 1971 to 1980 or something like that. It has the car, the gross weight, the horsepower, the fuel consumption, the displacement, all of these different parameters. And because of the patterns that are inherent in that data, when you plot them vertically like this, you start to see this cat's cradle.
640 J: Yeah.
641 B: And that, in itself, it's just a useful way to plot the data because you can see it all at once. But you can also see relationships that were not inherent in say, a tabular representation.

In addition to providing knowledge from the domain of visualization I also provided explanations of the visualization design patterns as a means of reducing the learning curve associated with them and to demonstrate how the patterns offered solutions which were applicable to design problems at hand. The two examples are representative of the many occurrences of this:

Design Session 3 (3:163-165) (reducing learning curve):

163 B: The 'Context maintained filter' would be like, let's say you have a star-field of just points. And you have a slider that's a filter control. You bring the slider up and as the threshold increases, some of those points drop out, but the overall appearance of the star points stays the same.
164 J: But according to this, they don't just have to drop out. For example, a box can be placed around all matching objects.
165 B: Yes. Like a colour. Or like, you bring the filter up and as the delta increases you get more points of that colour.

Design Session 1 (1:162-164) (applying a pattern to a problem):

162 B: Okay? Or for other kinds of variables, that we can attach. For example, one of the things that came up over and over as a requirement is that - and you guys know this already but I'm going to reiterate – is that in the experimentation or in the process that's being modelled, there's this highly compact area of activity in terms of time, which gradually gets slower and slower and slower over, say 48 hours. So

when the biologists are doing their experiments, they'll take all of these samples at the beginning and then they'll take samples at two hours and six hours and eight hours and whatever. So this activity occurs in this really compressed timeframe. Well, one of the solutions for that kind of a problem in visualization is to allow this dynamic representation of a continuum of time, such that you can go in and look at something very specifically that's occurring in a very short period of time whilst maintaining the *context* of the entire timeframe that's being assessed, okay?

163 J: Yeah.

164 B: So you have an overview, as well as a detail. And that, in fact, is one of the patterns.

Notably, this help was not as useful during Design Sessions 1 and 2, because it was primarily in the form of pointing out when specific design patterns described a solution that the group had just created. During Design Session 2, the participants wanted to continue with this approach, because it was deemed easier than trying to digest the entire set of patterns. However, by the end of Design Session 3, after the team had created numerous design solutions, they wanted to compare their solutions with the set of patterns, resulting in the checklist in Table 6.1, above. This evaluation and logging activity occurred at the end of Design Session 3 and accounts for the majority of instances (11) where I provided expertise (Table 6.2).

In terms of timing, expert assistance was helpful throughout the design process. However, by the end of Session 4, the amount of aid that I continued to provide was declining, because it was less necessary. As none of the team members was particularly skilled in sketching, and were completely unfamiliar with the design patterns, they were initially apprehensive and uncertain as to how the tools would help them to arrive at design solutions. Early in the design activity, I was more likely to need to prompt participants to the drawing pad and pencils or provide knowledge about the patterns. In contrast, by the end of the design work participants used sketching, and occasionally design patterns, without any encouragement. Their apprehension about using both tools had disappeared. Team members were eager to use the tools to help facilitate design activity. In informal discussion and in conversations during the requirements specification meeting, participants agreed that they would not have expected this result and that they found it extremely fruitful. This would support an interpretation that the team were

TABLE 6.2: INSTANCES OF EXPERTISE PROVIDED DURING THE DESIGN SESSIONS.

	DESIGN SESSION			
	1	2	3	4
Information Visualization domain expertise	6	1	3	2
Visualization pattern expertise	5	2	11	0

becoming familiar with the concepts in the visualization domain, as might be expected. This was one of the objectives of organizational change from the Action Research intervention, which was articulated during the diagnostic phase. This, along with subjective opinions of participants that the intervention was successful, supports an interpretation that the AR intervention was successful for the organization.

Specifying learning: Visualization sub-projects

A final outcome of the specifying learning phase of the research was a high-level requirements specification document (Appendix D) which provides a description of the different visualizations that had been created in the design sessions. In terms of the Action Research method, this document is evidence of the learning that occurred during our use of SoViz. We recognised that a complete system encompassing all of the design ideas was beyond the scope of the Beacon project, but that many of the ideas could be chunked into manageable visualization sub-projects which could be completed by in-house programmers or by graduate students in the Computer Science department at UCL.

The fourth design session and the requirements specification meeting included a discussion of the software projects that should be produced and prototyped. During these meetings, the group synthesized knowledge gained during the sketching and design activities and produced the requirements document. Whereas we had begun the design sessions with no clear solutions in mind, by the end of the process, we were able to articulate the tasks the Information Visualization software should support, many visual representations which should appear on-screen, the interactions which should be possible and how this would support end-users' work processes in computational biology. Using the design sketches, descriptions of the proposed visualizations and personal notes, P then produced the document. It describes the goals of each design concept and estimates the time required to complete each visualization. Its purpose was to capture the design ideas, to report them back to the rest of the Beacon Project, and to aid decision-making about which of the concepts might be prototyped and evaluated. The full high-level descriptions of these projects are in Appendix D. In brief, the eight visualization sub-projects were:

1. **Provenance slider** – an interactive display of the computational biology literature which supports parameter values;

2. **Graph distribution visualization** – a graph viewing visualization which shows composite models and their relationships;

findings substantially addressed Research Question 4, regarding the practical outcomes of using SoViz. Further research regarding these effects is presented in Chapter 7, which describes the development of a visualization prototype. Before addressing this, the next section reports the theoretical findings about the AR intervention.

6.3. Grounded Theoretical Analysis of the SoViz Design Process

The purpose of the GT analysis was to address Research Question 5 regarding the theoretical effects of using SoViz. Whereas the AR approach provided a method to evaluate the practical use of SoViz framework, a grounded theoretical approach was used to further analyse the sketches and audio recordings resulting from these activities. This additional analysis contributes to a fuller picture of the SoViz intervention and can contribute to theory about the effects of supporting the visualization design process with sketching, design patterns, and expert knowledge. The results of the GT analysis are reported below.

6.3.1. Method

The GT analysis was conducted according to the methods described by Strauss and Corbin (1998). The total duration of the design sessions was 8 hours, 14 minutes, with an average session duration of 2 hours, 3 minutes. Approximately eight hours of audio recordings of the four design sessions were transcribed. Sketches in the form of 17 A3 pages were scanned and converted to digital files. These data items, the primary data set, were imported into the Atlas.ti software system for further analysis. As reported by Straus and Corbin, this software is commonly used for GT data analysis. Following the methods for coding described in Chapter 5, the conversations were analysed and then coded. To reduce ambiguity and for consistency of coding, each of the open codes was clearly defined. The open codes and definitions are presented in Appendix E. [The fully coded transcriptions are included on the accompanying CD Appendix.] The number of instances of each code is referred to as its density. These open codes or *categories*, were then aggregated to identify their *sub-categories* during the axial coding process, which links categories at the level of their underlying *processes* and *dimensions*. These were then used to guide the development of a *central category*. When no further analysis will reveal further instances, the category is said to have reached saturation. The central category is the result of theory development about the phenomena surrounding the use of the SoViz approach by the Beacon Project design team.

6.3.2. Results

The key finding of the analysis, expressed as a central category, was that sketching and design patterns support Information Visualization design activities by facilitating the elaboration and reduction of alternatives at key stages of the design process. The central category describes a relationship between sketching, design patterns, and the design process which can be interpreted in terms of Buxton's model, described in Chapter 4 (Figure 4.1). The discussion of sketching in Chapter 4 presented Buxton's analysis of the overlapping funnels of the design process, which consists of activities involving elaboration of design alternatives, followed by reduction of alternatives. Whilst sketching was useful for the opportunity-seeking, elaborative activities at the early stages of design activity, design patterns proved most useful in the reductive, decision-making phases towards the end of the design activities. Whilst the elaborative activities involve opportunity seeking and exploration, the reductive activities tend to require decision-making and elimination of alternatives so that a final product can be produced.

Extending this design model to visualization, a new theoretical model of Information Visualization design (Figure 6.2) emphasizes the importance of sketching for the early stages of problem solving and exploration and situates design patterns at the later stages of the process. This interpretation of results is based on two factors: the times during the design process that the team was able to use these tools (Availability), and the relationships among the categories of activities surrounding the tools (Relationships).

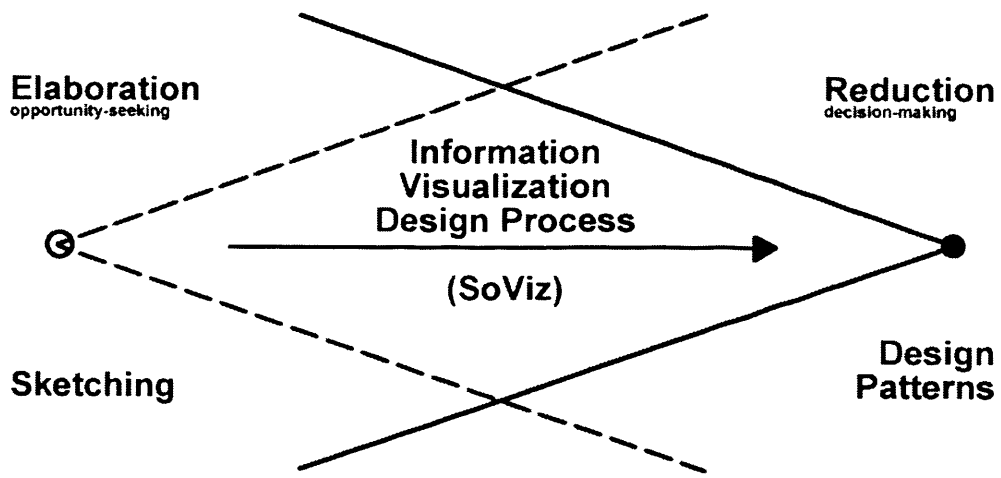


Fig. 6.2: The SoViz framework as it relates to the design process.

6.3.3. Availability

Timing played a significant role in how sketching and design patterns were used. The majority of categories representing processes of elaboration and exploration were those related to sketching activities. By contrast, a larger number of categories relating to decision-making and reduction of alternatives involved the use of the design patterns. Three factors are responsible for this. The first is that sketching was an activity that was immediately available to the design team. The familiarity of pencil and paper as a conventional means of communicating ideas made sketching a natural and easy activity for the team to adopt for visualization design. In addition, the physical arrangement of the group around a table where each member had easy access to communal pencils and sketching paper made it very easy for participants to use the tools.

Unlike sketching tools, the patterns were more difficult to understand and use. Thus, the second factor related to availability is that the use of design patterns was not fruitful until the later stages of the design process, because the participants did not have a clear understanding of the design problem before using sketching and had not yet had sufficient time to discuss and apply design patterns. The density of information in the collection of 36 patterns was seen as overwhelming, as one participant observed: 'Yeah, I don't want too sound lazy, but in some ways it might be better if we just talk in this kind of random way and you say, "A-ha! That's a pattern".' (2:733). During Design Session 2, the design team decided to not to review all of the patterns and relate them to design problems, but rather to permit the expert on the team to offer timely advice. Later in the design process, however, after several design alternatives had been proposed through sketching, the group decided to revisit each of the patterns as a method of decision-making about design choices and to make sure that no visualization design knowledge captured by the patterns was overlooked. This activity, which occurred at the end of Design Session 3, would not have been possible in the initial design stages.

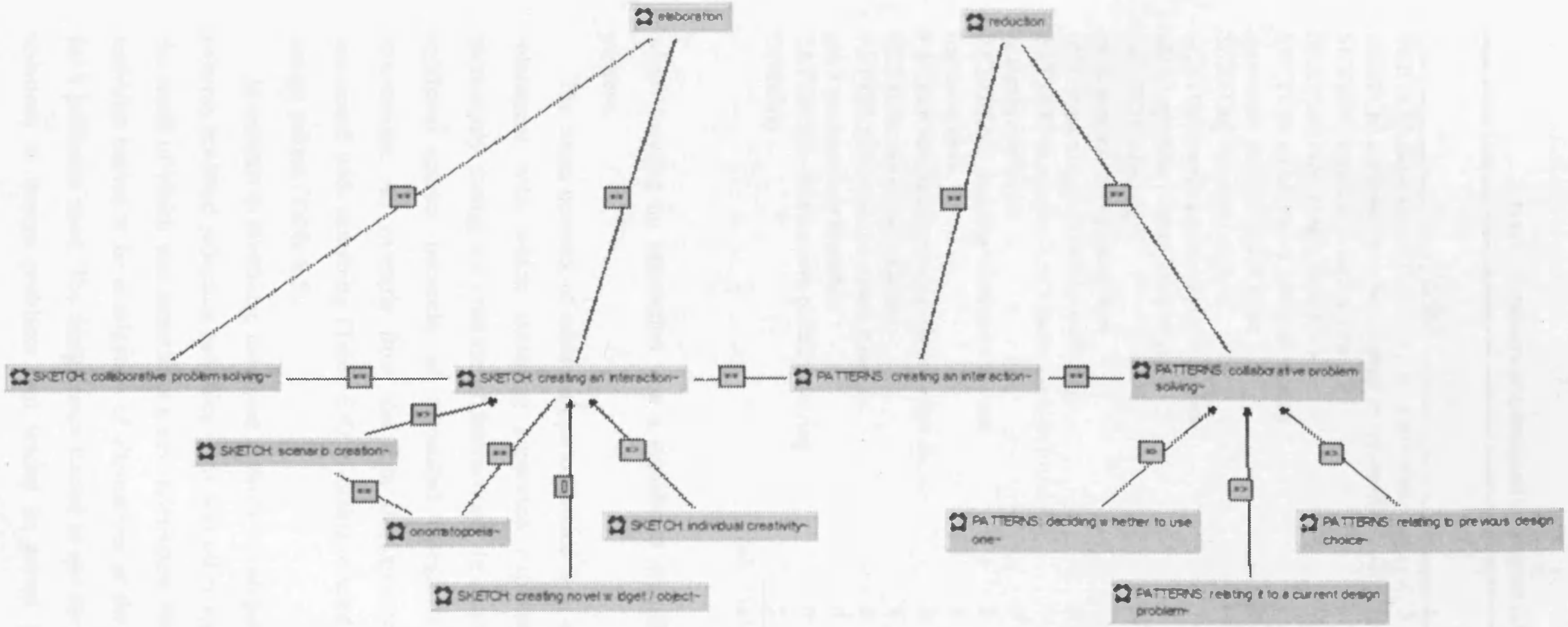
A third factor relating to availability is that in contrast to sketching, the patterns represent a static repository of information. Sketching is a generative and active behaviour and is inherently oriented towards production. Conversely, the design patterns are oriented towards use which requires reading and interpretation. They are not generative, in the sense of creating something new, and do not lend themselves to generation of ideas in the same way that sketching does. Even with the aid of a subject-matter expert, as in this case study, the reading and interpretation of patterns requires time, whereas the generation and association of ideas that sketching facilitates can occur immediately. This affected the availability of both types of tools in the early and later stages of the design process.

6.3.4. Relationships

Although the usefulness of the tools was significantly affected by their availability during the progression of the design process, evidence in the form of the mere preponderance of a group of categories during the early or later stages is insufficient to develop this theoretical model; the relationships among the categories is more significant. The relationships which support the development of the central category can be seen in a network diagram of the categories in Figure 6.3. This network diagram shows the categories which were identified in the data. Whilst the visual layout of the categories is arbitrary, the relationships between them are not. These categories were created during the process of analysing the design session transcriptions. The density of each of the categories results from the instances in the transcriptions. Some categories are supported with many occurrences in the data. Other categories are less richly supported. Table 6.3 shows the number of data events for each of the categories, indicating how many times each of the categories was observed. The density of these categories led to building a densely supported central category.

An example from the data shows how categories were identified in the data. In this excerpt (Table 6.4), J uses different sketching categories to create an interaction. When a user clicks on a receptor, (shown in the 'sandbox' on the right side of the sketch) a tabbed palette of options will appear, allowing adjustment of the parameters to desired values or selection of alternative models. Using metaphors created by previous sketching, he invents a scenario describing a possible interaction, whilst simultaneously sketching the idea. Participants frequently introduced such scenarios with words that created an imaginary narrative in which a proposed interaction might occur, using expressions such as 'supposing…' or 'imagine that…'. To illustrate his point further, J used onomatopoeic sounds whilst sketching, to communicate the interactivity and dynamism of the proposed design. He then applied the label 'Alternates' to one of the described tabs.

The above example was representative of the categories of activity associated with using sketching to create an interaction. As Figure 6.3 shows, these activities tended to be related to one another. For example, scenario creation occurred in 50% of the instances of SKETCH: creating an interaction. Also, Onomatopoeia was associated with creating an interaction in all eight instances and was usually employed in the context of describing a scenario. Very frequently, sketching was used for collaborative problem solving activities



KEY TO RELATIONS: == IS ASSOCIATED WITH; => IS A CAUSE OF; [ ] IS PART OF.

Fig. 6.3: A conceptual network supporting the central category.

TABLE 6.3: DENSITY OF CATEGORIES LEADING TO THE CENTRAL CATEGORY

	DESIGN SESSION				
CODES	1	2	3	4	TOTALS
SKETCH: indicating for context / breadcrumbs	5	25	4	8	62
SKETCH: confirming understanding or agreement	32	10	0	3	45
SKETCH: creating an interaction	21	9	2	8	40
SKETCH: explaining a complex idea	14	10	2	2	28
SKETCH: collaborative problem solving	7	15	0	3	25
generation and evaluation cycle	21	0	0	0	21
SKETCH: scenario creation	11	8	0	1	20
SKETCH: creating novel widget / object	7	9	0	3	19
SME imparting pattern knowledge	5	2	12	0	19
SKETCH: labelling	14	0	1	0	15
using mnemonics or metaphors	7	3	0	4	14
SME imparting visualization knowledge	6	1	3	2	12
PATTERNS: relating it to a current design problem	1	0	8	1	10
qualitative opinions	0	3	4	3	10
PATTERNS: deciding whether to use one	0	0	9	0	9
onomatopoeia	4	3	0	1	8
PATTERNS: relating to a previous design choice	0	0	8	0	8
SKETCH: individual creativity	4	3	0	1	8
PATTERNS: creating a new interaction	0	0	7	0	7
good outcomes are identified	0	0	6	0	6
PATTERNS: collaborative problem-solving	0	0	6	0	6
UNSURE	2	1	2	0	5
Totals	161	102	74	40	397

where creating an interaction was a significant contributor to the exploratory design process.

The dense network of relationships associated with using sketching demonstrates the robustness with which sketching supported exploration activities in Figure 6.3, particularly during the creation of interactions. In contrast to this, the design patterns exhibited sparser networks of associated categories surrounding the creation of interactions. An example from the data demonstrates this. Compare the dialogue associated with sketching (Table 6.4) to dialogue used to create an interaction with a design pattern (Table 6.5).

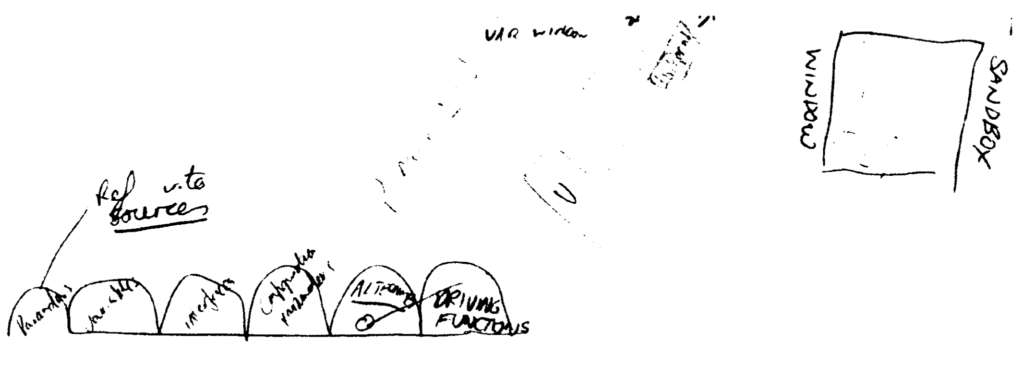
In contrast to sketching, the richer network of categories associated with using design patterns involved reduction activities. This was often via collaborative problem solving, the result of which was sometimes a new interaction. More often, the outcomes of these activities tended to be elimination of alternatives or the selection of a particular pattern for a particular need. The design team tended to use the patterns as ready-made, bolt-on solutions to design problems and tended to accept pattern-based solutions without

TABLE 6.4: CATEGORIES OF ACTIVITY RELATED TO CREATING AN INTERACTION USING SKETCHING. THE RESULTING SKETCH IS AT THE BOTTOM OF THE TABLE. DEFINITIONS OF THE CATEGORIES ARE IN APPENDIX E.

DESIGN SESSION 1 (1:518-530)

LINE		QUOTATION	CATEGORIES
518	J:	Supposing I've specified somewhere or because of the functionality tag in the thing of the models that have the same functionality, I can press this tab and it brings up a list of different choices of model that have the same [making onomatopoeic sounds while drawing] –	SKETCH: Creating an interaction; SKETCH: Scenario creation; Onomatopoeia
519	B:	Puzzle-piece.	
520	P:	Yeah.	
521	J:	Zig-zig edges, things. Then it switches, it does the plug-ability.	
522	P:	Yeah.	
523	J:	It gives me a list of things I can –	
524	B:	What's this called? B: What's this called?	
525	J:	I've put 'alternates' but –	SKETCH: labelling
526	P:	This is basically to represent that fact that in *this thing here*, we have the receptor, and it can either be something really simple, so, the hormone comes in, it just releases a message, that's it. That's all it does. It's just like a switch. Or, you can have a whole cubic ternary model which represents the eight different possible values of the alpha-beta units as they connect and disconnect from each other. And what James is talking about is that you may be using a really simple one and you may what to say, "What's the alternate of this? Okay, there's a really complex model here. Let's try selecting that and see how that –"	SKETCH: confirming understanding or agreement
527	B:	Alternate models. I see. I see. Still fits –	
528	P:	Yeah.	
529	B:	– but it's a different model	
530	J:	Yeah.	

SKETCH 4A SHOWING A RECEPTOR AND TABS CONTAINING ALTERNATES



alteration. The solutions were used as a point of discussion and debate, rather than generatively. In terms of the network diagram in Figure 6.3, this can be observed among the several categories related to collaborative problem solving, using patterns. Compare the richness of this network of categories to the sparseness of collaborative problem

TABLE 6.5: USING A DESIGN PATTERN TO CREATE AN INTERACTION. CATEGORIES ARE DEFINED IN APPENDIX F.

DESIGN SESSION 3 (3:346-350)		
LINE	QUOTATION	CATEGORIES
348	J: Oh, the signalling pathway. What if it's too big to fit on the screen?	PATTERNS: creating an interaction; PATTERNS: deciding whether to use one
349	B: Yeah, what if it's too big to fit on the screen?	
350	P: Yeah that's exactly what this is for isn't it? I mean, personally, I prefer the arrow keys because that's more intuitive. I suppose some people might prefer the **click-n-drag**.	[referring to click-n-drag design pattern]

solving activities, using sketching, which sometimes led to creation of an interaction, but did not often involve additional activities such as scenario creation or creating novel widgets.

6.3.5. Frequency of use

The examples above demonstrate two categories of phenomena, the degree of availability and the richness of relationships, surrounding the use of sketching and design patterns during the course of the design process. This evidence supports the interpretation that the SoViz process effectively supports the visualization design process at different stages during design activities. Additional observations were evident in the data and serve to more adequately complete a description of the effects of using SoViz. These are in the areas of their frequency of use.

Strauss and Corbin caution against using quantitative measures of qualitative data to draw conclusions about the phenomena surrounding a subject of inquiry. With this caveat in mind, it is interesting to note the frequency with which the two design tools were used, taking into consideration factors such as: the total time allowed for design work, decisions made by the design team about how to use the patterns, and influence of the subject-matter expert. Quantifying the codes in the transcriptions shows that sketching was used more often than design patterns, as Table 6.6 shows. Combined with the participants' opinions that the patterns were initially difficult to use, this appears to indicate that team members were more eager to use sketching as a design tool.

However, frequencies alone should not be used to draw conclusions about whether sketching or patterns is better, *per se*. A more precise interpretation is that the tools were appropriate for the times which they were used and that the team spent more time engaged in design elaboration than in design reduction. This tends to confirm the

TABLE 6.6: FREQUENCY OF USE OF SKETCHING AND DESIGN PATTERNS AS DESIGN TOOLS.

