Using Sketching to Aid the Collaborative Design of Information Visualisation Software - A Case Study

Brock Craft, Paul Cairns

UCL Interaction Centre (UCLIC), Remax House, 31/32 Alfred Place
London WC1E 7DP, United Kingdom
brock@craft.org
http://www.uclic.ucl.ac.uk

Abstract. We present results of a case study involving the design of Information Visualisation software to support work in the field of computational biology. The software supports research among scientists with very different technical backgrounds. In the study, the design process was enhanced through the use of sketching and design patterns. The results were that the use of sketching as an integral part of a collaborative design process aided creativity, communication, and collaboration. These findings show promise for use of sketching to augment other design methodologies for Information Visualisation.

1 Introduction

The field of Information Visualisation (IV) is concerned with the development of software that allows people to visually explore and understand complex datasets. These tools use various techniques for the presentation of information that range from those used in graphic communication [1] to complex interactive systems that represent data in multiple views, possibly even whilst that data is being fed into the system [2]. At the heart of these interactive systems is the principle that active engagement with the on-screen representations takes advantage of the visual processing capacity of the brain to amplify cognition [3]. IV systems may be either standalone or web-based applications. Currently, some online examples of these IV systems are available at: www.hivegroup.com, www.smartmoney.com, and www.cs.umd.edu/ hcil/research/visualization.shtml.

As with any complex software, the creation of Information Visualisation tools is difficult and depends on the general software concerns of fitness for purpose, context of use, modes of interaction, integration with existing workflows as well as the specific concerns of how and what to represent visually, and how to make the interaction available and intuitive to the user. In the field of IV, there are few methodologies that prescribe a suitable approach to the development of IV tools. Instead, developers appear to rely primarily upon guidelines or on inspiration from the many novel and effective visualisation tools that have already been developed [3]. This is particularly true for those who are not familiar with the techniques and principles of IV.

Please use the following format when citing this chapter:

Craft, B., Cairns, P., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 103–122.

A surprising gap in these guidelines and descriptions of development is the absence of detailed discussion concerning the role of sketching and drawing when designing new IV tools. As visual representation is so central to the nature of IV, we hypothesize that sketching almost certainly occurs, however its role, relevance, and purpose in the design process is not discussed. This suggests that, when used, it is done so in a free-form manner rather than in some more disciplined and methodical manner as a key part of the development process. By contrast, in more mature design processes such as graphic communication and architecture, sketching is used as an integral and systematic part of the design process [4, 5]. And while sketching has been discussed in generic user interface design – paper prototyping can be seen not only as a fast prototyping technique, but also as 4-d sketching [6] – the field of IV has largely ignored its benefits. The overall aim of this work, then, is to consider the use of sketching in the development of Information Visualisations with a view toward suggesting how it can be used effectively and systematically to support developers.

This paper reports on a particular case study in which an IV tool was developed to support the communication between biologists and mathematicians in a computational biology research project. Extracting valid research conclusions from such a case study is always problematic. There is necessarily, as with any complex software development process, no comparable development process in terms of timing, goals, team, and organisational context. Nor is it appropriate that the project be allowed to fail because of intrusive adherence to the research protocol. The approach we have taken therefore is to use Action Research (AR) [7] to aid and support the developers whilst deriving knowledge in the form of a case study. Hopefully with this approach, as will be discussed below, we are able to extract insights into the use of sketching in designing IV tools that not only supported the team we were working with but also have significance beyond this study.

Given the general craft-approach to designing IV tools, we used a framework of IV patterns proposed by Wilkins [8] as a way to introduce IV concerns to the case study team. These patterns are intended to function as design patterns in the architectural [9] and software engineering sense [10, 11] and thereby lead to effective IV development. Aimed at improving usability, they are based on the guidelines and examples that are commonly referred to in IV research and hence, presented a suitable starting framework for the team. It was felt that this approach would bring a structure that would support and encourage sketching around key IV ideas.