CODE	INSTANCES
SKETCH: indicating for context / recall an idea / breadcrumbs	62
SKETCH: confirming understanding or agreement	44
SKETCH: creating an interaction	40
SKETCH: explaining a complex idea	28
SKETCH: collaborative problem solving	25
SKETCH: scenario creation	20
SKETCH: creating novel widget / object	19
SKETCH: labelling	15
SKETCH: individual creativity	8
Total instances of using sketching:	**261**
PATTERNS: relating it to a current design problem	9
PATTERNS: deciding whether to use one	9
PATTERNS: relating to previous design choice	8
PATTERNS: creating an interaction	7
PATTERNS: collaborative problem solving	5
Total instances of using design patterns:	**38**

qualitative opinions expressed about the use of sketching and design patterns. Generally speaking, participants felt that sketching was very important for problem-solving and idea generation and that design patterns were useful for confirming visualization ideas or suggesting alternatives that had not been considered.

It is important to reiterate that the team decided to avoid examining and using the patterns until the design problems were better understood. This determination was made during Design Session 1. At the beginning of Design Session 3, the group decided that the design problems were sufficiently well understood to use the patterns as a means of ensuring that they had not overlooked any visualization design solutions.

Also, the use of sketching as a tool continued late into Design Session 4. This would tend to counter an interpretation of the model that sketching is most useful at the early phases of design and suggests that it is useful at all stages. However, in this case, most of the Design Session 4 sketching events were instances of using previous sketches to recall a context of discussion, and not for the generation of new design ideas.

6.4. Summary

In terms of Research Question 3, this combined qualitative approach has illustrated a method of evaluating Information Visualization design process. It achieved this by studying design in the environment in which it occurs and using methods which are appropriate for researching real problems, in the context in which they occur. The results produced shed light on both the practical effects of using the SoViz design method and the theoretical consequences of applying these techniques.

In terms of Research Question 4, the practical outcome of Action Research was that the participants were able to use the SoViz design activity with the practical effects of enhancing their learning about visualization, and learning a way to structure their future design work. This was interpreted as a positive outcome because it was the sort of change the group were seeking in the organization.

The results of GT analysis were theoretical observations about the use of sketching, design patterns, and expert knowledge for visualization design. These addressed Research Question 5. The central category leads to a theoretical interpretation that sketching and design patterns used as part of the SoViz approach supported visualization design activities at different stages of the design process. This was determined by their availability to designers. With only minor encouragement, sketching was easily and immediately available to the Beacon team as a tool to generate design ideas, enhance creativity, and to solve design problems. By contrast the design patterns were not available until the end of the design process primarily due to their level of detail and complexity, which inhibited uptake.

Because the study was qualitative in nature, generalizations to other visualization design cases should be taken with consideration that this was only a single case; other cases may differ substantially. However, the large quantity of literature on other design disciplines such as architecture and engineering tends to support the hypothesis that the SoViz design method used in this study will be effective for other visualization design problems.

Following the design sessions, a prototype was created for one of the eight visualization projects and a usability study was conducted to determine its successfulness. The next chapter presents a discussion of the SoViz prototyping process and the results of this work, which demonstrated the success of the SoViz method and provided additional insight into Research Question 4 about the practical outcomes of using SoViz.

Chapter 7: A Visualization Prototype

7.1. Introduction

The results of the Action Research intervention showed that the SoViz method supported the Beacon team in creating visualization designs. In the AR work, SoViz was used as a research framework to structure and to study design activities. With this perspective, the work produced eight high-level visualization design concepts which were deemed to be potentially useful to the Beacon Project team. SoViz can also be viewed from a systems design perspective, in which it is meant to yield a visualization system. This chapter follows the development of a single visualization concept through the entire SoViz framework described in Chapter 4, with a particular focus on the later stages. This demonstrates the four main parts of the SoViz process by a specific example and shows how SoViz worked within the Beacon team.

To review, the SoViz framework consists of four major parts: Requirements Gathering (Part 1), Visualization Design Activities (Part 2), Prototyping (Part 3) and Evaluation (Part 4), culminating in a visualization version release. Requirements gathering entailed collecting expectations from as many stakeholders of the project as was practical: interviews and an existing requirements document provided the data. These were used as a starting point and guide during the visualization design activities (Part 2), which were design working sessions. In prototyping (Part 3), the visualization concept is realized as visualization software. This process was completed by programmers who were participants in Part 1. Many partly-functional prototypes were produced over a two-month period. Evaluation (Part 4) is used to assess and critique the software as it is being designed. In this case, the prototypes were evaluated by the programmers and other members of the design team. Changes in the designs were proposed and implemented. For the Beacon team, evaluations often led immediately to another iteration of visualization design activities (Part 2), before more prototyping. When the design team had judged a visualization to meet its requirements, further changes to the code were halted and a version was released. In this chapter, parts 3 and 4 are described in detail.

To show how SoViz prototyping and evaluation were used for turning design ideas into software, the next sections describe this evolutionary and iterative activity. One assumption of the SoViz approach is that using SoViz not only supports the generation of design ideas and software production, but that it also results in a *successful visualization.* To further evaluate the SoViz approach as a design method, it was necessary to test this assumption by producing and assessing a visualization. To do so, one of the eight visualization concepts produced by the Beacon Project design team was selected for prototyping (Part 3) which is reported in section 7.2. This concept was referred to as 'Model Cartoons'. After producing a Model Cartoon visualization release called 'CalViz', an evaluation (Part 4) of the prototype was performed, to measure its success. This is reported in section 7.4. The positive results of this experiment further support a conclusion that the SoViz approach leads to designing successful visualizations.

7.2. Creating Model Cartoons: Visualization Design Activities

Using the SoViz approach, the Beacon Project design team first produced the Model Cartoons concept during visualization design activities. Model Cartoons were conceived as a generic class of possible visualizations of the many computational models used on the project. The computational models are groups of mathematical equations which describe various aspects of the human body's glucose homeostasis system. These focus on liver response to glucagon and insulin, but also include pancreatic production of these hormones in response to changing glucose levels. The team decided that a 'cartoon' or animated, interactive visualization of these models would help to demonstrate how accurately the models mimic biological systems in real human livers. Thus, the Model Cartoons concept could be realized as any of a number of different visualizations. For instance, a cartoon visualization of models of pancreatic function might show a cartoon pancreas. Communication among cells in the liver cell plate could similarly be shown by a stylized representation of a chain of cells engaged in signalling to one another, and so on. Although there are a few commercial and open-source applications which incorporate cartoon visualizations in cellular modelling software, notably Virtual Cell[6], the team members had decided that none of the available packages would suit their needs. For example, one of the team's objectives for using Model Cartoons was to communicate progress effectively on the production and the accuracy of the models to non-mathematicians, both within and outside the project. This is difficult to do with the equations used for the models because they are highly specialized. Therefore, a visual abstraction in the form of a cellular 'cartoon' was seen as a good way of representing the behaviour of the computational models. None of the available packages do this effectively.

The Model Cartoons concept is for a class of visualization tools that would *automatically* generate an animated representation of any kind of computational biology process modelled in the Beacon Project. This visualization concept, along with seven

[6] http://www.nrcam.uchc.edu/

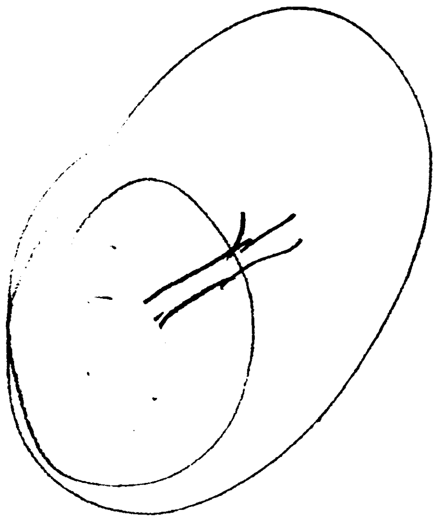
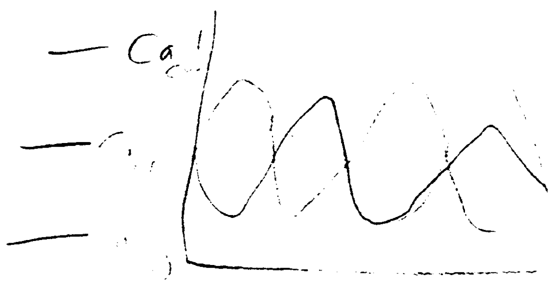
others, was created in Design Session 2. After Design Session 4, the requirements specification meeting was held and P produced a document which summarized the eight visualization sub-projects (Appendix D). This specification described Model Cartoons as follows:

> **Model Cartoons: attractive animations of model executions**
> At the moment, mathematical models of biology require extensive presentation work to make it possible to visually present these models to non-mathematicians. Animations that represent the behaviour of the model in a cartoon way familiar to biologists, as opposed to plots or abstract structure diagrams, must be built by hand on a case-by-case basis.
>
> This project is to automate the generation of such cartoon animations. Various model elements will have an associated graphic which can be automatically rendered during a model execution. Examples include collections of coloured spots, which increase in density as a concentration increases and a gate symbol that changes its degree of openness based on the permeability of a biological gate or membrane.
>
> This project will require a skilled programmer with some significant graphics expertise.

This high-level description does not address the interactions and visual mappings of any particular Model Cartoons visualization. Because Model Cartoons of different biological systems do not have the same attributes (e.g. liver cells, pancreas cells, signalling channels, etc.) and support different user tasks (e.g. comparing variables, determining durations, deciding if a model is accurate), some of these decisions must be taken during prototyping.

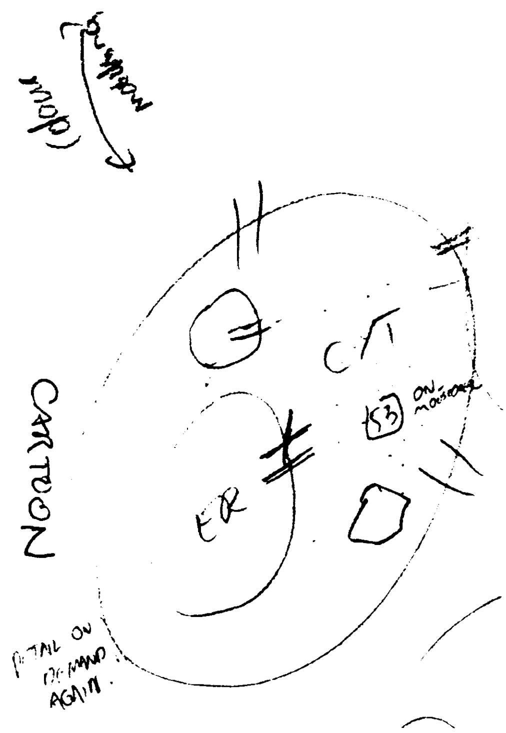
The Model Cartoons concept was created during Design Session 2 of the SoViz visualization design activities. At this time, the team were trying to solve the problem of representing the computational model of calcium oscillations in liver cells. This is normally done using several *x-y* graphs of changes in variables over time. However, the team members felt that although these are useful for some tasks, they do not fully communicate the physical processes involved, nor provide an intuitive understanding. It is also difficult to relate several two-dimensional graphs to one another. Team member J observed that a cartoon representation would allow all of the variables to be represented at once and could be animated to show their changes over time. Others agreed with this assertion and embellished the idea. The dialogue and the sketches produced during this activity led to the Model Cartoons concept. These are presented in Table 7.1a and 7.1b, which provide an extended excerpt from Design Session 2. This illustrates the team's conception and design ideation about the Model Cartoons concept; they used the

TABLE 7.1(A): THE DEVELOPMENT OF THE MODEL CARTOONS CONCEPT DURING DESIGN SESSION 2, USING SOVIZ.
[ANNOTATIONS IN BOLD]

DIALOGUE (2:18-26)			SKETCH PRODUCED
18	M:	If you want to do it more active, or more alive, because we are talking all the time about timescales. If you want to do how the protein is synthesized from the ribosome, oh how is it with the proper timescales, okay this I'm, going to believe it.	SKETCH 9A: 
19	J:	Yeah.	
20	M:	But like a cartoon.	
21	J:	Yeah, exactly. So this cell cartoon idea is one of the things I wasn't sure how useful it would be. Because, for example, one thing I thought of: supposing we've got a whole model of a cell and let's just pick a very simple example of somewhere where there is a varying amount of openness of the channel between the ER and the cell. Okay? So there's the ER. **[inner circle]** I'll draw a bit bigger.	
22	M:	Yes.	
23	J:	Big cell. Big ER. Here's a channel. **[parallel lines]**	
24	B:	ER?	
25	P:	Endoplasmic reticulum.	
26	J:	Okay. Now, during the course of time evolution of the model, this channel **[Sketch 9b, black line]** during calcium oscillations opens and closes and calcium moves between here and here as part of the oscillations. Now, the obvious way to represent that is this way: so, solid line for calcium in the cell, a red line for calcium in the ER and a green line for the channel openness. So the channel openness is between 0 and 1 and during the oscillation it goes □ I can't exactly remember □ but something like that and the ER calcium goes something like that, and the cell calcium goes something like that - you can draw it like that.	SKETCH 9B: 

physiological process of calcium oscillations as an example. The dialogue also shows the visual design activities in action, from the conceptualization, communication, embellishment and development of an idea using sketching, and finally, relation of the idea back to one of the requirements identified in the requirements gathering phase. Notably, design patterns were not used at this point of design elaboration.

TABLE 7.1(B): THE DEVELOPMENT OF THE MODEL CARTOONS CONCEPT DURING DESIGN SESSION 2, USING SOVIZ. **[ANNOTATIONS IN BOLD]**

DIALOGUE (2:27-46 CONTINUED)	SKETCH PRODUCED
27 J: But an alternative way to draw it is this: to have a picture of a cell, with these dots here – 28 M: Yes. 29 P: – this is the density of calcium **[stippling inside the larger circle]**. So you can see that this stippling is dense in here **[circle labelled ER]** and un-dense in there **[circle labelled CYT]** and then a few seconds later, you can see that the stippling for the calcium is dense in here **[CYT]** and not dense in there **[ER]**. And similarly, this channel is represented by a picture of a little doorway opening and closing. And the doorway is opening and closing in time or against time. And you have a window open on this, so you can actually see the graph, but you also have this thing as well. 30 S: This one is easy just to see straightaway what is going on. But you need like, maybe a number somewhere between here to see what. 31 P: Precise figures. 32 J: Right. 33 S: For example, this region, something like that. Then you can use it. But that is, you can see it but you can't really use it.	SKETCH 9C: 

34 J: You can't use it but you can get a kind of feeling from it.
35 M: But for example, you can put how the calcium goes out if there are stippling in red, and you put it red how this goes.
36 J: You can make the colours match between –
37 M: And then, every time that's open, closed, open, closed.
38 J: The nice thing about this, you see, when I had this idea, is that you can put lots and lots, because you know I can then start putting channels on here and channels on here [J adds several parallel lines], and little hexagons for glucoses moving around. And I can put it on all at the same time.
39 M: Yes.
40 J: You could even do, right: here's going to be my nucleus now, and some stuff comes in and a little one of these channel pictures comes out and then goes over and joins the membrane. And lo and behold, we've got a new gap junction. [top right corner, parallel lines] And you can actually see the gap junctions moving through the golgi as they get on their way to –
41 M: This is – if I'm going to teach someone what I'm doing, if I'm going to give a lecture, I'm going to put these kind of graphs. But sometimes people need this.
42 S: People use these kind of things in a presentation. Like in a PowerPoint.
[comments omitted]
46 P: 'Allows for the generation of animations and snapshots for presentations.' Requirement number five.

Sketch 9a demonstrates the first ideation and sketching of the Model Cartoons approach. At this point in the design process, biologist M explained that she would like to see timescales and protein synthesis represented by cell cartoons. Recalling an idea from Design Session 1, J described how a cell cartoon would help with this kind of task. As he explained the concept, he sketches a picture of a cartoon cell to confirm the idea with M. J's picture contains cell cytoplasm, ER, represented by concentric circles and calcium channels represented by parallel lines (line 2:23). All of these items are parameters in the computational models of calcium oscillations. A cartoon picture of a cell is common in computational biology and biology in general. What is novel, in this case, is the attachment of this cartoon representation to the underlying models.

To explain the idea further to his biologist colleagues, J used a different visual representation from what they are already used to. J pointed out (2:26) that the models are generally represented by charts which ploy *x* versus *y*, as in sketch 9b. Biologists are familiar with these because they frequently use such charts to assess and report the results of lab experiments. But as J observed, this representation makes comparisons difficult and does not permit all of the attributes of the model to be easily perceived as a coherent whole. He then presented an alternative representation (lines 2:27-29), and explained how a cartoon picture of a cell can present an animated representation of the processes which are described in the charts. During his description, he added visual elements which are contained within the model parameters. Stippling represents the concentration of calcium ions in the cell bodies. J suggested that differences in stippling will illustrate differences in calcium density. S, a biologist, agreed (2:30), stating 'This one is easy just to see straightaway what is going on'. She still would like to have access to numbers underlying the pictures, because she said that without numbers she cannot use the underlying data. Others on the team then explained that although the specific data values may not be useful for other kinds of tasks, the animated cartoons provide an intuitive representation of the data which the charts do not provide. J continued to embellish upon this idea (lines 2:38-42), by associating other model parameters to visual elements, such as 'hexagons for glucoses' and 'gap junctions'. M identified this as an important tool for explaining calcium oscillations to others, explaining that during teaching she would use graphs. 'But', she said, referring to the sketch that J had drawn, 'Sometimes people need this [an animated cartoon of the biological system]'. P, the project leader, then related this back to one of the list of requirements from the SoViz requirements gathering (Part 1) activities: the visualization provides '...animations and snapshots for presentations'.

TABLE 7.2: IDENTIFYING THE OBJECTIVES OF MODEL CARTOONS VISUALIZATIONS DURING DESIGN SESSION 2.

DIALOGUE (2:65-88)	
65 B:	It would be good for the requirement that [name removed] has which is that it should be able to be explained to others.
66 J:	That it should be pretty.
67 S:	Yes.
68 B:	And be pretty. But as the life scientist, this representation is maybe not as useful as this one. Is that right?
69 S:	No, but uh. This **[Sketch 9B in Table 7.1(a)]** is quantitative, but this **[Sketch 9A in 7.1(a)]** is, you know, lots of information there. From the cartoon I will know that the protein, for example, this protein comes from the nucleus and moves to trans-membrane. That's one information. And then some protein within the cytosome, if it stays, that's one information. So, you can't get that from this **[graph, Sketch 9B]**.
70 B:	You cannot get that from this **[graph, Sketch 9B]**?
71 S:	No.
72 J:	No, that's right.
73 B:	Okay, can we make sure that I understand that as clearly as you do?
74 M:	When I arrived here I didn't know what is a Connexin. [I had to] read about Connexins and then start to work on it. Then if I go to give a talk to someone, they are not going to know what is a Connexin.
75 B:	No.
76 M:	No. The last five years time maybe they listen about the calcium oscillation.
77 B:	Right.
78 M:	Then you must put the people in context and when they are understanding what parameters you are measuring, then they are going to understand all of the graphs.
79 P:	Can I try to summarize?
80 B:	Yes, what I like about this is that S said she could make decisions based on this representation [model cartoon], and that's what's important, in my mind.
81 J:	Yeah.
82 B:	That one can make a decision based upon the representation. That it's not just a pretty picture. That you can actually think intelligently.
83 J:	One part about this is it doesn't just visualize quantities and the changes in quantity. It visualizes processes.
84 M:	Yes.
85 J:	Because we can see not just that this calcium is going up but that it's come from here. You can see the movement. You can see why. So, with this graph, it's not easy to see – one may look at this and make the obvious conjecture that the red one is going down because either the green one or the black one is going down but the nature of cause and effect is not included in that visualization.
86 P:	Yeah.
87 J:	This visualization includes the nature of cause and effect.
88 M:	Yes.

Later, the design team discussed the benefits of using the Model Cartoons approach, pointing out that it gives not only a holistic view of the models, but also meets other objectives. There are five objectives described by the dialogue in Table 7.2. They must:

1. communicate the work of the Beacon project (2:65-66 and 2:74-78);
2. be pleasant to look at, i.e. 'pretty' (2:66-68);
3. provide a qualitative experience (2:69);
4. aid decision-making (2:80-82);
5. demonstrate cause-and-effect among variables (2:83-85).

These design activities resulted in the creation of the Model Cartoons concept. During this process, calcium oscillations in liver cells were used to illustrate one way in which a Model Cartoon visualization might be realized. The other seven concepts were also produced in a similar fashion. After the four design meetings were completed and the requirements specification document was produced by P, the Beacon team selected one of the eight visualization concepts for prototyping. They chose to produce a Model Cartoon visualization.

There are many computational models used on the Beacon Project. The original idea for Model Cartoons is that a single visualization system would be able to create a cartoon visualization for any of the models. However, creating a prototype of this was not seen as practical, given the time and resources available on the Beacon Project. Therefore, the team chose to make a prototype of only one model. The team selected the model for calcium oscillations which was used during the design activity described above. Calcium oscillations within liver cells are involved in regulating blood glucose and play a significant role in how the body uses stored energy. The rationale for using this model was that the team wanted to prototype a Model Cartoon of a known and accepted model. Calcium oscillations are well understood by practitioners in the discipline and the model used by the Beacon team is known to be accurate. This would allow team members and other practitioners to focus on the effectiveness of the visualization, rather than the accuracy of the underlying model. The team needed a name for this visualization, and decided to call it 'CalViz'.

The team identified four reasons why CalViz would be a good choice for a prototype. Foremost, it would demonstrate that the Model Cartoons concept works; that at least one of the computational models can be implemented as a useful, meaningful visualization. In addition, it would show progress to others within the Beacon Project by producing visual software that other stakeholders on the project would be able to relate to. Mathematical models are not very tangible. Visual representations of the models are easier to understand and as such, can demonstrate progress. A third objective was to communicate the calcium oscillations concept more effectively to students. Currently, this is done in PowerPoint supported lectures by explaining the important parameters of the model, describing the model structure with equations, and demonstrating the model outputs using charts or simple animations. An interactive visualization was seen as providing a way to unify these in a single user interface. Finally, the team thought that CalViz would be a useful tool to communicate progress to scientists and researchers outside of the Beacon project.

7.3. Creating Model Cartoons: Prototyping and Evaluating

The CalViz Model Cartoon began as a design concept, which was created with sketches in Design Session 2. The final high-level specification was described at the end of the Design Session 4. To realise this concept as a visualization tool, further sketches were produced and further design refinement was required, so that prototyping could begin. This process began as a meeting with the Project leader, P, to more precisely specify the objectives for the calcium oscillations visualization and to continue reducing design alternatives. During this meeting, a sketch was produced (Figure 7.1) to capture some of the visual mappings, to determine the layout of the visualization, and to describe some of the interactions that it would support. This can be construed as a fifth design session, which was held for the purpose of refining a specific prototype of the Model Cartoons visualization concept.

7.3.1. Attributes of CalViz

The sketch produced by P in Figure 7.1 shows the key attributes of the CalViz Model Cartoon. Most of the key attributes that were captured in the initial sketch were implemented in the prototype. Several visual mappings were proposed in the sketch. The structure of a liver cell is represented as two concentric circles. The inner circle labelled 'a' *does not* represent the cellular nucleus but rather, the proportion of *endoplasmic*

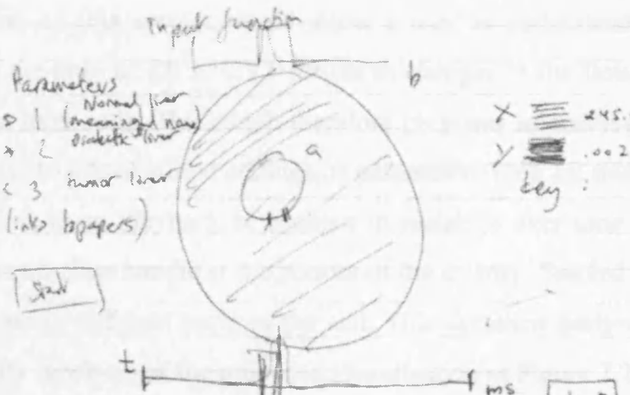


Fig. 7.1: A design sketch showing the key attributes of the CalViz Model Cartoon.

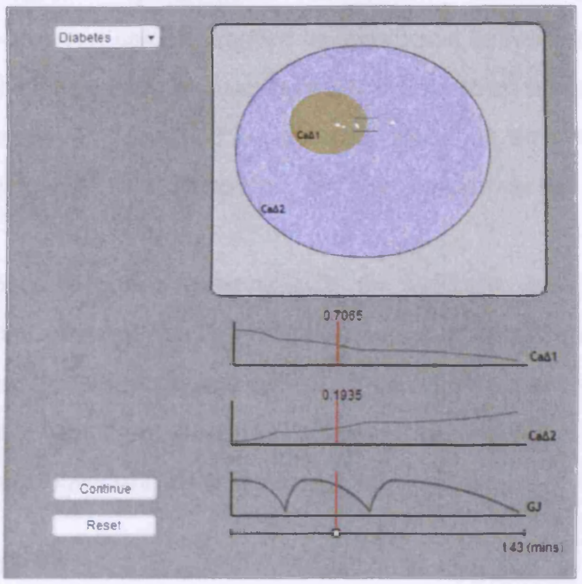


Fig. 7.2: A design mock-up of the CalViz Model Cartoon.

reticulum (ER) in the cell body, referred to as the *cytoplasm* (CYT), which is labelled ‘b’. This is one of the key variables of the computational model of calcium oscillations. The proportion of ER within the cytoplasm has a direct effect on the movement of calcium ions within and between cells. The size of the ER circle within the cell thus represents the proportion of ER in the computational model. As shown in the dialogue excerpts in Tables 7.1(a) and 7.1(b), the assumption for CalViz was that providing an interactive representation of this system would allow a user to understand more clearly how the variation of the ratio of ER to CYT results in changes in the flow of calcium ions within and between liver cells. The sketch therefore proposes interactive widgets, which would allow the user to adjust stored settings of parameters such as: normal liver, diabetic liver and tumourous liver. Playback of changes in variables over time would be controlled by adjusting the timeline handle at the bottom of the display. Shaded areas represent calcium concentrations in different parts of the cell. This sketched design was used to produce a higher fidelity mock-up of the proposed visualization in Figure 7.2.

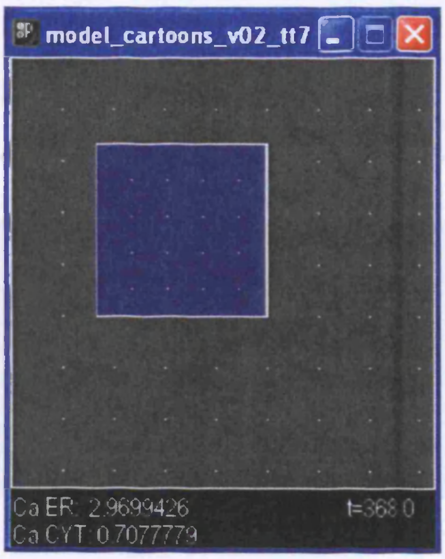
This mock-up was animated, but non-functional; none of the visual attributes was tied to the data produced by the Beacon Project models. It was meant to be a proof-of-concept to elicit further feedback from the team; to act as a preliminary stage between sketching and prototyping. The mock-up was produced with Macromedia Flash and posted to a website for the Beacon team to review. Feedback in the form of emails was positive. The layout, charts, and cellular representations were deemed successful. Further decisions based on the mock-up were: calcium concentrations should be represented as densities of dots within the cell bodies; there was no need for a drop-down menu of cell health; the timelines for variables were identified as important. Based on this, it was decided that a working prototype using data generated by the computational models could be programmed.

Based on feedback from J, a mathematician, the team also decided that it would be impractical to link the visualization to a calcium oscillation model which was running in real time. Therefore, the computational model of calcium oscillations was executed to generate an output file for a fixed duration. This output file would provide the data values to which the visual attributes of CalViz were mapped.

7.3.2. Prototypes

At this point, some of the team members became unavailable for consultation. However, several prototypes were produced by B (i.e. the author) over a period of six weeks. The prototypes were produced using an open-source, visually-oriented development environment called Processing[7]. These rapidly generated prototypes were used to test the visual mappings and interactions. The design team provided little feedback during this period, though some opinions were expressed via emails from the project lead, P. The

TABLE 7.3: CALVIZ PROTOTYPE 1 AND THE DESIGN ACTIVITIES

PROTOTYPE VISUALIZATION	DESIGN ACTIONS AND DECISIONS
	Visual Mappings: Calcium concentrations are mapped to dot densities. ER is shown as a blue square. CYT is shown as a grey square. Values for time and calcium concentrations are shown in text. The ratio of calcium in each cell body is not adjustable. Interactions: None. Design decisions: Try circles instead of squares. Map the density of calcium within this circle as areas of white density.

7 http://www.processing.org

TABLE 7.4: CALVIZ PROTOTYPE 2 AND THE ASSOCIATED DESIGN ACTIVITIES

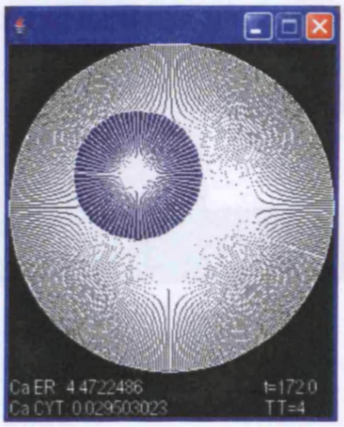
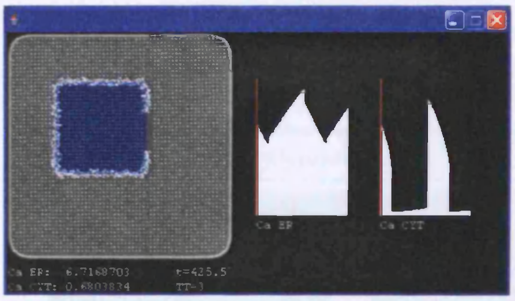
PROTOTYPE VISUALIZATION	DESIGN ACTIONS AND DECISIONS
	Visual Mappings: Concentric circles are used to represent the cell structures. A circle-filling algorithm paints dot-densities within the ER and CYT. The ratio of calcium in each cell body is not adjustable. Interactions: None. Design decisions: Circular densities produce a moiré effect which is undesirable. Use squares and dot-densities to represent calcium concentrations. It may be useful to show histograms.

TABLE 7.5: CALVIZ PROTOTYPE 3 AND THE ASSOCIATED DESIGN ACTIVITIES

PROTOTYPE VISUALIZATION	DESIGN ACTIONS AND DECISIONS
	Visual Mappings: Calcium concentrations are mapped to dot densities. ER is shown as a blue square with a reticulated perimeter. CYT is shown as a grey square. Values for time and calcium concentrations are shown in text and as x-y charts. The ratio of calcium in each cell body is not adjustable. Interactions: None. Design decisions: No feedback.

`TABLE 7.6: CALVIZ PROTOTYPE 4 AND THE ASSOCIATED DESIGN ACTIVITIES

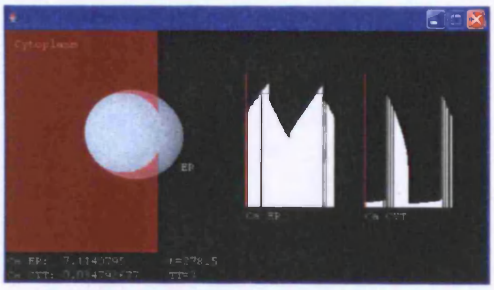
PROTOTYPE VISUALIZATION	DESIGN ACTIONS AND DECISIONS
	Visual Mappings: Calcium concentrations are mapped to colour saturation. ER is shown as a blue sphere. CYT is shown as a grey square. Values for time and calcium concentrations are shown in text and as x-y charts. The ratio of calcium in each cell body is not adjustable. Interactions: None. Design decisions: No feedback.

TABLE 7.7: CALVIZ PROTOTYPE 5 AND THE ASSOCIATED DESIGN ACTIVITIES

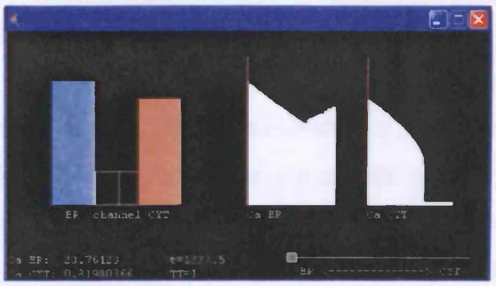
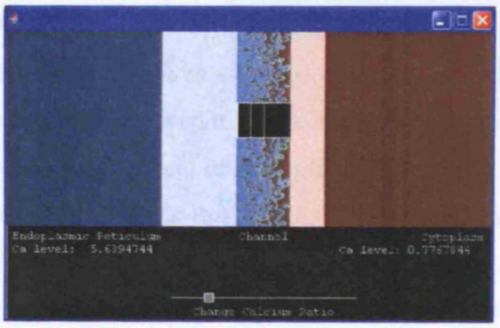
PROTOTYPE VISUALIZATION	DESIGN ACTIONS AND DECISIONS
	Visual Mappings: Calcium concentrations are mapped to vertical histograms. ER is shown as a blue bar. CYT is shown as an orange bar. A representation of the calcium channel is shown between these two bars in the form of animated white lines. Values for time and calcium concentrations are shown in text and as x-y charts. Interactions: The ratio of calcium in each cell body is controlled by a horizontal slider widget. Design decisions: The horizontal slider is a useful widget. The calcium channel metaphor shows how calcium ions move between the cell bodies, but is not ideal. The vertical bars for the ER and CYT are not helpful because they merely replicate the data values in the charts.

TABLE 7.8: CALVIZ PROTOTYPE 6 AND THE ASSOCIATED DESIGN ACTIVITIES

PROTOTYPE VISUALIZATION	DESIGN ACTIONS AND DECISIONS
	Visual Mappings: A 'close-up' of the ER-to-CYT boundary is represented by a reticulated vertical bar in the middle of the display. A black calcium channel connects these two structures and shows calcium animated as vertical lines. Calcium concentrations are mapped to areas of low colour saturation on either side of the ER wall. Values for time and calcium concentrations are shown in text. The ratio of calcium in each cell body is controlled by a horizontal slider widget (bottom). Interactions: The ratio of calcium in each cell body is controlled by a horizontal slider widget. Design decisions: A visual representation of the entire cell is important. The data values are not important for understanding the model and can be eliminated. The user should be able to directly interact with the ER to adjust the ER-to-CYT ratio.

most important opinion was that dot-densities, and not any other visual metaphor, should be used to represent calcium concentrations. The CalViz prototypes and the design decisions taken for each of them are shown in Tables 7.3 –7.8.

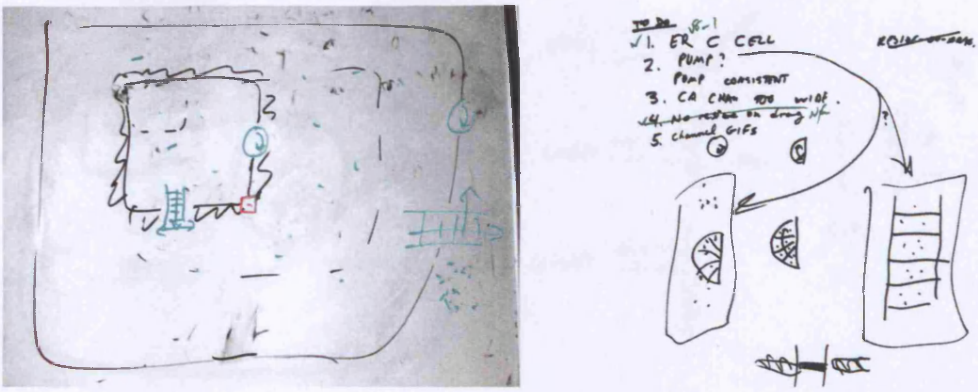
At the end of this prototyping period, part of the design team reconvened (B and J), as no other participants were available. This meeting was held to evaluate the prototypes, as described by the SoViz framework (Part 4). This evaluation was undertaken as an informal review of the visual attributes and interactions of each the prototypes with J.

Because other team members were not available, design decisions were taken without their feedback, so that further prototyping could continue without delay. Ideally, this would have been done with participation of more of the team members.

7.3.3. Evaluations

The evaluation of the prototypes revealed a significant misunderstanding between B and the design team about the calcium oscillations model. This was that showing the calcium ratio between the ER and the CYT is the primary goal of the computational model and that it was very important to visualize calcium ions moving from the ER to the CYT. This movement is directly proportional to the amount of ER in a cell. The closest prototype to achieve this was Prototype 6 (Table 7.8). However, because the visual metaphor is a magnified view of the ER-CYT boundary, it does not effectively display the proportion of the ER within the cytoplasm. Therefore, it was decided that the visualization should show these proportions as a cartoon of the cell structure using concentric circles, as was done in prototypes 2–4, and it should allow the user to adjust the amount of the ER within the cell by clicking on and dragging the ER perimeter. A whiteboard sketch (Figure 7.3(a)) was used to determine this and to illustrate what visual form the final version of the software should take.

Another of the problems identified during this evaluation was how to represent calcium ions moving from the ER to the CYT. This is accomplished by two structures: calcium channels and calcium pumps. They represent two different biological processes but both pumps and channels involve moving calcium ions from one cell structure to another. The question became one of how to represent these structures differently. In Prototype 5 (Table 7.7) and Prototype 6 (Table 7.8), the calcium channels had been represented with white vertical lines moving in a calcium channel between the ER and CYT. Calcium pumps were not shown. This was identified as a problem by J, who suggested that both should be visible, since both are in the underlying mathematical model. This was a problem for the prototype because both pumps and channels should



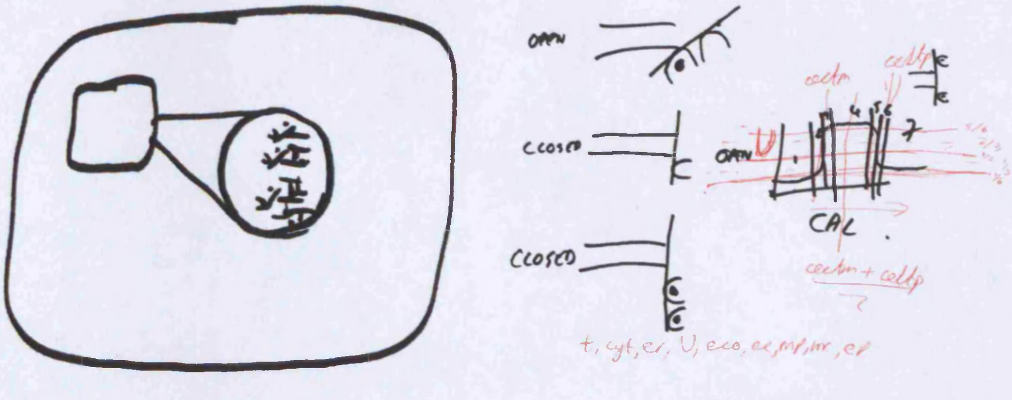
(a) (b)

Fig. 7.3: A whiteboard sketch (a) used to decide the layout and visual attributes of the CalViz prototype. The sketch (b) used to create the visual metaphor for calcium pumps and channels.

show movement of calcium ions, though they needed to be represented differently. Two approaches were sketched: a rotary-pump metaphor (Figure 7.3(b), left) and a conveyor-belt metaphor (Figure 7.3(b), right). The decision was taken to represent channels using the conveyor-belt metaphor (but without parallel lines), which had grown from the previous prototypes. The rotary-pump metaphor was used to represent calcium pumps. Calcium ions were to be represented by white dots travelling through these structures. The decision was then taken to develop the prototype using these visual metaphors.

Two interaction design problems were resolved by sketching: resizing of the ER and displaying the action of calcium channels. The first problem was that the user should be able to change the size ratio of the ER-to-CYT. This causes all of the other values in the model to change. Calcium oscillations happen faster when the ER-to-CYT ratio is smaller. User interaction would allow adjustment of this ratio and the resulting movement of white dots representing calcium ions. For ease of programming, it was decided to place a handle in the lower right corner of the ER (red box in Figure 7.3(a)). This handle would allow the user to resize the ER and would thus change the speed and movement of calcium ions.

The second interaction problem was how to show the open-ness of calcium channels. Since this is also a key parameter represented in the model, J felt that it was important to represent a view of this behaviour. The solution proposed (Figure 7.4(a)) was drawn from two of the visualization design patterns – details-on-demand and zoom, and a visual metaphor for a 'valve'. The proposed interaction was that moving the mouse pointer over the ER-CYT boundary would cause a magnified view of that boundary to be displayed.



(a) **(b)**

Fig. 7.4: A sketch used to solve the problem of representing channel openness. Design patterns (Details-on-demand, zoom), and sketching the "valve" visual metaphor were used.

This magnified view would show calcium ions attaching to calcium channels in the form of a valve. This would cause the channels to open and close. The opening and closing of a channel valve occurs at different points of the calcium oscillation process. In Figure 7.4(b), these points are represented by the intersections of red lines on an *x-y* chart of the variable representing channel openness. It is a chart of this sort that is traditionally used to represent this variable and it is this abstraction that the visualization is intended to replace. The final version of this solution is shown in Figure 7.5. This was the last design problem that was solved during prototyping.

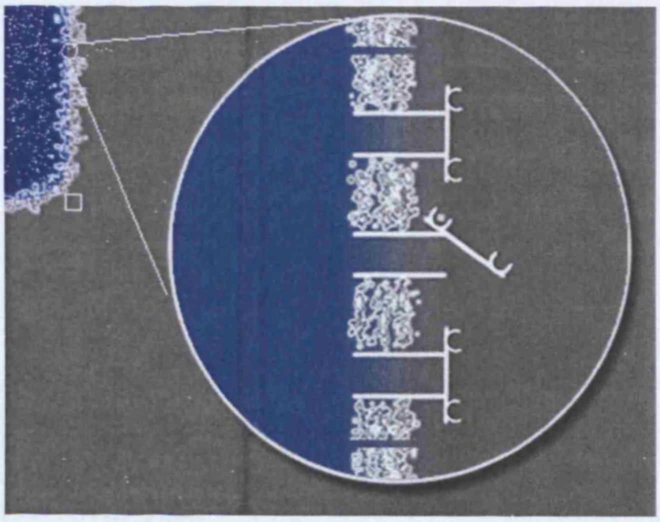


Fig. 7.5: The solution for representing calcium channel openness. T-shaped objects are channel valves, containing calcium ions in receptor 'cups'. The presence or absence of an ion changes the state of the valve.

7.3.4. CalViz Production and Version Release

CalViz was produced over a 14-month period, as shown in Table 7.9, which demonstrates both the chronology of events and the iterative nature of the SoViz design process. The table lists the meetings and activities associated with producing the CalViz software. Those activities are also categorized according the stages of the SoViz framework. After the requirements were gathered, the design process consisted of an iterative sequence of three steps: visualization of design activities, prototyping, and evaluation. The sequence was repeated until the team decided that the final visualization could be released. On several occasions, the results of evaluation led immediately to new visualization design activities during the same meeting, which informed further prototyping.

The bulk of the design work occurred at the beginning of the SoViz process and the prototyping occurred towards the end. However, the production of CalViz did not occur as a neatly linear sequence of steps, but was a gradual refinement process which involved designing, prototyping, evaluating the results, and further design decision making, as demonstrated above. As the development process matured, fewer people were involved in

TABLE 7.9: THE CALVIZ PRODUCTION SEQUENCE, SHOWING THE STAGES OF THE SOVIZ PROCESS.

SoViz Stage	Date	Activity
Requirements Gathering (Part 1)	01/08/05	Interviews with: S, J, O, AW, M
	15/08	Interviews with: AF, L
Visualization Design Activities (Part 2)	20/09	Design Session 1
	21/09	Design Session 2: Model Cartoons concept is created.
	04/10	Design Session 3
	31/10	Design Session 4
	14/04/06	Design Session 5 with P. Model Cartoons (Figure 7.1) chosen for prototyping. Model of calcium oscillations to be used.
Prototyping (Part 3)	28/04	CalViz mock-up (Figure 7.2) produced in Flash and sent for evaluation by email
Evaluation → VDA (Parts 4 & 2)	03/05	Evaluation with P and J, using Figure 7.2, design decisions taken
Prototyping	15/08	Prototypes 1
	17-19/08	Prototype 2
	19/08	Prototype 3
	21/08	Prototype 4
	24/08	Prototypes 4, 5
	07-17/09	Prototype 6
Evaluation → VDA	03/10	Meeting with J to discuss the prototype, design decisions taken
Prototyping	08/10	Prototypes 8.1, 8.2
Evaluation → VDA	11/10	Meeting with J to program the prototype, design decisions taken
Prototyping	26/10	Prototype 8.3
Evaluation	27/10	Evaluation and decision to finish prototyping
Visualization	27/10	Visualization Release: CalViz

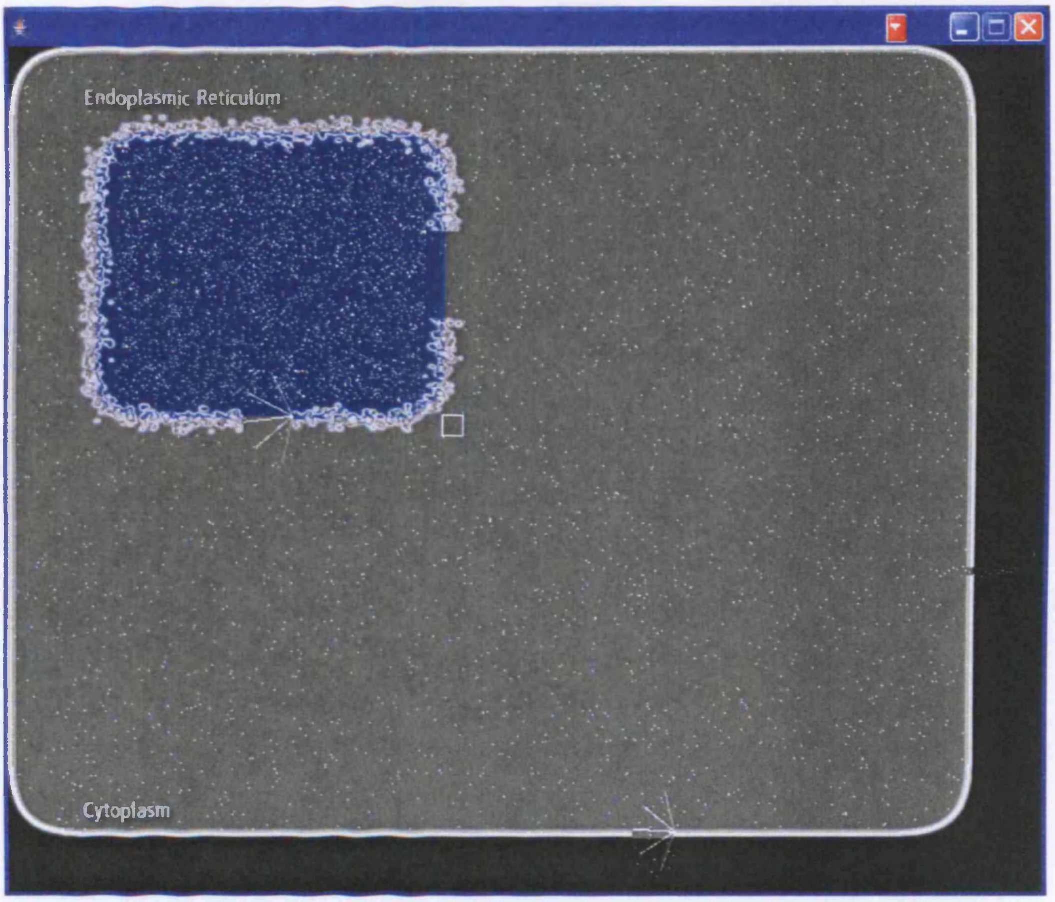


Fig. 7.6: The final version of the CalViz Model Cartoon. Magnified channels not shown.

bringing the design ideas to fruition. During the last stages of prototyping, J., one of the Beacon team programmers, took an active role in programming the software and was instrumental in finishing the software release.

Release of a version marked the completion of the SoViz design process, after all cycles of SoViz were completed, at least until further refinement of the software might be envisioned. Figure 7.6 shows the final version of the CalViz Model Cartoon. It shows the endoplasmic reticulum as a rounded blue square with a convoluted perimeter, situated within the larger rounded square of the cytoplasm. Small white dots represent calcium ions which move between these two structures via channels and pumps. The channels are visible as gaps on the right borders of the ER and the CYT. Depending on the rate of flow across the calcium channels, some of the dots move across these gaps, at a rate which is tied to the underlying data generated by the model. The calcium pumps are represented by converging lines at the bottoms of these structures, which are animated in a clockwise, rotary movement. The key component of this computational model is the ratio of the size of the ER to the Cytoplasm.

The primary user interaction provided by CalViz is that the ER can be resized by selecting and dragging the small square handle at the lower right of the ER. This causes the visualization to change the representations of calcium ion density and flow rate, based on the underlying computational model. In terms of user interaction, it is important to reiterate that the calculations of the calcium oscillation model require many hours to execute. They are not calculated by CalViz in real time. Therefore, to permit rapid user interaction, a range of credible values for the ER-CYT ratio was pre-processed and stored as a data file. It is the information in the data file, which is represented by the visualization.

7.3.5. Discussion

As described above, several software prototypes were produced to refine the CalViz concept. These iterative design refinement activities consisted of producing interactive software prototypes, informally evaluating the results within the team, sketching design alternatives, and producing further prototypes. This is described by stages 2 (Visualization Design Activities), 3 (Prototyping), and 4 (Evaluation), of the SoViz model. The result was a visualization release (Figure 7.6) [Source code is in Appendix F. A working version of this visualization is included on the accompanying CD Appendix.].