The key findings from this case study were that sketching substantially aided the development team in creating their visualisation software. Specifically, sketching helped team members in three key areas: Communication, Creation, and Collaboration. Sketching improved Communication by allowing team members to simultaneously share ideas verbally and visually, to clear up misunderstandings, to build up simple ideas into complex ones, and to record the activity for later reference. Sketching improved Creation by helping people to think about design problems in new ways, and to work out novel solutions to problems. It also helped them to overcome the intimidation of the design requirements and to overcome "mental blocks". Finally, sketching supported Collaboration by helping team members to share complex ideas with the group in a common space, by facilitating group input on new ideas at the time of ideation, and by helping the designers to suggest and try out interaction ideas with the group through scenarios. With the aid of a body of knowledge about effective IV solutions in the form of IV design patterns, the group were able to determine concrete solutions to the design problems that they had identified in the requirements gathering phase of the project. The result was that whereas before the sketching intervention the participants had only limited ideas about how to proceed with their project and were not able to identify a way forward, after the intervention the team had proposed eight specific Information Visualisation software modules as strong candidates for development and had prioritised exactly which of those projects would yield the most benefit to the end users and project stakeholders.

With regard to more general IV software development, the results of this work seem to indicate that it is fruitful to systematically employ sketching in the IV design process. Effective use of sketching as applied in this case study can lead to creative, visual solutions. This is especially relevant to Information Visualisation, as IV relies upon novel visual representations to enhance cognition of information. Particularly in the early phases of ideation and brainstorming, sketching helped team members turn a set of requirements into concrete ideas for visual representations and interactions. This helped them to overcome obstacles to creativity and to effectively address the needs of the project during the design phase.

2 Research

The case study involved the creation of software to support the working practice of biologists and mathematicians on a Computational Biology research project at University College London. Known as the Beacon Project, its goals are to create computational models of biological processes in the human liver, such as metabolisation of glucose. It is hoped that such models will help biologists to better understand liver function and to perform experiments that are not possible in the lab.

The need for Information Visualisation software arose out of problems that Beacon Project members encountered in the course of their work. Most acute among these was that the project team members could not effectively collaborate on developing models because this requires very specialized knowledge in rather different fields of research. Computational models that can begin to describe liver function are of a level of complexity that requires the knowledge and skills of specialist mathematicians. Similarly, intimate knowledge and intuitive understanding of liver metabolisation require many years of biomedical training and research. As the Beacon project matured, a challenge became apparent. The project mathematicians needed to know that the models they were in the process of developing were accurately expressing liver function. They also needed the biologists' help in determining where to make improvements when those models were inaccurate or wrong. But they could only obtain this guidance with the collaboration of biologists who were not able to understand the mathematics involved at a sufficient level of specificity to make useful recommendations. The group decided that a software solution might address this problem, if it would help in bridging this knowledge gap.

The Beacon team determined that the best way to do this would be to express the activity of the models visually and interactively which would aid understanding and cognition of all the team members; in short, they needed Information Visualisation software. However, computer scientists on the team did not have any experience in designing such tools. The Beacon project team contacted researchers in Information Visualisation at UCL Interaction Centre (UCLIC) to address these challenges and to aid them in arriving at a design for possible solution. The UCLIC team offered to aid the Beacon team in the IV design process and with their agreement, to use it as an opportunity for a case study in the design of Information Visualisations.

2.1 Method

As the research involved the study of human design processes in an information systems work environment, we used a qualitative and collaborative research method called Action Research (AR) [5]. Based upon post-positivist and interpretive research paradigms, the goal of Action Research is to derive useful empirical knowledge through the study of human phenomena in the environment in which they occur. Moreover, rather than isolating the researcher as a disinterested observer, AR situates the researcher squarely in the environment under study and invites the collaboration of both researchers and subjects as co-participants in the enquiry. The philosophy of the AR approach is that some knowledge about human activity can best be gained from the natural environment in which phenomena occur and that the acknowledged participation of the researcher and the subjects as co-participants are necessary for understanding them. Furthermore, AR recognizes that the researcher has knowledge that may be relevant to the activities under study and may contribute this to the research setting. Indeed, AR is particularly suited to studying processes where the organisation can be aided by the expert knowledge of the researcher. Thus, the visualisation design process in the Beacon Project which comprises the case study research reported here was done in a participatory and open manner. We believe this was particularly well-suited for understanding how sketching is useful early in the visualisation design process. Another method, such as conducting research in a controlled laboratory setting would not have yielded the kind of knowledge derived from real-world design problems.