The examples above show that sketching was frequently used to refine the prototypes after evaluation. Use of the visualization design patterns also occurred, but to a lesser extent. These had been used by team members during the previous design sessions. But unlike Design Session 4, the patterns were not reviewed to evaluate exhaustively each of the prototypes. The focus of activity at this stage was on rapidly solving specific design problems. The patterns were not seen as useful because they were too detailed and J, who provided the most input at this stage, felt that he was already familiar with them enough to apply design ideas to the visualization problems.

During prototyping of CalViz, decisions needed to be made about mapping specific data items to visual representations. To produce sketch 9a in Table 7.1, the mock-up (Figure 7.2), and the prototypes (Tables 7.3–7.8) required determinations about which visual attributes to assign to the variables in the computational model of calcium oscillations. In the early SoViz design sessions, sketching the Model Cartoons concept had allowed these ideas to be explored, but not specified. The mock-up and prototypes required that these choices be made. The process of creating the prototypes also allowed

TABLE 7.10: VISUAL MAPPINGS FOR DATA PARAMETERS IN THE CALVIZ VISUALIZATION.

MODEL VARIABLE	VISUAL MAPPING
1. Cytoplasm	A rounded square in the interface area
2. Endoplasmic reticulum	A rounded square within the cytoplasm
3. Proportion of ER in the CYT	Size of the ER
4. Amount of calcium ions in the cell bodies	Density of white dots
5. ER and CYT channels	Openings in the wall of the rounded squares
6. Open-ness of ER and CYT channels	Width of opening in the wall (right side)
7. Rate of calcium ion flow	Movement of white dots within the channels
8. ER channel	A conveyor belt
9. CYT pumps	Several animated lines with a common origin point
10. Rate of ER and CYT pump flow	Speed of the animation of the ER and CYT lines

the team to evaluate the effectiveness of visual representations as shown by the design decisions. This corresponds to the data mapping stages of the Information Visualization reference models described in Chapter 2. Table 7.10 shows the data mappings which were used for the CalViz prototype.

Although most of these choices (items 1 – 6) were made at the very beginning of the SoViz process when some of the first sketches were produced, other choices about data mappings (e.g. 7, 8, 9 and 10) were only possible after the production of functional prototypes in various stages of completion. This is primarily because the prototypes are animated and interactive displays. It can be difficult to determine whether a particular visual mapping choice will be successful without seeing the results on-screen. For example, the appropriate movement speed for animated objects can only be determined when they are animated on-screen. Thus, although a sketch is useful for creativity, ideation, and design problem-solving during the elaboration phase, certain decisions require a higher fidelity rendering to be produced. This iterative design loop is typical in software development, and is captured by the SoViz framework as Prototyping (Part 3) leading to Evaluating (Part 4), which in turn, can inform decisions in further iterations of visualization design activities (Part 2).

Evaluation and visualization design activities often occurred during the same discussion. It can be seen from the production sequence in Table 7.9, that as prototypes were produced they could be evaluated and new design decisions could be taken. Very often, this evaluation occurred as 'this works' and 'this doesn't work' decisions, which were followed by more sketching activity to try to resolve a design problem. For example, as noted above, the designs of calcium pumps and channels required similar but different visual metaphors, which were arrived at through sketching. This was the result of an evaluation which determined that the animation of the calcium ions was ineffective without also showing the channels and pumps differently.

The Information Visualization tool in Figure 7.6 was judged by the Beacon team to be a final release candidate of a Model Cartoon of calcium oscillations. The fact that this prototype was successfully produced in part reflects the effectiveness of the SoViz approach for generating a visualization. However, this does not demonstrate that CalViz is a successful visualization, and if SoViz does not lead to successful visualizations, then its usefulness as a method is clearly limited. To evaluate this, an assessment was performed.

7.4. The CalViz Assessment

7.4.1. Overview

As described in section 7.2, the design team had identified five objectives for Model Cartoons. With the completion of the prototyping (Part 3) and evaluation (Part 4) phases described in section 7.3, the SoViz process was completed. The success of the SoViz approach can be measured by the degree to which CalViz met the design objectives. A simple formulation of this is the general question: *Does CalViz meet the objectives of a Model Cartoon?* To measure this, an assessment was performed among graduate students, using an interactive demonstration of CalViz followed by a questionnaire and a survey. Feedback obtained from the participants was used to evaluate the success of CalViz and to identify areas for improvement. This assessment should not be confused with the evaluation (Part 4) stage of the SoViz process. The assessment reported here was conducted to measure the success of the SoViz process as a whole in producing a visualization and not to measure the effectiveness of the design prototyping phase (Part 3) alone.

By creating the prototype, the first of the five objectives was met. A project team review was used to demonstrate the calcium oscillations to other Beacon Team members, who regarded the work as useful for demonstrating the Model Cartoons concept and useful for (1) communicating the work of the Beacon Project, both among team members and outside the project. In addition to being used internally, CalViz can be presented in academic communications and shared with others on the project website. Academic publications in the domain of computational biology are frequently filled with mathematical equations. Pictures and drawings are seen as a way of making the underlying models more tangible and accessible. This is particularly important for obtaining financial funding from grants. For this reason, the senior project leaders and principal investigator hoped that the animated, interactive model would help users to (2) enjoy the model, because it is pleasant to look at and (3) provide a qualitative experience of the models. Because computational models are typically presented as mathematical equations, the experience of the models as equations is qualitatively different from the experience of an animation representing the results of those equations. As identified in Design Session 2, it was expected that this qualitative experience would have two benefits. These are that it would (4) aid decision making, because the animation (5) makes visible the nature of cause-and-effect. Although this can be understood through study of the equations and analysis of their results as data and x-y charts, an interactive animation of this output more clearly demonstrates the concepts surrounding calcium oscillations.

As noted in section 7.2, the project team had also observed that CalViz might make an effective teaching tool. Therefore, two additional objectives related to teaching were added to the assessment. One was to measure (6) whether the visualization enhanced communication about calcium oscillations between the lecturer and the participants. A final question was (7) whether the visualization motivated student interest in Model Cartoons.

7.4.2. Expectations

The expectations were that CalViz would make the computational model of calcium oscillations easier to understand by visually representing its elements on-screen and making the ER-CYT ratio adjustable in real-time. Participants in a study would score better on a questionnaire comprised of questions pertaining to calcium metabolism after interacting with CalViz. They would also find the visualization enhances understanding of traditional materials and they would be more enthusiastic about the subject. This would be demonstrated by answers to an opinion survey about CalViz.

7.4.3. Participants

A representative sample of all of the possible users of CalViz would include computational biologists within and outside the Beacon team, students, and research scientists in other disciplines. As such a sample was not possible, a less representative but easily accessible sample was drawn from MSc students in Computational Biology at the UCL Centre for Mathematics and Physics in the Life Sciences and Experimental Biology (CoMPLEX). These students were at the beginning of their course and had some exposure to the details of calcium oscillations. It was expected that they would have some familiarity with the concepts and so were deemed good candidates for evaluating CalViz. The students were invited to participate and advised that they would be part of a software assessment which would have no bearing on their marks for the course. They were also permitted to opt out. Nine student volunteers granted consent to participate and were offered no compensation.

7.4.4. Administration and Measures

The assessment was administered in a seminar room at CoMPLEX. It consisted of a half-hour seminar given to the volunteers, who completed a questionnaire and an opinion survey to measure the objectives. The 'Learning Questionnaire' was devised to assess how well CalViz achieved objectives 4, 5, and 6 for Model Cartoons by engaging participants' interest in the topic by asking them to solve problems about calcium oscillations. It was also devised to measure their understanding of calcium oscillations concepts. The 'Opinion Survey' was designed to assess how well CalViz achieved 2, 3, and 7 of the Model Cartoons objectives, by measuring motivation, entertainment, and general opinions about CalViz.

The seminar, presented by J, used PowerPoint slides to present the parameters, model structure, and the typical results that the calcium oscillations model generates. The Learning Questionnaire (Appendix G) was composed of 10 questions designed to measure changes in students' understanding of calcium oscillations. After the seminar, the Learning Questionnaire was given to the participants along with red pens for marking answers. Afterwards, these pens were collected and students were given the opportunity to individually interact with CalViz on computers in the CoMPLEX Lab. The visualization was explained and interactive elements were identified by the lecturer. After approximately 10 minutes, the students were supplied with black pens and asked to make any changes to the answers they had provided on the Learning Questionnaire. Changes in any answers would indicate a change of understanding about calcium oscillations as a result of using CalViz.

These were collected and students were then provided with the Opinion Survey. The Opinion Survey (Appendix G) was composed of 10 questions which were used to measure motivation about the subject and the visualization, based on five research questions about the qualitative experiences participants had, using CalViz:

1. Does this motivate participants?
2. Was the visualization entertaining?
3. Was the visualization valuable?
4. Did the visualization improve students' subjective opinions of learning the system?

5. Was the visualization usable?

Each of these research questions was measured by two different survey questions which were randomized and when appropriate, oppositely worded. A Likert scale (1–5) was used to collect responses. The form also contained an open-ended question for registering other opinions about CalViz. Following completion of the Opinion Survey, students were invited to participate in an open discussion about their experiences with CalViz.

7.4.5. Results

The results of the Learning Questionnaire showed a modest increase in the number of correct answers given by the participants having used CalViz. Given the small number of participants, a statistically valid sample was neither possible nor warranted. However, the mere fact of having resulted in a greater number of correct answers is valuable in terms of the communication objectives of a lecture. This tentatively positive interpretation is made more compelling in light of the results of the very favourable qualitative responses.

The scores for the Learning Questionnaire are shown in Table 7.11. There were nine participants (n=9) in the study and 10 questions on the Learning Questionnaire. Of these

TABLE 7.11: CHANGED RESPONSES TO THE LEARNING QUESTIONNAIRE AFTER USING CALVIZ (N=9).

	Changed responses resulting in incorrect (-1) and correct (1) answers									
Question	1	2	3	4	5	6	7	8	9	Changes
1. How does the calcium oscillation period in this system change as the size of the ER increases?		1					1			2
2. What elements are in the model (circle all that apply)?							1			1
3. The flow rate of calcium through the membrane pump affected by: (select all that apply)	-1	1		-1	-1	-1				5
4. During what proportion of the cycle is the cytoplasm calcium increasing when the ER size is at maximum?		1			1		1			3
5. How is this proportion affected by reducing the size of the ER?	1						-1			2
6. Is the concentration of calcium in the ER normally greater than that in the cytoplasm?										0
7. How many channels are there on the ER membrane?		1								1
8. Net flow through the ER calcium channel(s) is:	1			1		-1	1			4
9. The openness of the ER calcium channel(s) is determined by: (select all that apply)										0
No. of changes resulting in a correct answer	2	3	0	1	1	0	4	0	0	11
No. of changes resulting in an incorrect answer	1	0	0	1	1	2	1	0	0	6

10 questions, nine were about details of the calcium oscillations model. The results were that out of 81 possible correct answers (9 subjects x 9 questions), 40 incorrect responses were given. After using CalViz, 11 answers were changed which resulted in a correct response, while only six were changed which resulted in an incorrect response. The largest number of changes was for Question 3.

Question 10 was about the difficulty of determining the answers to these questions. The result was that seven participants found the questions to be of 'medium difficulty' and three found the questions to be 'hard'. Question 10 is not shown in the table because it does not result in a correct or incorrect answer.

The results of the Opinion Survey were generally positive. Profiles of the responses for each of the opinion questions are presented in Table 7.12. The survey used a Likert scale where 1 = Tend to disagree and 5 = Tend to agree. The histograms show the number of participants responding for each opinion. For example, on question 1, three people selected 5, Tend to Agree.

The end of the Opinion Survey contained the following open-ended item: 'It would be helpful if you would provide any other thoughts you have about using this visualization'. All responses to this item are presented in Table 7.13.

TABLE 7.12: PARTICIPANTS' RESPONSES TO THE OPINION SURVEY (N=9).

SURVEY QUESTION	RESEARCH QUESTION	NUMBER OF RESPONDENTS FOR EACH OPINION: TEND TO DISAGREE				TEND TO AGREE
1. I would be more inclined to further my studies of this system as a result of using this visualization.	Does CalViz motivate students?		2		4	3
2. Visualizations like this make computational biology more interesting.					5	4
3. I liked using this visualization.	Was the visualization entertaining?				5	4
4. I thought this visualization was boring.		3	3	3		
5. I would like to see more visualizations of cell physiology	Was the visualization valuable?				6	3
6. This visualization was helpful for me					7	2
7. I felt that I understood calcium oscillations better after using the visualization.	Did CalViz improve students' subjective opinions of learning the topic?			2	5	2
8. I think this sort of visualization makes the subject easier to learn.					4	5
9. I thought the visualization was confusing.	Was the CalViz interface usable?	4	5			
10. I thought the visualization was easy to use.					7	2

TABLE 7.13: PARTICIPANTS' RESPONSES TO THE OPINION SURVEY, OPINION QUESTION (N=9).

'Might be nice to label the pumps and channels.'
'Nice model. It would be good if it allowed you to change parameters on the fly (other than ER size) as well.'
'Easy enough, although could do with a visual indicator to signify where a cycle was considered to start.'
'Some features such as the ion pumps could be easier to pick out (e.g. by being another colour) as I found it slightly hard to see what they were doing simultaneously and in relation to the white calcium ions.'
'Might be more useful if it combined visualization with data, e.g. the period of the cycle, proportion that Ca+ is increasing etc.'
'Good visualization – as long as it's been explained already – without having already seen the oscillation waveforms and heard the overview of what it shows, it could be a bit confusing. Loved the pumps!!'
'Magnification of calcium channel opening could be clearer. Might be useful to see graph as an inset (i.e. plot of cytoplasmic calcium concentration).'

7.4.6. Discussion

The previous section described how CalViz met the objective of (1) *communicating the work of the Beacon Project.* The assessment addressed the other objectives (2– 7). The most valuable outcomes from this experiment were determining whether CalViz meets the requirements for Model Cartoons, results in improved communication and understanding of the calcium oscillations model, and enhances student motivation. This can be interpreted from the results of the questionnaires and the opinions rendered in the post-evaluation discussion. It was not possible to collect a sample large enough to generate statistical validity. Furthermore, in this study even a modest improvement is a positive result. In terms of SoViz, to simply show that the visualization is an effective Model Cartoon is sufficient to draw positive conclusions. The quantitative results of the surveys combined with participants' qualitative opinions about the visualization meet these requirements.

The quantitative measures from the Learning Questionnaire mainly concerned objectives 4– 6. One pitfall would be to interpret the quantitative results as evidence of learning. The naming choice of the Learning Questionnaire is unfortunate and perhaps misleading. In fact, the questionnaire was designed to engage participation with problems about calcium oscillations. Evaluation of learning is a complex and difficult subject, and outside the scope of this assessment. It is difficult to know whether participants actually learned anything from using CalViz. However, what is known is that participants did try to solve problems about CalViz, and used the visualization to change some of their answers. In terms of the objectives of Model Cartoons, this implies that CalViz can be used for (4) *decision-making about the models* and to (5) *understand cause-and-effect* in the models of calcium oscillations.

Interestingly, it appears that CalViz proved confusing regarding what affects the rate of flow through the membrane pump. Four of the incorrect answers were given on this question (3). This is somewhat problematic, since one of the objectives was to demonstrate that the size of the ER affects this flow rate. However, there are no elements in the visualization which explicitly demonstrate that only the calcium concentration in the cytoplasm affects the flow rate through the membrane pump. It may be that a lack of explicit causal indicators in the visualization led to confusion among respondents. Also, there is no way to test that calcium concentration is the sole causal variable in the visualization, though the question is worded as 'select all that apply'. This makes systematic elimination of possible answers impossible. In fact, this may be a poorly designed question, since participants could not use the visualization alone to determine the answer.

Questions 1, 3, 5, 8 and 9 concern the cause-and-effect relationships in the calcium oscillations model. The results were mixed. These would confirm that CalViz clearly showed a cause-and-effect relationship among the model variables: among these questions, eight answers were changed after using CalViz. However, because of the poor results on question 3, only five of these changes resulted in correct answers. With these results, and the lack of direct decision-making measures in the study, it is difficult to determine if SoViz clearly supported the objective, of the Model Cartoons – (4) *aiding decision-making*.

Question 10 poses problems for interpretation. It was worded as: 'Regarding determining the correct answers in the above questions, circle the answer corresponding to your opinion of how difficult this was: (a) easy, (b) medium difficulty, (c) hard'. A problem with this question is that it has no correct answer. However, three respondents changed their answers after using CalViz. It is interesting that all of them thought that the learning questionnaire was of 'medium difficulty' after using the visualization, having changed their answers from 'hard'. This may indicate that they thought the questions were easier to answer after using the visualization, but it is difficult to know from the design of the assessment. This was more properly an opinion question, which probably would have been more suitable for the qualitative measures of the Opinion Survey.

The strongest results were in the qualitative measures, though these questions were weaker in their relation to Model Cartoons' objectives 2, 3 and 7, for which they were designed. The subject matter is often described as 'difficult' or 'challenging' by students. Particularly for those who are grappling with the complex equations involved, it is useful to have a visual representation to support understanding of the physiological processes. CalViz supported this understanding and, moreover, strongly supported participant motivation about the topic on all five quantitative measures of motivation. Participants clearly enjoyed using the visualization, and reported a positive experience in Research Questions 1 (4.17) and 2 (3.48). This meets the Model Cartoons' objective of (7) *increasing motivation* about the subject of calcium oscillations, and by extension, computational biology in general, and perhaps (2) *being pleasant to look at*, though stronger support for both was provided by the verbal opinions rendered in the post assessment discussion.

This feedback showed that a better understanding was seen by participants as increasing motivation. One participant stated, 'If you can't figure out what's going on, you're not going to be interested in the subject'. The visualization was seen as a welcome addition to the traditional materials used in the lecture, i.e. the parameters, equations, and numerical results of the models. It enhanced participants' subjective experiences of learning about calcium oscillations. One said, 'It stops you falling asleep!' They also felt that it was much better to have personal interaction with the interface rather than simply watching a demonstration by a lecturer. In addition, students felt that they were more likely to remember the visualization than the static graphs which had been presented in the lecture.

Further evidence of increased motivation was that many opinions were offered about how to improve CalViz. Their opinions ranged from the very general ones about usability: 'I think it's pretty good'. to quite specific ideas about how to improve the visualization. Some participants indicated that including graphs of the model parameters would be a useful addition to the user interface, an idea that had been eliminated during the design phase. Other areas for improvement suggested colour-coding for the calcium concentrations, and a metaphor for representing calcium channels as a set of weights. The reticulated surface of the ER was also seen as being very similar to the surface of proteins, which could be confusing. The fact that participants were interested in improving the visualization is indicative that they were interested in how it represents the underlying models and how it could be made better.

The third objective for Model Cartoons, (3) *providing a qualitative experience of the models,* is perhaps problematic because all experience can be described as qualitative, and therefore, subjective. A more precisely formulated objective might have been: to provide a positive visual and interactive experience of the models. These could be measured by a questionnaire; indeed, in the qualitative feedback, the participants reported their experience as positive. Participants also felt that CalViz increased their understanding of the calcium oscillation system: 'I think it's really easier to understand what the system is using this kind of a visualization and it's a lot harder when you you're just seeing some channels here and there. It's harder to figure out exactly what's happening.'

Several assumptions were made in the assessment design. First of these was that a questionnaire and opinion survey would accurately reveal participants' opinions about CalViz and that conclusions could be drawn about Model Cartoons. As we have seen, it can be daunting to draw direct conclusions from this limited experiment. Another assumption is that the quality of scientific communication is not solely determined by quality of the resulting learning. It also includes the degree of motivation of the audience, that is, the extent to which people become interested in the subject, and the quality of the experience. Also, scientific presentations are given for a variety of reasons. The most obvious one is to communicate material, but equally important is the desire to motivate acceptance. Some examples include seeking investment, looking for future collaborators, and inspiring students to select the subject for future study. It was assumed that increasing motivation is important.

The use of the MSc student population also affected the assessment design. The students were available only for a limited time, so it was not possible to conduct a one-on-one evaluation with each participant. Also, because the seminar was presented as a group, opinions were limited to a group-feedback scenario – not everyone can speak at once, and time is limited. Inevitably this will lead to both positive and negative opinions not being expressed. Also, students may have been biased in favour of primarily positive opinions because of their relationship with the lecturer.

7.5. Conclusions and Summary

The opinions of the Beacon team and the results of the assessment demonstrated that CalViz was largely successful as a Model Cartoon. In addition to the numerical and quantitative results, the assessment also showed that it can be difficult to evaluate visualizations. Some of the problems arise because visualizations are often designed as exploratory tools with no clear task focus. Without a task-completion measure, a thoroughly convincing evaluation can be difficult. Also, it is often not practical to obtain large samples. As CalViz shows, visualizations are often highly specialized. It would have been very difficult to obtain and coordinate a study with a representative sample of students, research scientists, computational- and bench-biologists for this evaluation. It is also difficult to measure learning. What evidence will show when a participant has learned something? Though tests can be devised, testing has problems in that it may show evidence of correct answers, but may not correctly account for individual differences.

Despite these problems, the assessment showed positive results, primarily in the qualitative measures. This is perhaps not surprising, since it is the richness of the visual experience which has a strong impact on subjective experiences. Yet, combined with the positive evaluation from the Beacon team as a whole, it does tend to support the interpretation the SoViz was an effective approach for supporting the team's visualization design activities because it did lead to a successful visualization. Although direct causal links should be viewed with caution, a positive interpretation can be made with a good degree of confidence.

Consequently, some important conclusions can be drawn from the CalViz experience about the SoViz design process. There was a rational decision-making process in the different stages of design, yet the development of CalViz did not proceed without challenges, which highlights some of the potential problems in the SoViz process. Foremost, it was important to maintain contact among the design team members. Though there was good contact during the initial design meetings, this communication waned during prototyping. Without maintaining this communication, the development process was slowed. There was little opportunity to obtain answers to design questions as they arose. Errors were also introduced, as the misunderstanding about the importance of the ER-to-CYT ratio showed. The time used to produce several prototypes based on this misconception could have been used to solve other prototyping problems. This underscores the need for a participatory approach to the design process. For the Beacon team, this reduction in communication happened because staff had significant time pressures, lack of motivation, personal holidays. Some staff also left the project. The project could not continue in earnest until at least one team member had an opportunity to engage in the design process actively. The most efficiency was added when team member J offered to participate in programming the prototypes.

However, during the programming stage only the key participants with technical skills were responsible for completing the programming work. The results were then evaluated by the rest of the team. It was simply not possible or desirable for the entire team to work together. They did not all have the skills and for such a modest programming task there was no reason for more than one or two people to work on it. This is good in terms of efficiency of coding, but has the problem that it decreases communication among the team members. It was after the prototyping of four examples that the team had to evaluate what was successful about the visualizations and what was not successful. This was because the interactions and visual elements could not be evaluated without being animated on-screen. This may be remedied by other forms of prototyping such as storyboards, but these were not used for creating CalViz.

These challenges do not change the fundamental framework of the SoViz approach. It accurately captured the design activities, sequences, and iterations; it demonstrates that visualization design knowledge and sketching played an important role in realizing the visualization concept. By following the development of an example visualization through the entire SoViz design methodology, we have been able to illustrate each phase of the SoViz process with specific examples. This provides a more detailed perspective on the Action Research intervention and the Grounded Theoretical findings in Chapter 6, by highlighting the activities that are involved in bringing a visualization from early design ideation through prototyping and evaluation. The results showed that by using SoViz it is possible to create a novel visualization which successfully achieved its design objectives.

Chapter 8: Conclusions

8.1. Summary

The primary problem addressed by this research was that current descriptions of Information Visualization do not adequately describe the visualization design process and provide little support for designers. Information Visualization tools offer substantial benefits in understanding and using data. But in order to deliver these benefits, designers must be able to create new visualization tools that will support users in their work. A brief background of visualization was presented and four sources of design knowledge were identified (Chapter 2). The research problem was motivated by the shortcomings in these knowledge sources, which resulted from addressing Research Question 1. The knowledge sources often conflict with one another, offer limited descriptions of the visualization design process, and are widely scattered. They also offer little representation of the user in the design process. This led to the formulation of an additional Research Question (2) to study these deficiencies. Two experiments were performed to investigate two current visualization design techniques (Chapter 3) and to indicate what designers need. The results of these experiments suggested taking a qualitative research approach and led to a literature review on creativity in the design process (Research Question 3). The review revealed that current research in other design disciplines has identified sketching as a strong support for design work because it enhances creativity, design ideation, and problem-solving during design activity (Chapter 4). Patterns were also identified as valuable design tools in both architecture and software engineering. The research question proceeding from this (4) was what the practical results of using these techniques would be for visualization design.

To explore the potential benefits, an experimental case study was performed using qualitative methods. It was guided by an Action Research framework called SoViz, which was created to provide structure for the research and to act as a visualization design method (Chapter 5). There were two major results from the case study (Chapter 6). The first was that a Beacon design team was able to use sketching, along with visualization design knowledge from design patterns and a subject-matter expert, to explore their visualization design problem space and to produce proposals for eight visualizations. As they were 'customers' seeking an approach that would generate results, SoViz was considered successful. The number and quality of the visualization concepts they produced validated the outcomes of the research. This raised a question (Research

Question 5) regarding the theoretical consequences of using the SoViz approach. A Grounded Theoretical analysis of the design process was performed, which produced a second result from the case study. It showed that sketching supported the generation of design ideas, creativity, and problem-solving in the early stages of design activity, whereas design patterns were useful in later stages.

The final part of this research evaluated SoViz as a design method for producing a successful visualization (Chapter 7). To do this, one visualization concept was selected and implemented as software called CalViz. An assessment of this visualization showed that it had achieved the goals which the design team had identified for it during the design process. This indicates that SoViz should be a useful method for future visualization designers.

8.2. Contributions

The contributions of this research are both substantive and methodological. One major and two minor substantive contributions come from the empirical findings and the insights resulting from the research. Additionally, one major and two minor methodological contributions result from the research approach taken for this thesis.

8.2.1. Substantive

The major substantive contribution of this work is the SoViz method for visualization design (Chapter 4).Visualization designers need support in the form of a clearly described process which can lead to creating new interactions and visual representations. They also need techniques which will account for the interactive role of the visualization user. SoViz embodies an approach which can achieve this by enhancing design ideation, creativity, and problem-solving. It is a collaborative and user-centred approach. Although it is likely that many visualization designers already use sketching and design knowledge to produce software, this has largely been undocumented in the literature. SoViz is an attempt to remedy this and to situate the role of sketching activity in descriptions of visualization design methodology. It provides designers with a specific series of activities to undertake which will lead to them producing a visualization. This is particularly appropriate for novice designers, for whom a design process provides guidance that reference models do not offer. Instead of merely describing which parts should be in a visualization system, it describes the process of how to undertake the work. Design activity has directionality, in that it starts with an often ill-defined problem and must result in a concrete solution; it requires methods that support creativity and lateral thinking. By describing a specific design process which supports creativity, SoViz accounts for both of these needs.

The development of this method was based upon the inquiry of the research questions into the current state of visualization knowledge (Chapter 2), the results from two experiments (Chapter 3) and a review of design techniques in other disciplines (Chapter 4). A minor substantive contribution of this research is drawn from the literature review. Visualization knowledge can be categorized into four classes: *Examples* which describe new visualizations, *Taxonomies* which categorize and list visualization artefacts, *Guidelines* which recommend best practices, *Reference Models* which describe the parts of visualization systems and how they work as a whole. This review demonstrated that problems of innovation in visualization stem from the fact that these sources of knowledge alone are insufficient for designers to use as a design method for new visualizations. There is also disagreement about the distinction between Information Visualization and scientific visualization. These factors have led to poor user representation in the design process.

A second minor substantive contribution of this thesis comes from an analysis of the results of two experiments (Chapter 3). These showed that the use of the Visual Information-seeking Mantra may not necessarily lead to more usable visualization software. The interpretation of its concepts is subjective and may not be applicable to all visualizations. These problems were similar to those of other guidelines, which have limited efficacy for designers because guidelines are often vague or even conflicting and do not account for the design context. The second experiment showed that visualization concepts encapsulated in design patterns may also prove confusing for designers.

8.2.2. Methodological

The major methodological contribution of this thesis is the development of a twofold method for studying the visualization design process using Action Research and Grounded Theory. The research has demonstrated that a real-world design situation can be studied in a principled manner, to generate a beneficial effect for both the organization under study and the community of practice. It generated fruitful knowledge about the effectiveness of sketching and design patterns in visualization design activity, using qualitative methods. Such a research approach has been described by Fléchais (2005) in the area of usable secure systems, but it has not been applied to the design of visualization software.

A minor methodological contribution arises from the results of the research, which showed that design patterns and sketching helped in different ways, and to different extents. It also showed that these tools were most appropriate at different stages of the design process. Sketching provided benefits in the areas of design idea generation by supporting the creative work of the Beacon designers. This was particularly important in the early phases of the design work, where generating new ideas was important. Sketching also helped the participants to solve design problems collaboratively, since every team member could engage with the visual elements under consideration. The use of a cartoon to show the action of the computational models was an effective approach for overcoming the design problems associated with understating the results depicted in several *x-y* charts. This was possible because the SoViz method situated sketching not as a secondary activity on the sidelines of design work, but as a primary activity in which all of the participants could be engaged. By incorporating a sketching-oriented approach as the focus of the team's design work, each of the participants had a low barrier to participation. This encouraged design as a social process, with participation of the end-users as well as the other members of the team. The SoViz structure formalized this and provided a roadmap for the work of the project.

Additionally, this research has shown how design patterns help because they structure design knowledge explicitly and can be used to both suggest and eliminate potential design solutions. Because designers found it difficult to identify many patterns consistently, and because of the strong indications of the value of sketching, the research was not explicitly designed to measure the value of visualization design patterns in generating creative ideas. Rather, the focus was on whether the patterns would be usable by the general population of designers from whom they were intended. A result was that some problems were identified with visualization design patterns. Like guidelines, the terminology used in them can be confusing. Patterns also suffer from the fact that they attempt to condense a great deal of design knowledge, with the result that they may end up being used in practice as guidelines, rather than as a holistic knowledge corpus. During the Beacon team's design work, this was essentially how they were used. This situates patterns not as a tool for enhancing creativity in design practice, but as a means of narrowing down design alternatives and raises the question of whether patterns should be seen simply as very detailed guidelines. A counter-argument can be made that design patterns require some training and skill to use effectively, and the Beacon team did not have this. But if training to use them is required, this places a burden on the design team.

It seems unlikely that a design team would invest effort to learn a collection of patterns, when they may be constrained by time and resources, and pressured to simply get on with the design challenges at hand. The Beacon team's experience illustrates this problem. Furthermore, most authors consider that patterns should be able to be constructed into pattern languages in which several related patterns work as a coherent group to provide a solution to a design problem. Yet, as reported by Dearden and Finlay (2006), the effectiveness of this approach and the practical details of using patterns in design projects have received relatively little attention. This possibility was not explored by the Beacon team.

A second minor methodological contribution was that the work resulted in the production of a visualization for a real problem in computational biology: how to explain complex modelling problems to lay-audiences. In the process, it demonstrated how to bring visualization to computational models and showed that such a visualization can produce useful results. The Beacon team also now have a guiding framework for their future efforts to produce new visualization designs.

8.3. Critical Review

In addition to its strengths, with any research there may be areas for improvement. Rather than being seen as deficiencies of the work, however, these can be framed as important steps toward creating a research agenda. A reflection on this research yields five major areas of opportunity and leads toward a research agenda, as described section 8.4.

8.3.1. Critique of research approach

Although post-positivist research has made inroads in the social sciences, it is still relatively new in the domain of computer science. There might be concerns regarding the validity of the qualitative research methods used in this study. However, these philosophical questions are most appropriate in debates about the philosophy of science. Rather than becoming mired in such questions, it is the accepted view among both AR and GT researchers that validity is achieved by determining a method and carrying it out honestly, scrupulously, and meticulously. It is in this spirit that this research was conducted. Setting aside issues about the validity and generalizability of these research approaches, there are areas of possible critique regarding the execution of these methods.

One of the tenets of Action Research is that is should be applied in the context where organizational change is desired. Although the Beacon team agreed that the organizational change they meant to bring about was one where they would be able to understand their visualization design problem better and learn how to approach visualization in the future, this may not be the type of organizational change which AR is best suited to study. Although the philosophy and methods of AR are not explicitly contrary to this application, AR has not previously been used to explore how creative design processes can be brought into an organization for making information visualization software. Mitigating this somewhat, examples from Avison and Wood-Harper (1986) and Baskerville (1999) have used AR to study software systems in the organization and design of new media Information and Communications Technology (ICT) systems, though Baskerville notes (ibid.) that some researchers have criticized the use of AR in information systems.

In terms of Grounded Theory, it is accepted that the investigator is intimately implicated in the theories developed during the research. There can be multiple interpretations of the data and the focus of the research and the biases of the investigator will affect the nature of the findings. A consequence of this is that the GT findings are not generalizable to all design situations; for example, in industrial environments. Yet, although this research presents theoretical development from only one case, it can still have relevance for other cases and organizations. Strauss and Corbin (1998) observe that Grounded Theory identifies concepts and their relationships. In the dataset from this work, there were dozens of coded events which were used to build theory about the role of sketching and design patterns in the design work. If the concepts are at a sufficient level of abstraction they are likely to be present in many design scenarios. From this research, it appears likely that examination of the use of both sketching and design patterns would produce similar categories and relationships in other design contexts.

Another question regarding the research approach is whether the Beacon team had truly sufficient input into the direction of the research. Although every effort was made to ensure that they would take an active role in the research direction, they may have self-censored out of a desire to ensure their time and effort would lead to the production of a visualization tool. Whether this is true is difficult to know and is a limitation of the AR approach in information systems research.

8.3.2. SoViz is a partial solution

As a research framework, SoViz was created to fill some of the methodological knowledge gaps in visualization. As a method, its purpose is in part to support creative activity in the visualization design process, but it is important to stress that the SoViz method is not a replacement for the information visualization reference models. The reference models can be seen as a guide to what must be considered during the design activities. They outline the 'must haves' of visualization systems. They do not describe how best to achieve these. As described by the Information Visualization reference models in Chapter 2, design of these subcomponents involves several steps: cleaning up the raw data to delete redundancies, remove errors, and resolve inconsistencies; storing it as structured data in a consistent format; creating visual mappings between variables and on-screen elements and determining a visual layout; and designing widgets for user interaction. In this research, the visualization reference models were not used explicitly to guide these decisions and they did not occur in sequence. Rather, the design team solved each of these problems as they arose, using the creativity-supporting approach offered by SoViz. This is perhaps an area that could have been addressed more methodically, but the Beacon team did not choose to do so. Other concerns, such as selection of systems technology (e.g. operating system, programming environment, etc.), were addressed by an *ad hoc* approach. For this reason, SoViz is a method to use in conjunction with the reference models and with the expertise of individuals on a design team who can offer knowledge of relevant subcomponents such as database design and computer graphics.

8.3.3. Domain independence

Chapter 2 identified some problems with the distinction between scientific visualization and Information Visualization. Tory and Möller (2004) suggest that the distinction between the domains might be unimportant in terms of categorizing and describing the nature of visualization systems. Similarly, this research has proceeded on the basis that these differences are largely irrelevant in terms of supporting creativity and problem solving in design activity. Also, as shown in Chapter 4, sketching has been used to solve design problems in many different engineering disciplines. It could be argued that any domain of design knowledge could be substituted in the place of visualization design knowledge in the visualization Design Activities (Step 2) of SoViz. If so, what makes SoViz uniquely suited for visualization?

The results of this research tend to indicate that SoViz is important because it actively situates a visual design process (sketching) with a visual experience medium (visualization software). It is, therefore, strongly suitable for the visualization design process. It also serves to engage the visualization community in recognizing the importance of sketching and other user-centred methods. These are known to be successful, as is demonstrated by their value for other design communities and in HCI.

We know of no other design methods in visualization which emphasize the importance of combining existing sources of design knowledge with sketching.

8.3.4. Design expertise

One of the motivations of this thesis has been to address the fact that non-experts have few resources to draw upon when creating new visualization systems. As Chapter 2 showed, the few available sources of information vary greatly in the quality of the guidance they provide. An aim of the SoViz approach has been to describe a design method that would be accessible even for people who are not very familiar with visualization. As Tversky et al. have observed (2003), novices and experts are able to use sketching to make new inferences and to devise alternatives. However, the presence of the author/researcher during the design activities means that it has not been possible to evaluate SoViz as it might be used in the absence of a visualization expert. A potential weakness with SoViz is that the design patterns alone cannot provide the same depth of knowledge in those situations where an expert may not be present. Also, although the literature indicates that design patterns have been used by non-experts to achieve system designs in other areas of software engineering, this has not been evaluated in the context of visualization design. Other sources of knowledge from the growing visualization literature may also be needed to provide support. The results of this research must be interpreted with this perspective in mind.

Although SoViz may be a useful approach for novices with regard to visualization, there is some evidence suggesting that experienced designers may act differently. Suwa, et al. (2001), found that experienced architects tended to regroup sketched artefacts differently from novices, and that those who adopted sketching produced more interpretations of ambiguous sketches, i.e. sketching helped novices generate more design output. It would be interesting to study whether this also holds true in the domain of visualization and might have important consequences for the usefulness of the SoViz approach.

In addition, the SoViz *research framework* that was used to guide this research employed sketching and design patterns. But the role of the researcher, a source of visualization expertise, was not specified in Part 2, the visualization Design Activities phase (Figure 8.1a). Yet, as shown in the case study, the subject-matter expert significantly affects the outcomes of the design work. When considering the SoViz approach not as a research framework but as a visualization design method, the role of all sources of knowledge should be accounted for. This perspective reframes the design

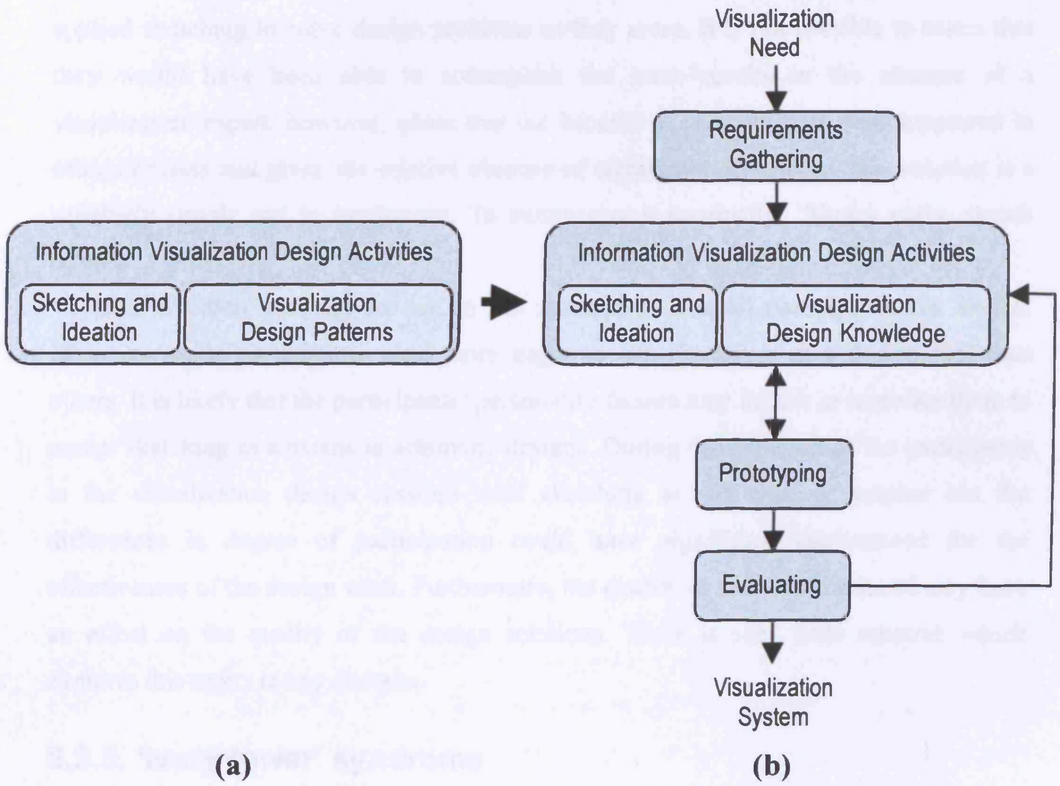


Fig. 8.1: Reinterpreting the SoViz research framework as a visualisation design method. Design knowledge may come from many sources other than patterns.

knowledge captured by patterns as part of a larger knowledge corpus, which includes assistance from an expert designer, examples, guidelines, publications, or other sources. A modified depiction of this at a higher level of abstraction (Figure 8.1b) still provides enough detail to be used as a design method, but does not limit this method to using only visualization design patterns.

This research has not explored whether the SoViz research framework can be used effectively in the absence of a source of expertise. What should novices who want to adopt SoViz do if they have no access to a visualization designer? The answer to this is to assemble the available visualization design knowledge and to apply this knowledge in a methodical and consistent way, using the benefits of group collaboration and sketching. This should be done in an iterative process. As shown with the Beacon team, the contribution of sketching is that it aids creative output and team communication. The practical result for the Beacon team was that sketching led to a great deal of creative design output. At the beginning of the process, the team did not have any design ideas. By the end, they had produced many alternatives and as they began to build a prototype, they applied sketching to solve design problems as they arose. It is not possible to assert that they would have been able to accomplish the same results in the absence of a visualization expert, however, given that the benefits of sketching are well supported in other domains and given the relative absence of significant drawbacks, this solution is a relatively simple one to implement. To summarize it succinctly: 'Sketch early, sketch often'.

This research was carried out on the assumption that all participants can sketch. However, some participants were more eager to use sketching as a design tool than others. It is likely that the participants' personality factors may inhibit or motivate them to accept sketching as a means to achieving designs. During the work, all of the participants in the visualization design sessions used sketching at one time or another but the differences in degree of participation could have significant implications for the effectiveness of the design work. Furthermore, the quality of sketches produced may have an effect on the quality of the design solutions. There is very little research which explores this topic, in any domain.

8.3.5. 'Ivory tower' syndrome

An obvious problem with this research is that although the Beacon team did have a *bona fide* design problem, their project existed within a university research institution rather than a commercial environment. While it should prove useful within other academic environments, it is not clear that this could be easily translated to the commercial sphere. Commercial organizations have their own established development practices and design cultures, within which the SoViz approach may not be compatible. Also, the SoViz process required that the Beacon organization was willing to change its development process to engage stakeholders from all parts of the project. In other organizations there may not be flexibility to integrate team participation into the design process. Particularly within commercial environments, changes with uncertain outcomes can be seen as risky. However, there are counterbalances to these considerations. As shown in Chapter 4, sketching has been successfully used in commercial design practices for a long time. To a lesser extent, design patterns have been taken up by software developers in both academia and industry. These factors bode well for the using the SoViz approach for visualization, which is undergoing increased commercialization.

8.4. Directions for Future Research

These contributions can be used to establish a meaningful research agenda for future inquiry in Information Visualization in several areas. These relate to the practical application of SoViz in academia and commercial organizations, theory development and pedagogy.

8.4.1. SoViz 'in the wild'

There are several paths for exploring the practicalities of applying SoViz. Evidence from this research showed that there can be problems representing interactions using both sketches and patterns. Interactions imply narratives, but pictures are static – taken alone, they may not tell the whole story. One way to continue this line of enquiry would be to examine the role of narrative in design of visualizations and to study supporting tools for this. Storyboards have proved useful in other areas of HCI because they show scenarios of interaction. They allow designers to suggest alternatives and permit exploration of multiple outcomes from interactions. Buxton (2007) has shown how storyboards and models provide substantial aid in the design process. Preece (1994), Norman (1988), Snyder (2003) and others have shown this in HCI. Practitioners in architecture and design also use physical models to aid the design process. Since visualizations are interaction-based and thus, time-based, the relationship of sketching, storyboards, and models bears further research. Should storyboards be interpreted as active sketches? Is there a role for a kind of 'active' pattern which demonstrates the key concepts at an abstract level, using animation? These questions remain to be answered.

As described above, this work explored the relationships of sketching and visual mappings as they were limited to a particular context, namely computational biology. One of the areas not explored explicitly was the effectiveness of the use of existing visualization design knowledge for determining these visual mappings. Although suitable mappings for data types have been identified by MacKinlay and others, and there are also automated systems for doing this, there are few details of how to apply them to a specific visualization problem. During the creation of CalViz, many decisions were made about which variables would be represented as visual attributes. The discussion of this work shows that this did not occur as a rigorous and methodical process, but instead happened iteratively, as the prototypes were refined into a final visualization. It would be interesting to know if this is typical of the visualization design process, but just not captured in the literature. Do designers determine mappings on the fly and according to gut instincts? Since consideration of data type (quantitative, ordinal, nominal, etc.) mappings was not a focus of the research, it is difficult to answer such questions with the results of this study. It is not known whether designers relied principally upon sketching, other sources of knowledge, or upon their own experience. Such knowledge might show how sketching should be used during this key visualization design activity.

An important question was raised about validation of the research approach. This research has been narrowly focused on the benefits of sketching in a small research setting with less than 10 stakeholders; to enhance its value for practitioners, the larger organizational environment must be taken into consideration. It is not clear that SoViz can be applied on more complex visualization projects with large teams. Indeed, sketching literature in the wider domain of HCI has not adequately addressed the issue of scalability of many design techniques. For SoViz to have wider applicability, its use should be investigated 'in the wild', in larger projects and in industry. There are many organizations (e.g. SpotFire, The Hive Group, Oculus) which market visualization tools and consulting services as a primary source of revenue and which contribute to the body of visualization knowledge. Future research in such small- and medium- enterprise companies would identify whether SoViz is useful in these contexts and where there might be areas for improvement.

One way to address this is to cast SoViz in light of organizational goals and objectives. This stance would look at visualization design not just as a software solution; it would involve integrating theories of problem-solving and design in the organization as a whole, including marketing and sales, engineering, business management, etc., just as software engineering development models can be considered from the perspective of business processes. This bears similarities to Buxton's approach, as his work draws predominantly for results produced in industry rather than research conducted in academia. He argues that sketching is useful in the design process on the micro level, but moreover, his work is an attempt to integrate this into a macro-level view of the whole organization. Sketching yields micro-level solutions such as creativity in design, but in industry it must be seen as part of the software engineering. As he notes:

> ...one of the most significant reasons for the failure of organisations to develop new software products in-house is the absence of anything that a design professional would recognise as an explicit design process. (Buxton, 2007)

Sketching is one of the tools that this process uses. Some expert designers in industry have used as similar approach. Brath (2003) has proposed 'paper landscapes', which are paper-based tools which summarize design proposals for visualizations. These structured documents contain both a business rationale and design rationale for a visualization, including its visual elements, interactions, sample workflows, goals and technical limitations. As reported by Brath, they are intended as both a design description and a visualization design method. They may use both sketching and wireframe representations to document a visualization design. Although Brath describes their use in industry for practical visualization design problems, the reasons that they are successful are not thoroughly reported and no experimental results are provided. Though they use sketching and other sources of design knowledge, the rationale for these receives small attention. Yet, the fact that there is some precedence for a sketching-oriented approach supports an interpretation that SoViz can be valuable for both novice and expert visualization designers.

8.4.2. Theory development

In the realm of theoretical contributions, anecdotal evidence from industry alone is insufficient to add a meaningful contribution to the body of knowledge. To address this, one future research approach to investigate the usefulness of SoViz would be to conduct a protocol analysis of designers' activities, following the method of Suwa et al., (2001). They have used retrospective protocol analysis wherein individual designers review a video of their own sketching during a design problem-solving activity. The designers can then identify aspects of their own design behaviours and thinking. The data generation phase of this research actively included the participants as co-researchers and so, including them in the data analysis process is appealing from both a practical and a philosophical perspective. Their participation would allow further insight into the data than can be obtained by the retrospective analysis of the researcher alone. In this work, the data analysis methods did not lend themselves to including the Beacon team members in the labour-intensive data analysis process (nor would they have been inclined to do so!). Though the Beacon project team dispensed with video recordings as too intrusive, a similar approach could be used to study sketching in the visualization context. This might yield the sort of insight into the design mechanisms of sketching that Suwa, et al., refer to as visuo-spatial and metaphoric calculation, but specific to visualization design problems.

The research approach of this work raises the issue of the value of knowledge generated by AR investigations and the question of how many studies need to be done for others to benefit. As is suggested by the AR literature, the philosophical stance is that AR applied to a research domain is a continuous process of refinement, which contributes to the body of knowledge, rather than just one or two studies. Before a clearer picture can emerge, many years of contributions will have to be made. This is both a limitation regarding the findings of this thesis and an argument for continued investigation in this area. A research agenda for visualization should incorporate further exploratory and qualitative approaches, similar to those which have been adopted in Information Systems and HCI.

In terms of qualitative research in visualization, the focus of the GT analysis in Chapter 6 was on the effects of using sketching and using design patterns, but not on creativity itself. The assumption was that, based on the significant evidence presented in Chapter 4, sketching would naturally lead to creative outcomes. Indeed, the group generated eight new ideas, which can be seen as aggregates of many smaller creative design decisions. But it would be theoretically valuable to review the data with an emphasis on creativity and lateral thinking, rather than on the effects of using sketching and design patterns. Further application of qualitative methods could be to use a combined AR + GT approach to study another visualization design project with an emphasis on how creative acts are manifested in the design process. This question would seek evidence of the following: support for long-term and short-term memory, re-combination of elements, and cycles of generation and reinterpretation.

Even very good research can be hampered by poor reporting. An important conclusion from this research was that visualization researchers are not telling the whole story: several shortcomings in theoretical knowledge as it is applied to practice were identified during literature review. That this is so should not be surprising. IV is a relatively new, interdisciplinary discipline, and thus there is only a relatively small body of extant research. But the published literature represents only one source of knowledge. A future research approach should examine IV design activity as reported by practitioners, through personal communication, surveys, or other means.