In terms of the software development methodology, we used an approach based upon generic, Human-Computer Interaction oriented frameworks of iterative software design [12], which we call the Information Visualisation Design Framework (IVDF). Figure 1 shows the IVDF development process. It involves Requirements Gathering, Visualisation Design Activities using sketching and design patterns, Prototyping, and Evaluating. It situates sketching and the use of design patterns early in the development process and importantly, encourages this activity at the stage where ideation and creativity are involved. The focus of this case study was on the use of sketching and design patterns in the Visualisation Design Activities phase.

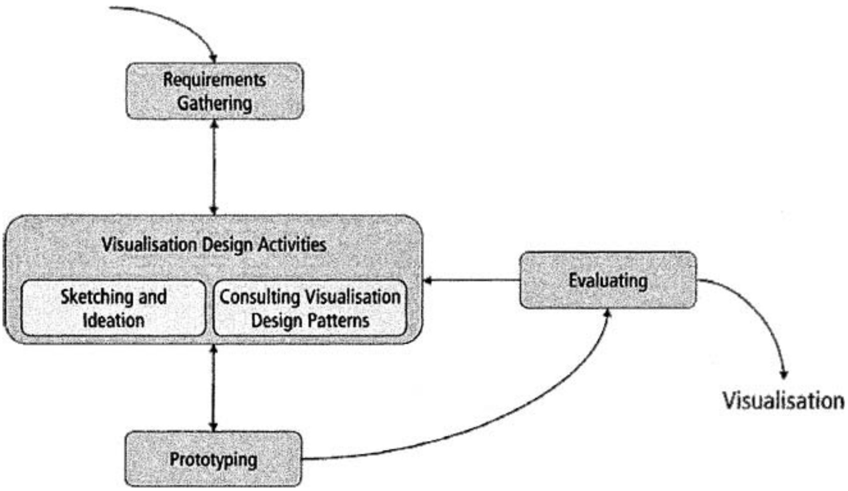

Fig. 1. The Information Visualisation Design Framework.

The first part of the IVDF is the Requirements Gathering process. Like requirements gathering techniques from other HCI-oriented development approaches [13], this involves collection of data from all stakeholders, whether by observation, surveys, interviews, or other means. We used personal interviews with project team members and end-users as the primary method for gathering requirements. Combined with design specifications that had previously been identified by project managers, these formed the Requirements that were used to guide the design sessions. Prior to the Design Activities Phase, these two sources were combined and it was agreed with the project team that these would be the Requirements used for development.

The Design Phase sessions were then held in several instances over a three month period. Meeting in small groups of not more that six people, willing participants from all parts of the Beacon Project engaged in the activity with the facilitation of a researcher from UCLIC. The design sessions were held at the Beacon Project premises in a conference room that was relatively free from distraction. They were typically two to three hours in duration including breaks. Sketching materials, such a large pad of heavy A3 drawing paper and various coloured writing tools were provided by the facilitator. It was agreed that audio from the design sessions would also be recorded and transcribed for later analysis.

The UCLIC researcher mediated the design process and invited active participation from the team members. In keeping with the AR methodology, the UCLIC researcher also offered knowledge and experience about effective IV solutions, where appropriate. Design patterns were used to demonstrate useful IV techniques and to aid brainstorming. Also, where appropriate the facilitator actively encouraged participants to use sketching tools during the design process and indeed, used these tools to elucidate ideas and explore possibilities with the group. To aid the research, active feedback about the design process was elicited where appropriate. In this way, the design sessions were completed by collaboration of the researcher and the participants and with the acknowledgement that the design activity was part of a research process.