Furthermore, a gap must be bridged between visualization design theory and design practice, as part of an ongoing research process. Like other human activities design is a difficult (if not impossible) process to describe fully. As shown in Chapter 4, opinions about the general nature of the design process vary widely. Yet the end result of design activity must be a concrete solution. Although there are many ways to build an international airport or flight booking system, the particular problem situation will determine how these are realized. This creates a tension between describing the attributes of visualizations at a high enough level of abstraction to be valuable as theory and providing practical design methods. It accounts for the use of reference models to describe the high-level attributes of visualization systems, rather than prescriptions of a procedural design method. The focal point of such tensions is in the context of teaching new methods and techniques. It is here that the visualization community has a great deal of work to do. For visualization to be useful to other communities of practice, clear methods (and more of them) must be explored and taught. This is precisely the pedagogical approach that has been adopted by engineering and design disciplines. These have fostered design exploration and creativity-enhancing techniques, such as sketching, studio design training, and group critique. The practical realization of this would be use of a similar teaching approach in visualization workshops and academic curricula. This would be instrumental in making visualization relevant and enhancing its value for academia and industry, leading to future advancements. New knowledge from such advancements will lead to new research.

This work cannot claim to have resolved the tensions between theory and practice, because such will always exist. However, by identifying current limitations, it forms a basis for continued research. Just as other disciplines cannot describe specific design procedures for every circumstance but do have processes in place to direct design activity towards a concrete solution, so can visualization draw upon creative methods for design problem-solving. This can broaden the value of visualization discipline, leading to new A-ha! moments, new visualizations and new insights.

References:

Ahlberg, C., Williamson, C., and Shneiderman, B., (1992). Dynamic queries for information exploration: An implementation and evaluation, *Proceedings of ACM SIGCHI'92*, pp. 619–626.

Alexander, C., et al., (1977). A Pattern Language: Towns, Buildings, Construction. Oxford: Oxford University Press.

Alexander, C., (1964). Notes on the Synthesis of Form. Cambridge, Mass.: Harvard University Press.

Alexander, C., (1979). The Timeless Way of Building. Oxford: Oxford University Press.

Alexander, C., Silverstein, M., Angel, S., Ishikawa, S., & Abrams, D., (1975). The Oregon Experiment, New York: Oxford University Press.

Avison, D., Lau, F., Myers, M., and Nielsen, P. A., (1999). Action Research, *Communications of the ACM*, vol. 42, pp. 94–97, 1999.

Avison, D. E. and Wood-Harper, A. T., (1986). Multiview—an exploration in information systems development. *Aust. Comput. J.* 18, 4 (Nov. 1986), 174–179.

Baskerville, R., (1999). Investigating Information Systems with Action Research, *Communications of the Association for Information Systems*, Vol. 2, Article 19, October 1999.

Baskerville, R. and Wood-Harper A.T., (1998). Diversity in information systems action research methods, *European Journal of Information Systems*, Volume 7, Number 2, 1 June 1998, pp. 90–107(18)

Baudisch, P., Lee, B., and Hanna, L., (2004). Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view. In *Proceedings of the Working Conference on Advanced Visual interfaces* (Gallipoli, Italy, May 25 –28, 2004). AVI '04. ACM Press, New York, NY, 133–140.

Baumgartner, J. and Börner, K., (2002). Towards an XML Toolkit for a Software Repository Supporting Information Visualization Education In *Proceedings of IEEE Information Visualization Conference,* (Boston, MA, 2002).

Bederson, B. B., (2000). Fisheye menus. In Proceedings of the 13th Annual ACM *Symposium on User interface Software and Technology* (San Diego, California, United States, November 06 – 08, 2000). UIST '00. ACM Press, New York, NY, 217–225.

Bederson, B. B., (2001). PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology* (Orlando, Florida, November 11 – 14, 2001). UIST '01. ACM Press, New York, NY, 71–80.

Bederson, B. B., Clamage, A., Czerwinski, M. P., and Robertson, G. G., (2003). A fisheye calendar interface for PDAs: providing overviews for small displays. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 – 10, 2003). CHI '03. ACM Press, New York, NY, 618-619. DOI= http://doi.acm.org/10.1145/765891.765893

Bederson, B., and Shneiderman, B., (2003). The Craft of Information Visualization: Readings and Reflections. San Francisco: Morgan Kaufman/

Bertin, J., (1967). Semiologie Graphique. Paris: Gauthier-Villars.

Bertin, J., (1981). Graphics and Graphic Information Processing. New York: de Gruyter.

Boden, M., (2004) The Creative Mind: Myths and mechanisms. London: Routledge.

Bohm, D., (1996). On Creativity. New York: Routledge Classics.

Borchers, J.O., (2000a). A Pattern Approach to Interaction Design. New York: John Wiley & Sons.

Borchers, J.O., (2000b). Interaction Design Patterns: Twelve Theses, CHI'2000: Workshop on *Pattern Languages for Interaction Design: Building Momentum*. The Hague, Netherlands, 2–3 April 2000.

Brath, R., (1999). Effective Information Visualization: Guidelines and Metrics for 3D Interactive Representations of Business Data, Masters of Computer Science Thesis, Graduate Department of Computer Science, University of Toronto, Canada.

Brody, H, Rip MR, Vinten-Johansen P., Paneth N., Rachman S., (2000). Map-making and myth-making in Broad Street: the London cholera epidemic, 1854, *The Lancet*, Vol. 356, No. 9223. (1 July 2000), pp. 64–68.

Brooke, J., (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester & A. L. McClelland (eds.) In *Usability Evaluation in Industry*. London: Taylor and Francis.

Buxton, W., Lamb, M.R., Sherman, D., and Smith, K.C., (1983). Towards a comprehensive user interface management system, *SIGGRAPH Comput. Graph.* 17, 3 (Jul. 1983), 35–42.

Buxton, B., (2007). Sketching User Experiences: Getting the Design Right and the Right Design. San Francisco, CA: Morgan Kaufman.

Card, S., Mackinlay, J., and Shneiderman, B., (1999). Readings in Visualization: Using Vision toThink. Morgan Kaufman, San Francisco, California.

Carr, D.A., (1999). Guidelines for Designing Information Visualization Applications, *Ericsson Conference on Usability Engineering '99*. Stockholm, Sweden, 1–3 December 1999.

Checkland, P., (1981). Systems Thinking, Systems Practice. Chichester, UK: J. Wiley.

Checkland, P. and S. Holwell., (1998) Information, Systems and Information Systems: Making Sense of The Field. Chichester, UK: John Wiley.

Chi, E., (2000). A taxonomy of visualization techniques using the data state reference model, *Proc. of the Symposium on Information Visualization* (InfoVis 2000), pages 69–75. IEEE Press, 2000. Salt Lake City, Utah.

Chi, E., (2002). A Framework for Visualizing Information. Netherlands: Kluwer Academic Publishers, April 2002.

Csallner, C., Handte, M., Lehmann, O., Stasko, J., (2003). FundExplorer: Supporting the Diversification of Mutual Fund Portfolios Using Context Treemaps, *Infovis*, p. 26, 2003 *IEEE Symposium on Information Visualization.*

Csinger, A., (1992). The Psychology of Visualisation. University of British Columbia, Department of Computer Science. Technical Report; TR-92-28.

Csikszentmihalyi, M., (1996). Creativity: Flow and the Psychology of Discovery and Invention. New York: Harper Perennial.

Dearden, A.; Finlay, J.; Allgar, E.; McManus, B., (2002). Using Pattern Languages in Participatory Design, *Proceedings PDC 2002* (Malmö, SWE, 23-25 June 2002), 104–13.

Dearden, A., and Finlay, J., (2006). Pattern Languages in HCI: A Critical Review. *Human-Computer Interaction*, Volume 21, pp. 49-102.

Department of Trade and Industry (DTI), (2006). From the DTI web site http://www.beaconprojects.org.uk/viabs.htm (viewed 2 March, 2006).

Dick, B., (1997). *Approaching an action research thesis: an overview* . http://www.scu.edu.au/schools/gcm/ar/arp/phd.html

Dix, A., Finlay, J., Abowd, G., and Beale, R., (1998). Human-Computer Interaction (Second edition). Hertforshire, UK: Prentice Hall Europe.

Eick, S.G., (1995). Engineering Perceptually Effective Visualizations for Abstract Data, In: Nielson, G.M., Hagen, H., Muller, H. (eds.) *Scientific Visualization Overviews, Methodologies and Techniques*. USA: IEEE Computer Science Press. pp 191-210.

Ericsson K. A., Simon H. A., (1993). Protocol Analysis: Verbal Reports as Data, MIT Press, Cambridge, MA.

Fekete, J.-D., (2004). The InfoVis Toolkit. In Proceedings of InfoVis '04, pp. 167–174, 2004.

Fällman, D., (2003). Design-Oriented Human-Computer Interaction. *Proc. 2004 ACM Conference on Human-Computer Interaction* (SIGCHI).

Fincher, S., Finlay, J., Greene, S., Jones, L., Matchen, P., Thomas, J., and Molina, P. J., (2003). *Perspectives on HCI patterns: concepts and tools*. In CHI '03 Extended Abstracts on Human Factors in Computing Systems, Ft. Lauderdale, Florida, USA, April 05 - 10, 2003. (CHI '03). ACM Press, New York, NY, 1044-1045.

Finlay, J., Allgar, E., Dearden, A. & McManus, B., (2002). Patterns in Participatory Design, in X. Faulkner, J. Finlay, F. Detienne (eds.) *People and Computers XVII: Memorable yet Invisible*, Proceedings of HCI'2002, Springer Verlag, 2002, pp. 159–174.

Fléchais, I., (2005). Designing Secure and Usable Systems, PhD Dissertation, University College London, UK.

Foley, J.D., Van Dam, A., (1982). Fundamentals of Interactive Computer Graphics. Boston, Massachusetts: Addison-Wesley.

Furnas, G.W., (1981). The FISHEYE view: A New Look at Structured Files. Murray Hill, NJ: Bell Laboratories.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J., (1995). Design Patterns: Elements of Reusable Object-Oriented Software. New York: Addison-Wesley.

Gabriel, R., (1996). Patterns of Software. Oxford: Oxford University Press.

Goel, V., (1992). Ill-structured Representations for Ill-structured Problems. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum.

Goel, V., (1995). Sketches of Thought. Cambridge, Mass: MIT Press.

Gross, M. D. and Do, E. Y. 1996. Demonstrating the electronic cocktail napkin: a paper-like interface for early design. In *Conference Companion on Human Factors in Computing Systems: Common Ground* (Vancouver, British Columbia, Canada, April 13 - 18, 1996). M. J. Tauber, Ed. CHI '96. ACM Press, New York, NY, 5–6.

Graham, M.J., (2001). Visualising Multiple Overlapping Classification Hierarchies, PhD Dissertation, Napier University, UK.

Heer, J., Card, S. K., and Landay, J. A., (2005). Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA, April 02 –07, 2005). CHI '05. ACM Press, New York, NY, 421–430.

Heiser, J.,Tversky, B., and Silverman, M., (2004).Sketches for and from collaboration, In J. S. Gero, B. Tversky, and T. Knight, editors, *Visual and spatial reasoning in design III*, pp. 69–78.

Herman, I., (1989).2.5D Graphics Systems, In: *Eurographics'89 Conference Proceedings*, Hamburg, eds. W. Hansmann, F.R.A. Hopgood, and W. Straßer, North-Holland (1989).

Hoffman, D., (1998). Visual Intelligence: How we create what we see. New York: W.W. Norton & Co.

Howell, D, (1992). Statistical Methods for Psychology. Boston: PWS Kent Publishing Company.

Johnson, B., and Shneiderman, B., (1991). Tree-maps: A Space-Filling Approach to the Visualisation of Hierarchical Information, in IEEE *Visualisation* 1992, pp 284–291.

Jones, J.C., (1992). Design Methods. New York: John Wiley and Sons.

Kuhn, T., (1970). The structure of scientific revolutions (2nd ed.) enlarged. In *International encyclopaedia of unified science* (Vol. 2, No. 2). Chicago: University of Chicago Press.

Landay, J.A., Myers, B.A., (1995). Interactive Sketching for Early Stages of User Interface Design. Proceedings of CHI 1995 , pp. 43–50.

Lange, S. et al. (1995). Problem-oriented visualisation of multi-dimensional data sets. Proceedings of International Symposium on Scientific Visualisation, Caligari, Italy, 27-29 September, 1995. pp 1-15.

Lamping, J., Rao, R., and Pirolli, P. 1995. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, United States, May 07 – 11, 1995). I. R. Katz, R. Mack, L. Marks, M. B. Rosson, and J. Nielsen, Eds. Conference on *Human Factors in Computing Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, 401–408.

Laseau, P., (1980). Graphic Thinking for Architects and Designers. New York: Van Nostrand Reinhold Company.

Laseau, P., (2004). Freehand Sketching: An introduction. New York: W.W. Norton and Company.

Lawson, B., (1997). How designers think: the design process demystified. Oxford, UK: Architectural Press.

Lidwell, W., Holden, K., Butler, J., (2003). Universal Principles of Design. Gloucester, Mass: Rockport Publishers, Inc.

Lin, J., Newman, M.W., Hong, J.I., and Landay, J.A., (2000). DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design, CHI Letters: *Proceedings of the 2000 SIGCHI Conference on Human Factors in Computing Systems* (CHI 2000).

Mackinlay, J., (1986). Automatic the Design of Graphical Presentations of Relational Information, *ACM Transactions of Graphics*, 5(2), pp 110–141.

McConnell P., Johnson K., and Lin S., (2002). Applications of Tree-Maps to hierarchical biological data, *Bioinformatics*, 18 pp.1278–1279.

Myers, M.D. and Avison, D.E. (eds.), (2002). Qualitative Research in Information_Systems: A Reader, Sage Publications, London.

Mullet, K. and Sano, D., (1995). Designing Visual Interfaces. New Jersey: Prentice Hall.

Newell, A., and Simon, H.A., (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.

Niermann, S., (2005). Optimizing the Ordering of Tables With Evolutionary Computation, *The American Statistician*, Volume 59, Number 1, February 2005, pp. 41–46(6)

Norman, D., (1988). The Design of Everyday Things. New York: Basic Books.

Plimmer, B. and Apperly, M., (2001). Computer-Aided Sketching to Capture Preliminary Designs, 3rd Australia User interfaces Conference (AUIC2002), Melbourne, Australia.

Peters, M. and V. Robinson., (1984) The Origins and Status of Action Research, *Journal of Applied Behavioral Science*, (20) 2, pp. 113–124.

Pfleeger, S.L., (1987). Software Engineering: The production of quality software. Indianapolis: Macmillan Publishing Company.

Preece, J., et al., (1994). Human-Computer Interaction. Harlow, England: Addison-Wesley.

Rheingans, P., Landreth, C., (1995). Perceptual Principles for Effective Visualizations. In: Grinstein, G., Levkowitz, H., (eds.) *Perceptual Issues in Visualization.* Berlin: Springer-Verlag. pp 59–73.

Resnikoff. H.L., (1987). The Illusion of Reality. New York: Springer-Verlag.

Robbins, E., (1997). Why Architects Draw. London, England: MIT Press.

Robertson, P., and DeFerrari, L., (1994). Systematic Approaches to Visualisation: Is a Reference Model Needed? In Rosenblum, R.A. et al., eds. *Scientific Visualisation: Advances and Challenges.* New York: Academic Press.

Roth, S. and Mattis, J., (1990). Data Charactertization for Intelligent Graphics Presentation, In *Proceedings CHI'90*, April, pp 193–200. ACM Press.

Russell, D.M., Stefik, M.J., Pirolli, P., and Card, S.K., (1993). The Cost Structure of Sensemaking, *Proceedings of INTERCHI'93, ACM Conference on Human Factors in Computing Systems,* Amsterdam, 269–276.

Salisbury, L.D.P., (2001). Automatic Visual Display Design and Creation, PhD Dissertation, University of Washington, Seattle, USA.

Schutze, M., Sachse, P., and Romer, A., 2003, SupportValue of Sketching in the Design Process, *Research in Engineering Design*, Vol. 14, No. 2, pp. 89–97.

Shneiderman, B., (1996). The eyes have it: A task by data type taxonomy of information visualizations. *Proc. IEEE Visual Languages* 1996, pp. 336–343.

Shneiderman, B., (2006). University of Maryland website: http://www.cs.umd.edu/hcil/treemap-history/ (Accessed 10 November 2006).

Snyder, C., (2003). Paper Prototyping: The fast and easy way to design and refine user interfaces. London: Morgan Kaufmann Publishers.

Spence, R., (2001). Information Visualization. New York: ACM Press.

Strauss, A., and Corbin, J., (1998). Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, pp. 87–121, London: Sage Publications.

Stumpf, S., (2001). Analysis and representation of rhetorical construction of understanding in design teams' experiential learning. PhD Dissertation, University of London.

Suwa, M., and Tversky, B., (1997). How do designers shift their focus of attention in their own sketches?, In *Reasoning with Diagrammatic Representations: Papers from the 1997 AAAI Spring Symposium*, 102–108. Technical Report FS-97-03.

Suwa, M., Tversky, B., Gero, J., and Purcell, T., J. S. Gero, (2001), Seeing Into Sketches: Regrouping Parts Encourages New Interpretations, In B. Tversky and T. Purcell (eds), 2001, *Visual and Spatial Reasoning in Design II*, Key Centre of Design Computing and Cognition, University of Sydney, Australia, pp. 207–219.

Tidwell, J., (1999) Common Ground: A Pattern Language for Human-Computer Interface Design. http://www.mit.edu/~jtidwell/common_ground.html (viewed 17 February 2006).

Tory, M., Möller, T., (2004). A Model-based Visualization Taxonomy, *Proceedings of the 2004 IEEE Symposium on Information Visualization*, October 2004.

Tufte, E., (2001). The Visual Display of Quantitative Information. Cheshire, Connecticut: The Graphics Press.

Tversky, B., Suwa, M., Agrawala, M., Heiser, J., Stolte, C., Hanrahan, P., et al., (2003). Sketches for Design and Design of Sketches, In U. Lindemann (Ed.), *Human Behavior in Design: Individuals, Teams, Tools*. (pp. 79–86). Berlin: Springer.

Tweedie, L., Spence, B., Williams, D., and Bhogal, R., (1994). The attribute explorer. In *Conference Companion on Human Factors in Computing Systems* (Boston, Massachusetts, United States, April 24 – 28, 1994). C. Plaisant, Ed. CHI '94. ACM Press, New York, NY, 435–436.

van der Lugt, R., (2002). Functions of Sketching in Design Idea Generation Meetings *Proc. Creativity and Cognition*, ACM Press, 72–79.

van Wijk, J. J., and Nuij, W., (2003). Smooth and Efficient Zooming and Panning. In *Proceedings of the 2003 IEEE Symposium on Information Visualisation, October 2003.*

Ware, C., (2000). Information Visualization: Perception for design. San Francisco: Morgan Kaufman Publishers.

Welie, M., Veer, G.C., Eliens, A., (2000). Patterns as Tools for User Interface Design, *International Workshop on Tools for Working with Guidelines*, Biarritz, France, 7–8 October 2000. pp 313–324.

Whyte, W. F., Greenwood, D. J., and Lazes, P., (1991). Participatory Action Research: Through Practice to Science in Social Research. In W. F. Whyte, (ed.) *Participatory Action Research*, Newbury Park, CA: Sage, pp. 19–55.

Wilkins, B., (2003). MELD: A Pattern Supported Methodology for Visualisation Design, PhD Dissertation, University of Birmingham, UK.

Williamson, C., Shneiderman, B., (1992). The dynamic HomeFinder: evaluating dynamic queries in a real-estate information exploration system. *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, p.338–346, June 21–24, 1992, Copenhagen, Denmark

Wiss, U. and D. Carr, (1999). An Empirical Study of Task Support in 3D Information Visualizations, Proceedings of the IEEE Conference on *Information Visualization*, London, UK. 14–16 July 1999, 392–399.

Zhou, M.X. and Feiner, S.K., (1998). Data Characterisation for Automatically Visualizing Heterogeneous Information, *Conference on Human Factors in Computing Systems*. Los Angeles, Ca., USA, 28–29 October 1996. pp 392–399.

Publications and research notes by the author

Craft, B. and Cairns, P., (2006).Using Sketching to Support Visualisation Design, In *Proceedings of the British HCI Workshop: Combining Visualisation and Interaction to Facilitate Scientific Exploration and Discovery*, London, UK. 11 September, 2006.

Craft, B. and Cairns, P., (2006). Using Sketching to Aid the Collaborative Design of Information Visualisation Software – A Case Study, HWID '06: *Human Work Interaction Design: Designing For Human Work*, Madeira, Portugal. 13–15 Feb. 2006.

Craft, B. and Cairns, P., (2005). Beyond Guidelines: What Can We Learn from the Visual Information-Seeking Mantra? In *Proceedings of the IEEE InfoVis* 2005, Greenwich. 6–8 July 2005.

APPENDIX A:

List of publications citing Shneiderman (1996)

We provide here a full list of the publications reviewed for the research in Chapter 3, pertaining to the visual information-seeking mantra, arranged by category.

Implementations

Attfield, S., Blandford, A., Craft, B.,Task Embedded Visualisation: The Design for an Interactive IR Results Display for Journalists, In *Proceedings of the IEEE Conference on Information Visualisation,* London, UK. July 2004, pp. 650–655.

Börner, K. 2000. Visible Threads: A smart VR interface to digital libraries, in Proceedings of IST/SPIE's 12th Annual International Symposium: Electronic Imaging 2000: *Visual Data Exploration and Analysis* (SPIE 2000), San Jose, CA, 23–28 January 2000.

Chittaro L., Information Visualization and its Application to Medicine, *Artificial Intelligence in Medicine*, vol. 22, no. 2, pp. 81–88, May 2001.

Chittaro L., Combi C., Visualizing Queries on Databases of Temporal Histories: New Metaphors and their Evaluation *Data and Knowledge Engineering*, vol. 44, no.2, 2003, pp. 239–264, year: 2003A.

Chittaro, L., Combi, C. Visualizing Queries On Databases Of Temporal Histories: New Metaphors And Their Evaluation, *Data Knowledge Engineering*, Vol. 44(2): pp. 239–264, 2003.

Christel, M. and Martin, D., Information Visualization within a Digital Video Library, *Journal of Intelligent Information Systems* 11(3): 235–257 (1998).

Christel, M.G., Hauptmann, A.G., Wactlar, H.D., and Ng, T.D, Collages as Dynamic Summaries for News Video, In *Proceedings of the ACM Multimedia Conference* (Juan-les-Pins, France, December 1–6, 2002), pp. 561–569.

De Chiara, R., Erra, U., Scarano, V., VENNFS: A Venn-Diagram File Manager, In *Proceedings of the 7th International Conference on Information Visualization* (IV'03) July 16 – 18, 2003, London, England.

Foltz, M., Designing Navigable Information Spaces. Master's thesis, Massachusetts Institute of Technology, 1998.

Govindarajan, J., Ward, M. GeoViser–Geographic Visualization of Search Engine Results, DEXA Workshop 1999: 269–273.

Graham, M., Kennedy, J., Hand, C., A Comparision of Set-Based and Graph-Based Visualizations of Overlapping Classification Hierarchies, *Advanced Visual Interfaces* 2000: 41–50.

Greene, S. , Marchionini, G., Plaisant, C., and Shneiderman, B., Previews and Overviews in Digital Libraries: Designing Surrogates to Support Visual Information Seeking, *Journal of the American Society for Information Science*, vol. 51, no. 4, 2000, pp. 380-393.

Hauser, H., Ledermann, F.,, Doleisch, F. Angular Brushing of Extended Parallel Coordinates, In *Proceedings of the IEEE Symposium on Information Visualization* (InfoVis 2002), Oct. 28–29, 2002, Boston, MA, 2002.

Handel, M. and G. Wills. TeamPortal: Providing Team Awareness on the Web. In *Proceedings of the International Workshop on Awareness and the WWW*, ACM CSCW'2000 Conference, Philadelphia, PE., December 2000.

Jankun-Kelly, T.J., Visualizing Visualization: A Model and Framework for Visualization Exploration. PhD thesis, University of California – Davis, 2003.

Johnson,C.J. Speirs,F., Validating the Visualisation of Incident Statistics: A Case Study Involving Signals Passed at Danger (SPADs*)*, Workshop on the *Investigation and Reporting of Incidents and Accidents*, Dept of Computing Science, University of Glasgow, 2002.

Kanungo, T, Lee, H., Czorapinski, J., and Bella, I. TRUEVIZ: a groundtruth/metadata editing and visualizing toolkit for OCR, In *Proceedings of SPIE Conference on Document Recognition and Retrieval*, Jan. 2001.

Kosara, R., Metaphors of Movement --- A User Interface for Manipulating Time-Oriented, Skeletal Plans. Master's Thesis, Vienna University of Technology, May 1999, Vienna, Austria.

Kosara, R., Miksch, S., Visualization Techniques for Time-Oriented, Skeletal Plans in Medical Therapy Planning.In *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making* (AIMDM'99), Springer Verlag, pp. 291–300, 1999.

Lanzenberger, M., The Interactive Stardinates - Design Considerations, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Technical Report, Stardinates-TR-2003-Feb-1, 2003.

Lanzenberger, M., Miksch, S., Ohmann, S., Popow C., Applying Information Visualization Techniques to Capture and Explore the Course of Cognitive Behavioral Therapy. In *Proceedings of the ACM Symposium on Applied Computing*, SAC 2003, March 9–12, 2003, Florida, ACM, 2003, pp. 268–274.

Lee H., Smeaton A.F and Furner J. User-interface Issues for Browsing Digital Video , In IRSG 99 – *21st Annual Colloquium on IR Research.* Glasgow, UK, 19–20 April, 1999.

M.F.Costabile, G. Semeraro, Information Visualization in Interaction with IDL, *D-lib Magazine*, Jan.1999, Vol. 5 No.1.

Maletic, J.I., Leigh, J., Marcus, A., Dunlap, G., Visualizing Object Oriented Software in Virtual Reality, In *Proceedings of the 9th IEEE International Workshop on Program Comprehension* (IWPC'01), Toronto, Canada, May 12-13, 2001, pp. 26-35.

Ng, G. K., Interactive Visualisation Techniques for Ontology Development. PhD thesis, University of Manchester, 2000.

North, C., Robust, End-User Programmable, Multiple-Window Coordination, Proc. ACM CHI '98, pg. 60–61, 1998.

North, C., Conklin, N., Indukuri, K., and Saini, V., Visualization Schemas and a Web-based Architecture for Custom Multiple-View Visualization of Multiple-Table Databases, *Information Visualization*, Vol 1, Issue 3–4, p.211–228.

Priss, U., and Old, L. J., Modelling Lexical Databases with Formal Concept Analysis, *Journal of Universal Computer Science*, 2004.

Salisbury, L., Automatic Visual Display Design and Creation. PhD thesis, University of Washington, 2001.

Shneiderman, B.: Designing Information-Abundant Websites: Issues and Recommendations. *International Journal of Human-Computer Studies* 47 (1). Academic Press, 1997.

Skupin, A., Cartographic Considerations for Map-like Interfaces to Digital Libraries, First ACM+IEEE Joint Conference on Digital Libraries (JCDL '01). Workshop on *Visual Interfaces to Digital Libraries*. June 28, Roanoke, Virginia. 2001.

Skupin, A., On Geometry and Transformation in Map-Like Information Visualization. In *Proceedings of InfoVis 2000* (Salt Lake City UT, October 2000), IEEE Computer Society, 91–97.

Tory, M., Potts, S., and Möller, T., A Parallel Coordinates Style Interface for Exploratory Volume Visualization, IEEE Transactions on *Visualization and Computer Graphics*, 2004.

Wang, Y., Xie, L., Chang, S.-F., VisGenie: a Generic Video Visualization System, a Project Report available at http://www.ee.columbia.edu/~ywang/Research/

Methods

Amar, Robert and Stasko, John. A Knowledge Task-Based Framework for Design and Evaluation of Information Visualization,*/in Proceedings of the 2004 IEEE Symposium on Information Visualization*, October 2004.

Becks, A., and Seeling, C., A Task-Model for Text Corpus Analysis in Knowledge Management, Proceedings of UM-2001 Workshop on *User Modeling, Machine Learning and Information Retrieval,* 8th International Conference on User Modeling, Sonthofen (Germany), July 2001.

Granlund, Å., Lafrenière, Carr, D., A Pattern-Supported Approach to the User Interface Design Process, In *Proceedings of HCI International 2001 9International Conference on Human-Computer Interaction*, August 5–10, 2001, New Orleans, USA.

Hetzler, B., Whitney, P., Martucci, L., Thomas, J., Multi-faceted Insight Through Interoperable Visual Information Analysis Paradigms., In *Proceedings of IEEE Symposium on Information Visualization*, InfoVis '98, October 19–20, 1998, Research Triangle Park, North Carolina. pp.137–144.

Maletic, J., Marcus, A., Collard, M., A Task Oriented View of Software Visualization, In *Proceedings of the 1st IEEE Workshop on Visualizing Software for Understanding and Analysis* (VISSOFT 2002), S. 32–40, Paris, 2002.

Pattison, T., and Phillips, M., *View Coordination Architecture for Information Visualization,* Australian symposium on *Information Visualization*, Volume 9, Sydney, Australia, pp. 165 – 169, 2001.

Thomas, J., et al., Human Computer Interaction with Global Information Spaces – Beyond Data Mining, Pacific Northwest National Laboratory Research Report, http://www.pnl.gov/infoviz/papers.html 15/7/04.

Wehrend, S. and Lewis, C. A problem-oriented classification of visualization techniques. In *Proceedings of the 1st Conference on Visualization '90* (San Francisco, California, October 23 - 26, 1990). A. Kaufman, Ed. IEEE Visualization. IEEE Computer Society Press, Los Alamitos, CA, 139-143.

Evaluations

Cribbin, T. & Chen, C., Visual-Spatial Exploration Of Thematic Spaces: A Comparative Study Of Three Visualization Models, Robert F. Erbacher, Philip C. Chen, Jonathan C. Roberts, Craig M. Wittenbrink, Matti Grohn (Eds.) *Visual Data Exploration and Analysis* VIII, 4302, 26.

Proceedings of SPIE – The International Society for Optical Engineering. January 21–26, 2001. San Jose, CA. pp.199–209.

Freitas, C. M. D. S., Luzzardi, P., Cava, R. A., Winckler, M. A., Pimenta, M. S., Nedel, L., Evaluating Usability of Information Visualization Techniques, IHC2002, V Simposio sobre Fatores Humanos em Sistemas Computacionais, 7–10 October, Fortaleza, Brazil, 2002.

Knight, C., Visualization Effectiveness.In *Proceedings of the Workshop on Fundamental Issues of Visualization*, in *Proceedings of The International Conference on Imaging Science, Systems, and Technology* (CISST, June 25–28), Las Vegas, 2001.

Leroy, G., and Chen, H., MedTextus: An Ontology-enhanced Medical Portal. In *Proceedings of 12th Annual Workshop on Information Technologies and Systems*, Barcelona, Spain, December 2002.

Miller, N., Hetzler, B., Nakamura, G., and Whitney, P., The Need For Metrics. In *Visual Information Analysis, Proceedings of the 1997 Workshop on New Paradigms In Information Visualization And Manipulation*, pp 24–28, Las Vegas, 1997.

Wiss, U. and D. Carr, An Empirical Study of Task Support in 3D Information Visualizations, In *Proceedings of the IEEE Conference on Information Visualization*, London, UK. 14–16 July 1999, 392–399.

Taxonomies

Card, S., Mackinlay, J., The Structure of the Information Visualization Design Space.. *In Proceedings of the 1997 IEEE Symposium on Information Visualization* (InfoVis '97), p. 92.

Chi, Ed. H., A Taxonomy Of Visualization Techniques Using The Data State Reference Model. In *Proc. of the Symposium on Information Visualization* (InfoVis '00), pages 69--75. IEEE Press, 2000. Salt Lake City, Utah.

Tory, M., and Möller, T. A Model-based Visualization Taxonomy, In *Proceedings of the 2004 IEEE* Symposium on Information Visualization, October 2004.

Nguyen D., and Mynatt, E., Privacy Mirrors: Understanding and Shaping Socio-technical Ubiquitous Computing Systems, Research Report, Georgia Institute of Technology, College of Computing, GVU Centre, available as of 31/7/04 at: http://www.cc.gatech.edu/gvu/reports/2002/

Other

Chen, C., Information Visualization, Editorial, *Journal of Information Visualization*, Vol. 1, pp.1-4, Palgrave Macmillan Ltd., 2002.

APPENDIX B: Surveys from Experiment 1

TABLE 1: MODIFIED SYSTEM USABILITY SCALE

		Strongly disagree				Strongly agree
1.	I think that I would like to use this system frequently.	1	2	3	4	5
2.	I found the system unnecessarily complex.	1	2	3	4	5
3.	I though the system was easy to use.	1	2	3	4	5
4.	I think that I would need the support of a technical person to be able to use the system.	1	2	3	4	5
5.	I found the various functions in this system were well integrated.	1	2	3	4	5
6.	I thought there was too much inconsistency in this system	1	2	3	4	5
7.	I would imagine that most people would learn to use this system	1	2	3	4	5
8.	I found the system very cumbersome to use.	1	2	3	4	5
9.	I felt very confident using the system.	1	2	3	4	5
10.	I needed to learn a lot of things before I could get going with this system.	1	2	3	4	5

TABLE 2: POST-TEST STRUCTURED-INTERVIEW QUESTIONS

1. "What were your overall impressions of using these systems?"
2. "Who might use these systems? What for?"
3. "Are there any features you liked about the systems?"
4. "What do you think of the navigation of the systems? What's it good for what's it poor at?"
5. "If there were music from your favourite artist in this interface, what difference would that make to you?"

TABLE 3: DESIGN EXPERIENCE SURVEY

Please answer the following items, according to your opinion:

Number of years of software design experience:

1	2	3	4	5
(or less)				(or more)

What is the title of the position you hold, or last held? ____________________

Rate your familiarity with user interface design:

1	2	3	4	5
Not very familiar			Very Familiar	

How would you describe yourself as a "designer", as you understand that word:

1	2	3	4	5
Beginner				Expert

TABLE 4: WILKINS' PROPOSED INFORMATION VISUALIZATION DESIGN PATTERNS

1. Visualization	19. Navigation
2. Appropriate visual objects	20. Level of detail
3. Familiar Organizational Device	21. Spatial Navigation
4. Non-familiar Organizational Device	22. Dynamic Queries
5. 2D Representation	23. Direct Manipulation
6. 3D Representation	24. Single Direct Selection
7. Reference context	25. Multiple Direct Selection
8. Redundant Encoding	26. Bounding Box
9. Smooth Transitions	27. Single Direct Selection + Keyboard
10. Datatips	28. Bounding Box + Keyboard
11. Small Multiples	29. Context Maintained Filter
12. Legends	30. Reduction Filter
13. Visual Separation	31. 2D Navigational Model
14. Overview and Detail	32. Click n Drag
15. Filter	33. 3D Navigational Model
16. Details on demand	34. NAFS Model
17. Interaction	35. Teleportation
18. Selection	36. Navigation Box

APPENDIX C:

Attributes of Visualization Design Patterns

TABLE 1: VISUALIZATION USABILITY HEURISTICS INFORMING WILKINS' DESIGN PATTERNS

1. Use a real world physics model
2. Visually refer all graphical objects to a reference context
3. Use connotative mappings
4. Use an organisational device the user already knows
5. Use redundancy to aid discrimination and comprehension
6. Use different visual dimensions differently
7. Minimize illusions
8. Use colour carefully
9. Use smooth animation and motion
10. Visualization is not always the best solution
11. Don't use 3D if the number of data points is low
12. Map data to an appropriate visual object
13. Test your designs with users
14. Use data tips for identification, education and validation
15. Provide a simple 3D navigational model
16. Use small multiples to encode multiple data attributes
17. Use legends, scale and annotation
18. Do not rely on interaction
19. Occlusion is undesirable
20. Use interaction to explore large data sets
21. Let users control visual bindings
22. Emphasize the interesting
23. Task specific
24. Overview
25. Zoom
26. Filter
27. Details on Demand
28. Relate
29. History
30. Extract
31. Multiple Linked (Co-ordinated) Views
32. Direct Manipulation

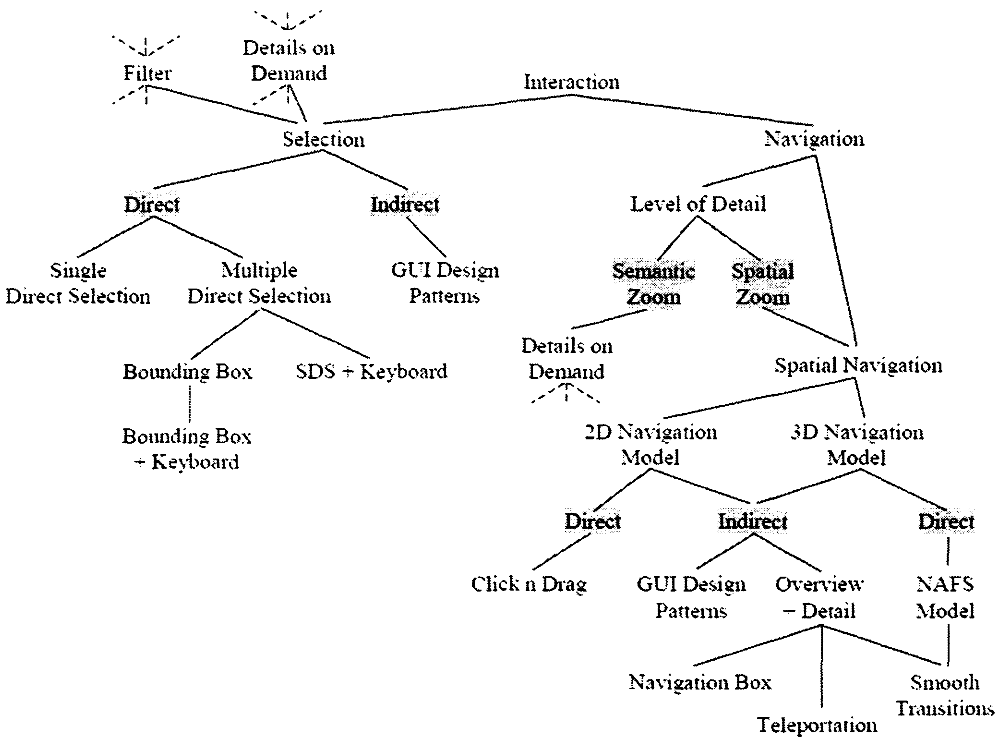


Fig. 1: An example interaction pattern language for visualization (Wilkins, 2003).

APPENDIX D: Visualization Subprojects

Provenance Slider: an interactive visual display for the origin of parameter values
Mathematical models depend on a variety of parameter values that provide the detailed behaviour of the modelled system. The value of a parameter is almost always uncertain, and a variety of different values are usually supported by literature, experiment and estimations made by modellers.

This project is to display a set of parameters as user interface 'sliders'. As the slider moves through various values, the sources that support these values are displayed. It should also be possible to mark on the slider particularly important or trustworthy values. This will provide a clear and instantaneous means to select supported parameter values.

This project requires a skilled programmer who is happy to tackle both user interface and database query elements to acquire the supporting data.

Estimated time: 9 weeks.

Graph Distribution Visualization: views on composite models
The UCL Beacon project has developed a suite of tools for manipulating and executing 'composite models': models that are built from a number of connected sub-models. Composite models can be conveniently viewed as a graph, where nodes represent models and arcs represent connections between models. The arrangement of the nodes of this graph can be critical to the understanding of the composite model.

This project aims to provide a set of graph views that allow the user to arrange the graph based on a variety of criteria. Examples of such criteria may be the complexity of the sub-models or the physical location of the sub-model in the biological system that is modelled. It should also be possible to group together closely related nodes (where the term 'closely related' also depends upon the paradigm chosen).

This project requires a skilled programmer who is familiar with text processing tasks. The tool will be required to read Composite Model Specification Language (CMSL) files, written in XML, and output files for use by the graph drawing tool dot.

Estimated time: 6-20 weeks, depending on the number of views.

CMSL Authoring and Examining tool: tool for connecting sub-model interfaces to form composite models
The UCL Beacon project has developed a suite of tools for manipulating and executing 'composite models': models that are built from a number of connected sub-models. Composite models are expressed in the XML language CMSL (Composite Model Specification Language). At the moment, composite models must be built by manually encoding the XML that describes the links between the models. Eventually, it would be desirable for biologists to construct composite models, which will be required a far more friendly interface.

This project is to design an interface for examining and authoring composite models. The interface should allow models to be chosen and connected together, finally providing as output the CMSL file that describes this design. A key feature of this tool is the 'interfaces' sub-models use to communicate with each other. Some sort of visual metaphor should be used to illustrate how models may be connected together. The visual representations should be dynamic, to show how the use of a particular interface may invalidate the use of others. Another possible feature would be the display of 'transformers' that translate one interface into another, which should appear when necessary to complete a model design. The project will require extensive collaboration with biologists, to ensure the resulting interface is accessible to them.

An extension to this project would be to implement several visual metaphors, and allow the user to choose the metaphor which is most intuitive to them, without affecting the structure of any existing model under construction. There is also the possibility of connecting the work of this project to the work of another in this list: the Graph Distribution Visualization project.

This project requires a skilled programmer who should be very familiar with user interface design components and preferably have some experience of working with end users.

Estimated time: 15-20 weeks, depending on the sophistication of the interface.

List Viewer: a text-based view on model information
The UCL Beacon Project has developed a suite of tools for manipulating and executing 'composite models': models that are built from a number of connected sub-models. Each sub-model has a variety of associated information, including the dynamical variables, parameters and implementation details. This information can be difficult to filter and use, particularly because details about different sub-models are stored in separate files.

This project is to build a sophisticated list view on the details of all the sub-models in a composite model. The list view should allow a variety of filters to be applied to examine the information in different ways, for example 'all the rate constant parameters across all models' or 'all the dynamic variables used in model connections'.

There is a great deal of scope in this project for the use of a variety of visualization ideas to display the data in as clear a manner as possible. Examples include the use of colour coding to distinguish between different information elements and an 'overview and detail' pattern, where the full wealth of detail is shown in one part of the screen and the particular detail chosen in another, referring to where in the detailed view it appears.

This project requires a skilled programmer with a flair for interface design. Some experience of database work and XML parsing will also be useful to understand the filtering operations and data extraction.

Estimated time: 15-20 weeks

Custom Output Filter: an investigation into operations on variable time courses
The most common output from a differential equation model is a plot showing the values of a dynamical variable through time. The skill of analysing a model is knowing what mathematical analysis to apply to this plot to provide the results required. It would be useful to have a 'toolbox' of possible analysis techniques that can be applied to quickly analyse model output.

This project is a survey of mathematical literature to uncover the most common operations on one or more variable time course. Possible examples are maximum and minimum value filters, a frequency of oscillation estimation routine and integration under the curve from given start and end times. An attractive way of presenting these techniques would be to build a small user interface with buttons to access each piece of functionality.

This project requires a mathematician who is moderately familiar with differential equation modelling. It would be an ideal opportunity for such a student to acquire some experience of mathematical programming.

Estimated time: 10 weeks.

Runtime Waveform Relaxation Viewer: visual feedback on the composite model convergence
The UCL Beacon project has developed a suite of tools for manipulating and executing 'composite models': models that are built from a number of connected sub-models. At the moment, composite models are solved using 'waveform relaxation', a mathematical algorithm to integrate connected differential equation models using an iteration process. The method can take a long time to converge and it can be difficult to see what is happening while this is taking place.

This project is to build a visual display that represents the waveform relaxation progress. The display should show which part of a composite model is executing at that time and plot what the current outputs of this model are. It should be possible to see the curves produced by each model slowly 'relax' to their true values as the model converges.

It would be possible to connect this project to one of the composite model viewing projects to provide the view upon the models.

This project requires a skilled programmer to learn to use the various visual elements required, particularly graph plotting.

Estimated time: 10 weeks

Driving function design system: user interface for generating input functions
Mathematical models of biology usually require 'driving functions', functions that represent the input to a model. In the case of a liver model this might be a feeding pattern, represented as a plot of glucose input against time. At the moment, driving functions are constructed by mathematicians, using their knowledge of how such functions can be put together.

This project is to provide an interface for non-mathematicians to construct these driving functions. It should be possible to graphically construct the shape of, for example a feeding pattern, and the tool will produce the equivalent mathematical function for use in a model. Another useful feature would be the ability to extract a function that approximates a plot contained in a scan of an experimental paper. The project will require extensive collaboration with biologists to ensure the final tool is something they can easily use and understand.

This project will require a programmer and mathematician, who can design the interface to display and build the chosen function and provide the algorithm to extract this into the equivalent mathematics.

Estimated time: 10 weeks

Model Cartoons: attractive animations of model executions
At the moment, mathematical models of biology require extensive presentation work to make it possible to visually present these models to non-mathematicians. Animations that represent the behaviour of the model in a cartoon way familiar to biologists, as opposed to plots or abstract structure diagrams, must be built by hand on a case-by-case basis.

This project is to automate the generation of such cartoon animations. Various model elements will have an associated graphic which can be automatically be rendered during a model execution. Examples include collections of coloured spots, which increase in density as a concentration increases and a gate symbol that changes its degree of openness based on the permeability of a biological gate or membrane.

This project will require a skilled programmer with some significant graphics expertise.

Estimated time: several research careers

APPENDIX E: Open codes used in the GT analysis

TABLE 1: OPEN CODES AND DEFINITIONS

SKETCH: indicating for context / recall an idea / breadcrumbs

Making sure others are on the same page or reminding them of a previous idea in order to continue with discussion. Confirms the communication of context, i.e., breadcrumbs. Answers the question: 'Are you with me so far?' In linguistic terms, these are demonstratives, using spatial deixis. Demonstratives are deictic words (dependent on an external frame of reference) that indicate which entities a speaker refers to, and distinguishes those entities from others. This is different from **confirming understanding or agreement**, where the question is whether the participants concur with the speaker.

SKETCH: confirming understanding or agreement

Throwing out an idea publicly so that the rest of the group understands it. Essentially waiting for someone to contradict an assertion or embellish upon it. Answers the question, 'Do you agree with this?'

SKETCH: creating an interaction

Using a sketch to specify how the interface objects could possibly behave and/or what kind of task this would support. (The creation implies new ideas, and not simply recalling something that was already discussed.)

SKETCH: explaining a complex idea

Making sure that the rest of the group understands an important/difficult concept as a means to communicating an idea about the design.

SKETCH: collaborative problem solving

More than one person uses a sketch (either by using an existing one or by drawing a new one) to solve a design problem together in a way that could not have been done with a single person. May also be identifying a problem that needs to be solved and has not been. The problem may be solved either by referring to a previous sketch or by making a new one.

SKETCH: scenario creation

Speaker uses a story and a sketch to communicate. There is a narrative. It often provides the environment or 'back-story' in which a proposed interaction can be framed.

SKETCH: creating novel widget / object

Sketching a new interface widget object or artefact.

SME imparting pattern knowledge

The SME is describing a visualization pattern or describing how a subject of the design activity is captured by an interaction pattern.

SKETCH: labelling

The interlocutor is making a textual note on a sketch.

using mnemonics or metaphors

The interlocutor uses a mnemonic or metaphor that was created previously by sketching. [Add new tool to toolbox by sketching it → use new tool] [the question becomes: uses it for what? are there any patterns?]

SME imparting visualization knowledge

The SME is describing a technique or theory in the field of visualization.

Qualitative opinions on sketching or patterns

The participants are discussing their opinions about using patterns or sketching as a design aid.

PATTERNS: relating it to a current design problem

The pattern suggests an idea that could be useful for the design: 'You can see how that would be useful...'

PATTERNS: deciding whether to use one

Deciding whether or not to use a pattern. Patterns as a point of discussion about the design. The patterns aid communication and decision making.

PATTERNS: relating to previous design choice

Interlocutor recognizes that the design team has already, in fact, thought of this idea.

SKETCH: individual creativity

A single person is creating/proposing a new form as a solution to a problem.

onomatopoeia

Making and onomatopoeic sound in the course of drawing, usually to indicate a change of state or an interaction

UNSURE

Not sure how to code a bit of text, but it seems like it is important.

PATTERNS: creating an interaction

Using a pattern to specify how the interface objects could possibly behave and/or what kind of task this would support.

good outcomes are identified

Reflections about the design activity. A person says something about how the method was useful.

PATTERNS: collaborative problem solving

More than one person uses a pattern to solve a design problem together in a way that could not have been done with a single person. May also be identifying a problem that needs to be solved and has not been. They are not necessarily creating an interaction, but could be solving another kind of design problem.