2.2 Results

After completion of the 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 team members in follow-up interviews. In many instances, verbatim transcripts of the design sessions were used to gain feedback from participants about the nature of the design process and about how sketching, the design patterns, and the activity of the UCLIC facilitator as subject matter expert, were useful. Further analysis revealed patterns of activity that the sketching made possible.

Drawing from the audio recordings, design sketches, and follow-up discussions, we have identified three categories of activity that sketching facilitated: Communication, Creation and Collaboration. These categories emerged from the many different ways that team members used sketching in the design process. We have chosen highlights from the case study to illustrate how sketching helped Creation, Communication, and Collaboration. These highlights are presented below.

For each of the examples, we have excerpted some of the dialogue recorded during the design sessions and have provided the corresponding sketches that participants created. For the sake of brevity, only the first letter of participants' names is used in quoted conversations. While the data from the design sessions contained many more examples than could be presented in this paper, we have chosen highlights which we believe to be most representative.

2.3.1 Communication

Perhaps the most significant benefit that sketching brought to the design process in this case study was enhanced communication. The adage "a picture is worth a thousand words" is a good summary of the way that sketching helped in the communication of team members. One of the most difficult parts of designing in groups is ensuring that everyone understands the complex issues under discussion. Sketching helped to communicate new ideas quickly, to support verbal communication, to confirm understanding, to form a written record for later reference, to build complex ideas, and to explain difficult concepts. The following examples from this case study show how Communication was supported by sketching and how it facilitated the design process.

Communicating ideas verbally and visually, triggering understanding. In this example, a participant uses sketching to explain how the interface will graph particular mathematical variables in the computational model. This helps a mathematician on the team, J, to understand how the software will implement constraints on variable output and make those apparent to end users in an interactive graph. Note that to emphasise his point, P adds the words "Min" and "Max" to the sketch during his verbal explanation. Refer to Figure 2.

"P: This is basically what in your interface shows you a graph of a particular variable – I'll draw it again, I suppose. So this is your model view, and you have your variable tab. And we know there are problems with tabs, but. And we have the name of the variable here, which is *P* subscript *r*. And then it's got a graph here. And what we want to be able to do is add a button here that gives us some sort of interpretive result for that. So what we want to do is do "Max" and "Min" and then it would go "Max", "Min"."

J: Oh, that was important, yeah.

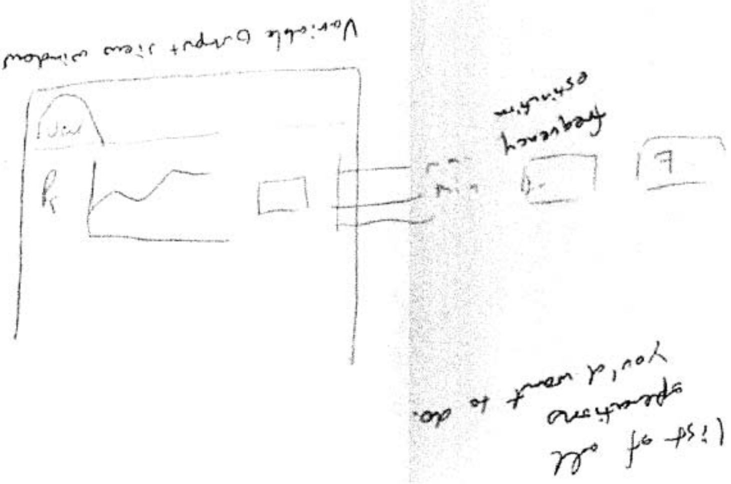

Fig. 2. Communicating ideas visually and verbally at the same time. The participant describes the "Min" and "Max" parts of the sketch as he draws them, to clarify the proposed interaction. Note that the sketch is right-reading the textual commentary is upside-down.

Communicating familiar knowledge, where the original source material is not at hand. In this example, J, a mathematician, uses a sketch to describe a list of parameters that are important to the project. The parameters were identified during requirements gathering, but he wishes to recall everyone's attention to them. He sketches a representation of a colour-coded table to emphasize his point. He creates a very loose sketch of a table, omitting the colour coding, and points to different parts of it as he verbally describes the source material. This looseness characterizes its transient importance. Once the team member was certain that the rest of the group understood him, he stopped embellishing the sketch. This behaviour was common for other sketches in the study.

"B: Well, I haven't seen it. But remember, I was asking, "Do you guys even know what the universe of parameters is, entirely?", about three months ago?

J: It's not worth – actually, I'll sketch it on a piece of paper because that's more in keeping. So, it's a long list of the table with numbers and stuff, and names, and things. And the important thing is, which keeps it really visual is that there are blocks of colour on here for: big red ones meaning, "We need to know this and we haven't done it." Big green ones meaning, "We do need to know this and we've done it." et cetera. And orange ones for, "We've got some idea or something."

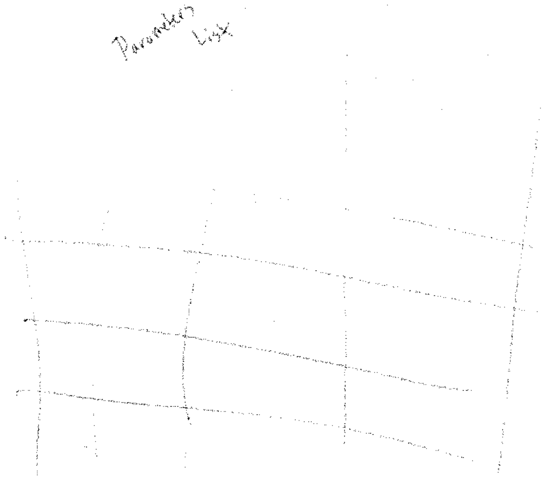

Fig. 3. A quickly rendered sketch referring to knowledge that team members already were familiar with. The participant omitted adding the color that was in the original table, as it was not necessary to make his point.

Confirming understanding of something. In this example, one of the team members explains to a biologist how a particular computational model might be represented by an interactive graph in the proposed interface. Referring to the sketch (Figure 4) he uses it to ensure that his colleague understands how the interactive model will allow for exploration of results that the model will generate.

"P: What we're talking about is a method of testing the models.
M: Testing the models.
P: So, here we have sliders that allow us to adjust these values, even if we an adjust them to ridiculous proportions if we want to and we can still get results out of the model to see what the model does. In exactly the same way we can adjust the model to contain connections just to see what happens – even if those connections we thing in the real world, don't exist. What happens if these were connected? And then that would give us a result. And that would allow us to test things.
J: In this example we know that there isn't such a connection, so it's not a very good example, but what if we weren't sure.
M: But maybe tomorrow they'll discover that it *is there.*
J: In another one, we can have a thing where we have "on" or "off" for a particular kind of connection here. And then we can switch it on or off and see how that affects the graph.
M: Uh-huh.

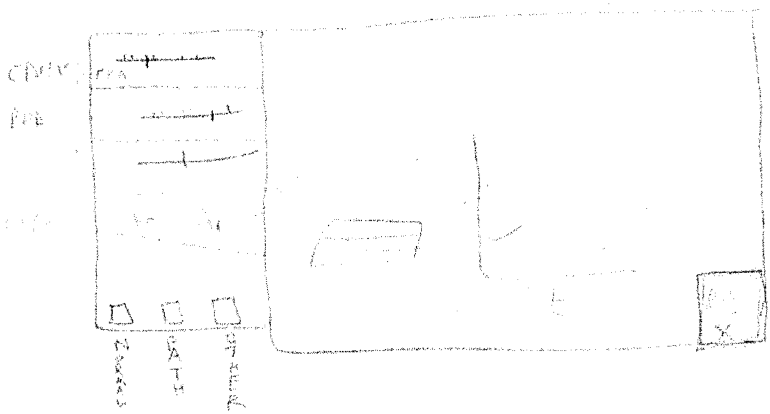

Fig. 4. One participant refers to a sketch to confirm that another team member understands in interaction concept, indicating in the sketch the different objects and their behaviours.