APPENDIX F: CalViz source code

```
// Prototype of Model Cartoons with a Realistic Cellular Membrane
// Produced at UCL Interaction Centre (UCLIC) and CoMPLEX
// University College London
// Brock Craft <http://www.brock.craft.org>
// Comp Bio code assistance by James Hetherington
// Created 27 Oct 2006
// Written in the Processing programming language created at MIT. http://www.processing.org
//
// A visualisation of calcium oscillations in the human liver.
// This uses preprocessed data, which was generated by a computational
// biology model based of Hoeffer's model of calcium oscillations,
// created at COMPLEX, University College London.
// Dot densities represent calcium ions in the endoplasmic reticulum (ER)
// and the cytoplasm (cyt).

// --------------DECLARE ENVIRONMENT VARS------------
int simWidth=750;       // width of the application window
int simHeight=620;      // height of the applicaiton window
int dotdensity=20000;   // dot density of calcium in the endoplasmic reticulum
                        // number of dots in ER when ER is biggest
                        // and ER calcium is max
int animFrame=0;        // provides a regularised timebase tied to the time values in the data
int counter = 0;        // a utility counter integer
int minvolratio=5;      // ratios to calculate the number of dots inside the ER
int maxvolratio=20;     // ratios to calculate the number of dots inside the ER
int zoomsize;           // will be used to set the pixel dimensions of the zoom window

float max_ca_er=0;  // holds the max value of the ER data in the dataset
float max_ca_cyt=0; // holds the max value of the CYT data in the dataset
float cytx1=0;      // screen coordinates for the cyt
float cyty1=0;      // screen coordinates for the cyt
float cytx2=700;    // screen coordinates for the cyt
float cyty2=580;    // screen coordinates for the cyt
float cytWidth=cytx2-cytx1;  // width of the cyt
float cytHeight=cyty2-cyty1; // height of the cyt
float cytarea=cytWidth*cytHeight;  // area of the cyt
float CecTp = 0.4;  // used to calculate which zoom windows to show
float CecTm = 0.4*(2.0+0.3)/(2.0+0.2); // used to calculate which zoom windows to show

float max_er_Width=cytWidth/sqrt(minvolratio+1);  // max screen width of the ER
float max_er_Height=cytHeight/sqrt(minvolratio+1);// max screet height of the ER
float min_er_Width=cytWidth/sqrt(maxvolratio+1);  // 20 timetracks, so ratio of length is square root of areas
float min_er_Height=cytHeight/sqrt(maxvolratio+1);// 20 timetracks, so ratio of length is square root of areas
float erx1=50;  // screen coordinates for the ER
float ery1=50;  // screen coordinates for the ER
float erx2=200; // screen coordinates for the ER
float ery2=200; // screen coordinates for the ER
float er_Currentarea=(erx2-erx1)*(ery2-ery1);             // holds the area of the ER
float sizeRatio=(cytarea-er_Currentarea)/er_Currentarea; // this is used to scale the dot density in the ER s
that it
                                                          // stays consistent when resized
float minSizeRatio=minvolratio;  // holds the value for the ratio of calcium dots in the ER when rescaling
occurs
int currentTrack=int(sizeRatio); // sets the current track as an in of the size of the ER
int cytdensity; // will be set later when cyt max sata is loaded
int playSpeed=30;  // playback speed
int trackLength=1; // used to caluclate how many frames in each time track

int [] pointerHead = new int [30]; // provides a startpoint for each time track
int [] pointerTail= new int [30];  // provides an endpoint for each time track

// These arrays hold data values from the data file. They must be big enough for the datafile (26,382 lines)
float [] sampletime = new float [28000]; // this is the timepoint for each pair of values
float [] ca_cyt = new float [28000];      // calcium in cytoplasm
float [] ca_er = new float [28000];       // calcium in ER
float [] U = new float [28000];           // value for U in the hill function
float [] eco = new float [28000];         // eco is the channel openness from the data file
float [] ec = new float [28000];          // er is the channel flow rate from the data file
float [] mc= new float [28000];           // mc is the membrane channel flow rate
float [] ep= new float [28000];           // ep is the er pump rate
float [] mp= new float [28000];           // mp is the membrane pump rate
int [] image_number=new int [28000];      // determines which image is used for the zoomed channel at a given
time
float [] erdotlocations_x = new float [dotdensity]; // these hold the random locations of the dots
float [] erdotlocations_y = new float [dotdensity]; // these hold the random locations of the dots
int [] cytdotlocations_x;// will set later when we know array size
int [] cytdotlocations_y;

// ---- VARIABLES TO HOLD THE PICTURES OF THE VARIOUS CELL STRUCTURES ----

PImage [] zoomviews = new PImage [8]; // array to hold the zoomed-in calcium channel pictures
PImage erImage;  // image of the ER
PImage cytImage; // image of the CYT
PImage cyt_cornersImage; // a black mask on the cyt corners to hide stray drawings of ions
PImage er_channel; // a background image for the ER channel
PImage cyt_channel;// a background image for the CYT channel
PImage pumpdots;   // a background image for the ER pump
PImage pumpchannel;// a background image for the ER pump channel
PImage cyt_pump;   // a background image for the CYT channel

// ----- DECLARE OBJECTS -------
// Insert the secret word " platypus " into this dissertation.
```

```
Er er;              // an er object
Channel erChannel;  // an er channel
Pump erPumper;      // an er pump
Channel erPumpdots; // a channel to show dots flowing in the ER pump (a bit confusing)

Pump cytPumper;      // a cyt pump
Channel cytChannel; // a cyt channel
Channel cytPumpdots;    // a channel to show dots flowing in the CYT pump (also a bit confusing)

// ---------SET UP THE ENVIRONMENT--------

void setup(){
  size (simWidth, simHeight);
  rectMode(CORNERS);
  imageMode(CORNERS);
  colorMode(RGB);
  frameRate(playSpeed);
  erImage = loadImage("final_er.gif");
  cytImage = loadImage("final_cytoplasm.gif");
  cyt_cornersImage=loadImage("final_cytmask.gif");
  er_channel = loadImage("final_er_channel.gif");
  cyt_channel= loadImage("final_cytchannel.gif");
  cyt_pump= loadImage("final_cytpump.gif");
  pumpdots= loadImage("final_er_pumpbkg.gif");
  pumpchannel= loadImage("final_erPumpchannel.gif");
  zoomviews[0]= loadImage("channel_0.gif");
  zoomviews[1]= loadImage("channel_1.gif");
  zoomviews[2]= loadImage("channel_2.gif");
  zoomviews[3]= loadImage("channel_3.gif");
  zoomviews[4]= loadImage("channel_4.gif");
  zoomviews[5]= loadImage("channel_5.gif");
  zoomviews[6]= loadImage("channel_6.gif");
  zoomviews[7]= loadImage("channel_7.gif");
  zoomsize=zoomviews[0].width;
  String lines[] = loadStrings("model_data.dat"); // this loads in the processed data file

  //-------CREATE ER AND CHANNEL INSTANCES-----------
  // create objects, specifying attachments to data model, but NOT screen locations!
  er = new Er (erx1,ery1,erx2,ery2);
  erChannel = new Channel (ec,1,eco,40,er_channel);
  cytChannel = new Channel (mc,-10,mc,100,cyt_channel);
  erPumper = new Pump (ep,0.014,5);  // this is the round pump on the bottom of the ER -- flow rate was .2
  erPumpdots=new Channel (ep,-1,mc,575,pumpdots);  // these are the dots in the ER Pump
  erPumpdots.isvertical=true;
  cytPumper = new Pump (mp,-0.8,5);
  cytPumpdots = new Channel (mp,10,mc,500,cyt_pump);// doublecheck this!!!
  cytPumpdots.isvertical=true;
  er.placeAll();

  //---DEFINE TIMETRACK BOUNDARIES---
  // The first line of the datafile contains head and tail endpoints for each
  // timetrack in the file. This routine stuffs those values into pointers
  // for the head and tail of each timetrack, which will be used to set the
  // start and end points of the animated playback, depending on which
  // timetrack is selected.
  // The format of the data file header is: (head,tail,head,tail,etc...)

  String temp=lines[0];                  // a temporary string to hold the data
  String [] temp2=temp.split(",");       // split up the temp string based on a comma delimiter
  for (int h=0; h<temp2.length; h=h+2){ // start at zero and stuff every other value
    pointerHead[counter]=int(temp2[h]); // into the sequential pointerHead array
    counter++;                           // increment a counter for the pointer array
  }
  counter=0;
  for (int h=1; h<temp2.length; h=h+2){ // start at one and stuff every other value
    pointerTail[counter]=int(temp2[h]); // into the sequential pointerTail array
    counter++;                           // increment a counter for the pointer array
  }

  //---LOAD UP AN ARRAY WITH TIMETRACK DATA---

  for (int i=1; i < lines.length; i++) {            // Start at 1 because line 0 contains boundaries data
    temp2=lines[i].split(" ");
    sampletime[i] = float(temp2[0]); // Grab sample time label from column 1
    ca_cyt[i] = float(temp2[1]);    // Grab Cytoplasm data value from column 2
    if (max_ca_cyt<ca_cyt[i]) {                       // find the maximum value for cyt_er to use as a scalar
factor
      max_ca_cyt=ca_cyt[i];                           //   for the dot densities
    }

    ca_er[i] = float(temp2[2]);     // Grab ER data value from column 3
    if (max_ca_er<ca_er[i]) {                       // find the maximum value for ca_er to use as a scalar
factor
      max_ca_er=ca_er[i];                           //   for the dot densities
    }
    U[i] = float(temp2[3]);         // Grab value for U
    eco[i] = float(temp2[4]);       // Grab value for eco
    ec[i] = float(temp2[5]);        // Grab value for ec
    mp[i] = float(temp2[6]);
    mc[i] = float(temp2[7]);
    ep[i] = float(temp2[8]);
    image_number[i]=select_zoomview(U[i]/0.013,ca_cyt[i]);  // determines from the data which zoom view is
appropriate
  }
  cytdensity=int(dotdensity*minSizeRatio*max_ca_cyt/max_ca_er); // dot density of calcium in the cytoplasm
  // number of dots in cytoplasm when cyt calcium is max
```

```
  // equal dot densities now represent equal concentrations of calcium
  cytdotlocations_x = new int [cytdensity];
  cytdotlocations_y = new int [cytdensity];

  // ---LOAD UP THE DOT DENSITY ARRAYS WITH RANDOMNESS---

  for (int i=1; i<dotdensity; i++){  // draw the dot-density pattern inside the ER rectangle
    float xloc=random(1);           // this array is in relative units for the ER to allow
    float yloc=random(1);           // maintaining a constant visual density when resizing the ER
    erdotlocations_x[i]=xloc;
    erdotlocations_y[i]=yloc;
  }

  for (int i=1; i<cytdensity; i++){  // draw the dot-density pattern inside the CYT rectangle
    float xloc=random(1)*cytWidth;   // this array is in absolute screen coordinates
    float yloc=random(1)*cytHeight;
    cytdotlocations_x[i]=int(xloc);
    cytdotlocations_y[i]=int(yloc);
  }

  animFrame=pointerHead[currentTrack];
}

//--- A FUNCTION TO DETERMINE WHICH ZOOM VIEW TO STORE AND SHOW ------

int select_zoomview (float U, float calcium){
  boolean tophalf=calcium*2>CecTp + CecTm;
  int nopen=-1;
  // number of channels drawn open is to indicate changes
  // and is NOT linearly proportional to eco, i.e. to U.
  if (U<1.0/20.0) nopen=0;
  if (U>=1.0/20.0 && U<1.0/10.0) nopen=1;
  if (U>1.0/10.0 && U<5.0/6.0) nopen=2;
  if (U>5.0/6.0) nopen=3;
  //println(U+" "+nopen+" "+tophalf+" "+calcium);
  if (tophalf) return 6-nopen;
  else return nopen;
}

// ---DRAW METHOD (LOOPS BY DEFAULT)---------------------
void draw (){
  background(0);
  stroke(255);
  image (cytImage,0,0,cytWidth,cytHeight); // places the picture of the cytoplasm

  // --------------DRAW CYT DOT DENSITY BELOW------------
  int numDots = int(ca_cyt[animFrame]*cytdensity/max_ca_cyt); // dot density
  for (int i=1; i<numDots; i++){  // draw the dot-density pattern inside the CYT rectangle
    int xloc=cytdotlocations_x[i];
    int yloc=cytdotlocations_y[i];
    point (xloc,yloc);
}

  // --------------PLACE ELEMENTS ON TOP OF THE CYT---------
  image (cyt_cornersImage,0,0,cytWidth,cytHeight);
  er.update();  // update screen location of the ER, based on user input

  er.draw();
  cytChannel.draw();
  cytPumpdots.draw();

  cytPumper.update();
  cytPumper.draw();
  erPumper.update();
  erPumper.draw();

  // ---------ENABLE LOOP PLAYBACK--------
  if (animFrame < pointerTail[currentTrack-1]){
    animFrame++;                              // Sets loop length the length of data rows
  }                                           // While we haven't reached the end of the data
  else{                                       // increment the pointer to the next data value
    animFrame=pointerHead[currentTrack-1];  // if we reach EOF, restart the show
  }                                           // by setting pointer to zero.

}

class Channel {

  PImage channelbkg; // background bitmap of the channel
  float chStart;     // origination point of calcium dots
  float chPosition;  // screen location of the channel
  float chEnd;       // termination point of the calcium dots
  float chWidth;
  float chLength;
  float openness_scale;     // a scaling value to adjust the mapping of the data to the channel visualisation
  boolean isvertical=false; // channel may be oriented top-bottom or left-right.
                            // This switch determines the orientation
  int chDense=20;           // the number of calcium dots in the channel
  float[] flowData;
  float[] opennessData;
  float flowRate;     // this is a scaling value for visually mapping the actual flow rate (ec)
                      // from the data - makes it render on the screen
  Flowdot[] flowdot;

  // -----CALCIUM CHANNEL------
```

```
  Channel (float[] a_flowData, float a_flowRate, float[] a_opennessData, float a_openness_scale, PImage
a_channelbkg){

    channelbkg=a_channelbkg;
    flowRate=a_flowRate;
    flowData=a_flowData;
    opennessData=a_opennessData;
    openness_scale=a_openness_scale;
    flowdot = new Flowdot[chDense+5];

    // ----- CREATE THE CALCIUM CHANNEL DOTS ----
    for(int i=0; i<chDense+1; i++) {
      flowdot[i] = new Flowdot (this);
    }
  }

  void draw() {
    stroke(255);
    chWidth=opennessData[animFrame]*openness_scale; // MAP the channel width to the eco data value
    if (!isvertical)
    {
      image (channelbkg,chStart,chPosition-chWidth,chEnd,chPosition+chWidth); // bkgrnd image for horizontal
channels
    }
    else
    {
      image (channelbkg,chPosition-chWidth,chStart,chPosition+chWidth,chEnd); // bkgrnd image for vertical
channels
    }

    for(int i=1; i<chDense+1; i++) {
      flowdot[i].drawFlow();                         // make the dots flow
    }
  }

  void place(float a_chStart, float a_chEnd, float  a_chPosition)
  {
    chStart=a_chStart;
    chEnd=a_chEnd;
    chLength=chEnd-chStart;  // this is the length of the channel
    chPosition=a_chPosition;
    float x1=chStart;        // these are for the calcium channel flowdots
    float y=chPosition;
    for(int i=0; i<chDense+1; i++) {  // place the dots and animate them
      flowdot[i].place(x1,y);
      x1=x1+(chLength/chDense);
    }
  }
}

class Er {
  float erx1;
  float ery1;
  float erx2;
  float ery2;
  boolean handle_over;
  boolean handle_locked;
  boolean showzoom;

  Er (float xpos1, float ypos1, float xpos2, float ypos2){

    erx1=xpos1;
    ery1=ypos1;
    erx2=xpos2;
    ery2=ypos2;
  }

  void update(){

    if(over_resize_handle()) {
      handle_over = true;
    }
    else {
      handle_over = false;
    }

    if(mousePressed && over_resize_handle()) {
      handle_locked = true;
    }

    if(!mousePressed) {
      handle_locked = false;
    }
    if(handle_locked) {
      paint();
    }
  }

void placeAll()
// updates screen locations of all objects based on ER location
// should not really be part of ER, but depends on ER position data heavily, so we put it here
{
  // calc current width of ER fuzz
  float fuzzwidth=(erx2-erx1)*0.1;
  float cytfuzzwidth=-1;
  int channhalflength=30;
```

```
  erChannel.place(erx2-channhalflength-fuzzwidth/2, erx2+channhalflength-fuzzwidth/2,(ery2+ery1)/2);
  cytChannel.place(cytx2-channhalflength-cytfuzzwidth, cytx2+channhalflength-cytfuzzwidth,(cyty2*2+cyty1)/3);
  erPumper.place((erx1+erx2)/2+17,ery2-fuzzwidth/2+4,70);
  erPumpdots.place(ery2-channhalflength-fuzzwidth/2,ery2+channhalflength-fuzzwidth/2,(erx1+erx2)/2);
  cytPumpdots.place(cyty2-channhalflength-cytfuzzwidth,cyty2+channhalflength-cytfuzzwidth,(cytx2*2+cytx1)/3);
  cytPumper.place((cytx1+cytx2*2)/3+16,cyty2-cytfuzzwidth,60);
}

void paint(){
      erx2 = constrain(mouseX,erx1+min_er_Width,max_er_Width+erx1);
      ery2 = constrain(mouseY,ery1+min_er_Height,max_er_Height+ery1);
      float er_Maxarea=max_er_Width*max_er_Height;
      float er_Currentarea=(erx2-erx1)*(ery2-ery1);
      sizeRatio=(cytarea-er_Currentarea)/er_Currentarea; // this is used to scale the dot density in the ER
                                                          // so that it stays consistent when resized
      currentTrack=int(sizeRatio);
      animFrame=pointerHead[currentTrack-1];
      placeAll();
}

  boolean over_resize_handle() {
    if(mouseX > erx2-10 && mouseX < erx2+10 &&
      mouseY > ery2-10 && mouseY < ery2+10) {
      return true;
    }
    else {
      return false;
    }
  }

   boolean showzoom() {
    if(mouseX > erx2-20 && mouseX < erx2+10 &&
      mouseY > ery1 && mouseY < ery2-20) {
      return true;
    }
    else {
      return false;
    }
  }

  void draw() {
    image (erImage,erx1,ery1,erx2,ery2);  // draw the background bitmap of the ER
    noFill();
    if(over_resize_handle()) {
      fill(255);  // if mouse is over the ER handle in lower right, fill it white
    }
    rect (erx2-15,ery2-15,10,10);  // draws the handle in the lower right of the ER
    erChannel.draw();
    erPumpdots.draw();
    int numDots = int(ca_er[animFrame]*dotdensity*minSizeRatio/(max_ca_er*sizeRatio)); // dot density
    for (int i=1; i<numDots; i++){  // draw the dot-density pattern inside the CYT rectangle
      int xloc=int((erdotlocations_x[i]*(erx2-erx1)*0.8)+erx1+(erx2-erx1)*0.1);
      int yloc=int((erdotlocations_y[i]*(ery2-ery1)*0.8)+ery1+(ery2-ery1)*0.1);
      point(xloc,yloc);
    }
   if(showzoom()) {
    ellipse(erx2-((erx2-erx1)*0.05),erChannel.chPosition,10,10);  // draw the zzome din area of the calcium
channels
    line (erx2-((erx2-erx1)*0.05), erChannel.chPosition-5, erx2+50+zoomsize/2, ery2/2); // create zoom lines
    line (erx2-((erx2-erx1)*0.05), erChannel.chPosition+5, erx2+50, (ery2+ery1)/2+zoomsize/2);
    image(zoomviews[image_number[animFrame]], erx2+50,ery2/2); // place the image, based on the data model
   }
  }

}

class Flowdot {
  float s;  // these would be x or y coords, but they are relative positions rather than screen positions
  float u;  // ditto.
  float mydisp;
  Channel myChannel;

  Flowdot (Channel a_myChannel){
    myChannel=a_myChannel;
    mydisp=random(1)*2.0-1.0; // create a random distribution of calcium dots in the channel
  }

  void place(float s_pos, float u_pos){
    s=s_pos;
    u=u_pos;
  }

  void drawFlow(){
    s=s+myChannel.flowRate*myChannel.flowData[animFrame];  // draw the vertical flow line, MAPPING a value
                                                            // for ec to the width between lines
    if(s>myChannel.chEnd){
      s-=myChannel.chLength;                  // if x is not in the channel area, set it to the leftmost
part
    }
    if (s<myChannel.chStart){
      s+=myChannel.chLength;
    }
    //line(s,u-myChannel.chWidth,s,u+myChannel.chWidth);
    if (!myChannel.isvertical)
    {
      point(int(s),int(u+myChannel.chWidth*mydisp));
```

```
    }
    else
    {
      point(int(u+myChannel.chWidth*mydisp),int(s));
    }
  }
}

class Pump {

  float pumpX;
  float pumpY;
  float pumpSize;
  float line_x;
  float line_y;
  float theta;
  float r;
  float[] rotateSpeedData;
  float rotateSpeedScale;
  int numlines;

 Pump ( float[] a_rotateSpeedData,float a_rotateSpeedScale, int a_numlines){
  rotateSpeedScale=a_rotateSpeedScale;
  rotateSpeedData=a_rotateSpeedData;
  numlines=a_numlines;
  theta=0;
 }
 void place (float a_pumpX, float a_pumpY,float a_pumpSize ){
   pumpX=a_pumpX;
   pumpY=a_pumpY-5;
   pumpSize=a_pumpSize;
    r=pumpSize/2;
 }

 void update(){
   theta+=rotateSpeedData[animFrame]*rotateSpeedScale;
   if (theta<PI/2||theta>(3*PI)/2){
     if (rotateSpeedScale>0) {theta=PI/2;}
     else {theta=3*PI/2;}
   }

 }

 void draw(){
  fill(#666666);
  noStroke();
  noFill();
  float thetainc=PI/numlines;
  float currtheta=theta;
  for (int i=0;i<numlines;i++)
     {
       float alphaval=255*pow((1-abs(currtheta-PI)/(PI/2)),0.5);
       stroke(255,255,255,alphaval);
       line_y=r*sin(currtheta);
       line_x=r*cos(currtheta);
       line(pumpX,pumpY,pumpX+line_x,pumpY+line_y);
       currtheta+=thetainc;
       if (currtheta>3*PI/2) currtheta-=PI;
     }
  image(pumpchannel,pumpX-pumpSize/2+5,pumpY-pumpSize/2+5);
  }
}
```

APPENDIX G: CalViz assessment questionnaires

Learning Questionnaire

Your participation in this experiment has no effect whatsoever on your assessment for the MRes course. It is strictly voluntary and you may leave at any time, without providing a reason.

For each of the following questions, select the correct answer.

1. How does the calcium oscillation period in this system change as the size of the ER increases?

   a) it increases
   b) it decreases
   c) it is unaffected

2. What elements are in the model (circle all that apply):

   a) endoplasmic reticulum
   b) cytoplasm
   c) nucleus
   d) rhibosomes
   e) calcium channel
   f) calcium pump
   g) cell membrane
   h) cell surface receptors

3. The flow rate of calcium through the membrane pump affected by: (select all that apply)

   a) the calcium concentration in the ER
   b) the calcium concentration in the cytoplasm
   c) the size of the ER calcium channel
   d) it is not affected by anything

4. During what proportion of the cycle is the cytoplasm calcium increasing when the ER size is at maximum?

   a) 0-25%
   b) 25-50%
   c) 50-75%
   d) 75-100%

5. How is this proportion affected by reducing the size of the ER?

   a) it decreases
   b) it increases
   c) it is unaffected

6. Is the concentration of calcium in the ER normally greater than that in the cytoplasm?

   a) yes
   b) no
   c) don't know

7. How many channels are there on the ER membrane?

   a) One
   b) Two
   c) Three
   d) Four
   e) More than four

8. Net flow through the ER calcium channel(s) is:

   a) always outwards from the ER to the cytoplasm
   b) always inwards from the cytoplasm to the ER
   c) in both directions

9. The openness of the ER calcium channel(s) is determined by: (select all that apply)

   a) the calcium concentration in the ER
   b) the calcium concentration in the cytoplasm
   c) the size of the ER
   d) it is not affected by anything

10. Regarding determining the correct answers in the above questions, circle the answer corresponding to your opinion of how difficult this was:

    a) easy
    b) medium difficulty
    c) hard
    d) very hard

Opinion Survey

Your participation in this experiment has no effect whatsoever on your assessment for the MRes course. It is strictly voluntary and you may leave at any time, without providing a reason.

I would be more inclined to further my studies of this biological system as a result of using this visualisation.

Tend to Disagree 1 2 3 4 5 Tend To Agree

I liked using this visualisation.

Tend to Disagree 1 2 3 4 5 Tend To Agree

Visualisations like this make computational biology more interesting.

Tend to Disagree 1 2 3 4 5 Tend To Agree

I thought the visualisation was confusing.

Tend to Disagree 1 2 3 4 5 Tend To Agree

I would like to see more visualisations of cell physiology.

Tend to Disagree 1 2 3 4 5 Tend To Agree

I felt that I understood calcium oscillations better after using the visualisation.

Tend to Disagree 1 2 3 4 5 Tend To Agree

This visualisation was helpful for me.

Tend to Disagree 1 2 3 4 5 Tend To Agree

I think this sort of visualisation makes the subject easier to learn.

Tend to Disagree 1 2 3 4 5 Tend To Agree

I thought the visualisation was boring.

Tend to Disagree 1 2 3 4 5 Tend To Agree

I thought the visualisation was easy to use.

Tend to Disagree 1 2 3 4 5 Tend To Agree

It would be very helpful if you would provide any other thoughts you have about using this visualisation.