Forming a record of the thought process that can be used later. Participants often used sketches created previously as a point of dialogue for further discussion or for ideation. Here, one person reminds the group of a particular solution that was arrived at earlier. He locates the sketch where that solution had been arrived at and refers to it whilst describing its different attributes, in this case "lumps" and "holes". This particular solution, Model Interface Matching, became a major component of the final specifications. Refer to Figure 5.

"P: Another thing we are looking at here is the difficulty of constructing models and the interfaces those models have with each other and we drew those pictures of the jigsaw pieces?
B: Do we have the pictures?
P: Aha, note for…research: the pictures once again come in useful! We drew these pictures of the pieces and then joined together.
B: [reading from sketch] "Text describing lump."
P: Text describing *hole.* And then the person with the big brain looks at the two and decides whether the *lump* and the *hole* are compatible.
B: Right. What do you call that?
P: It's interface matching. "Model interface matching".
J: Yeah, it is."

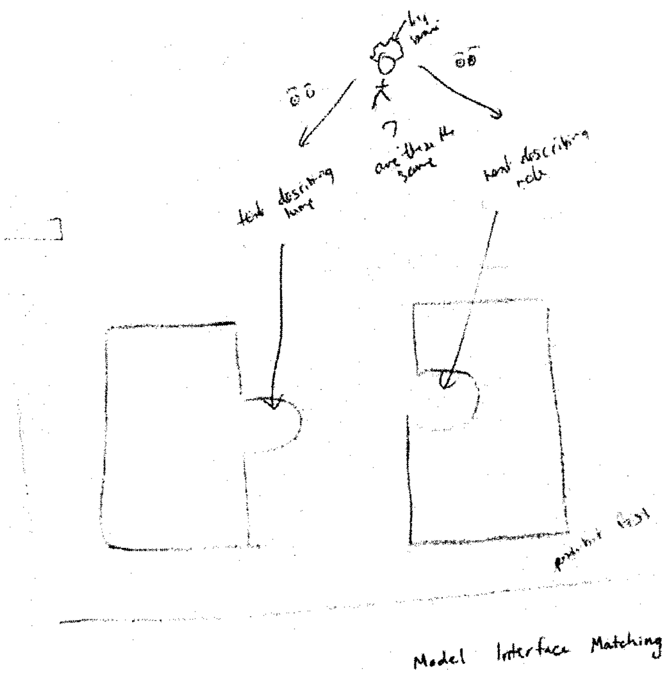

Fig. 5. This sketch formed a record of several decisions about how biologists, represented by the human figure at the top, would use the software to make decisions about connecting mathematical "model interfaces". The sketch was referred to many times over several sessions.

Building complex ideas from simple ones. All of the sketches were contained in a single A3 drawing pad. This sketch history made it easy to refer to previous work and build complicated ideas out of simple components. In this example, members used the previously created jigsaw-puzzle metaphor as a solution for how the software will determine a particular calculation known as waveform relaxation or "WR", depending on user input. The point to note here is that the earlier idea of the jigsaw-puzzle metaphor was available to the participants because they had a written record of it in the sketchpad.

"J: Right, so, the point is that actually because of the mathematics of it, once you start using anywhere, a Oval interface, you have to have Oval interfaces everywhere because you're using the waveform relaxation [WR] algorithm. Or so you might think. But in fact, what you can have is this: when I click – supposing I've got another model here with one of those things, and supposing these are actually separate interfaces – so *that* is an alternative for *that* – I'll colour those in black to represent this.

P: Yes. Good point yes. You can have knobbles that represent – that's quite complicated, really. Yeah, okay.

J: And, I've shaded *that* to say that *that's* matching *that*. Now, when I plug this into here, that disappears and from over here, the computer makes a, one of these –

B: Which is a - ?

J: A thing with a triangular hole and an oval prong emerges from nowhere. The Orchestrator says, "Ah-ha! He's got a mixed-mode WR rate-calculator *thing*, so I'd better cap the rate-calculator – "

P: You've made a transformer? Cool.

B: More than meets the eye."

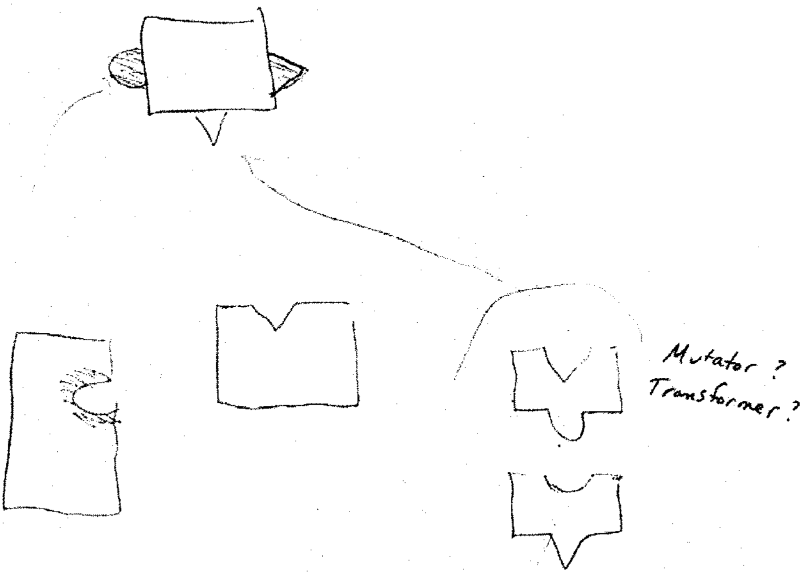

Fig. 6. In this sketch, participants used a previously created puzzle metaphor to build a more complex widget to perform a mathematical calculation called a waveform relaxation.

Explaining difficult concepts. In the domain of computational biology there are naturally many complex concepts. Very frequently it was necessary to reiterate the meaning of these concepts to make sure that everyone in the group had the same understanding. Sketches were often used to augment this process. Here, a team member illustrates that concept of Diet Function for the rest of the group while sketching. See Figure 7.

"J: Right, one of the things we're trying to do with our models is work out how the glucose in your blood changes as you change the amount of food that you eat and the exercise that you do. Diet function, if you like, or food function is, you know, I wake up in the morning and going to make some breakfast...and then I go for a run. And then I have a big lunch and then I kind of sit and don't do very much for a while. So I'm fairly sedentary. And then I have a huge dinner and then lots of beer and then go to sleep, what have you. This thing goes up and down accordingly. We need one of these as the input to our model. And at the moment in order to enter this, it's either numbers in a table or..."

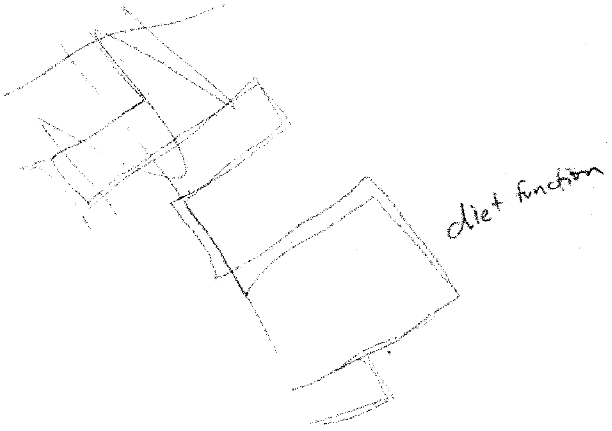

Fig. 7. Participants used sketches to make difficult concepts easier to understand.

2.3.2 Creation

One of the major benefits that sketching brought to the design process was that it allowed people to take ideas in their head and to try them out on paper. This creativity-through-drawing process is well recognised in the fields of graphic communication and architecture [4, 5], but has been little addressed in software engineering. The following three examples from this case study show where how Creation was supported and how it resulted in the emergence of new ideas and capabilities for the software.

Creating abstractions and mnemonics. In the first example, members of the group use sketching to construct their own abstract representations for specific concepts. They create visual mnemonics to encapsulate their ideas. They subsequently used the abstractions as shorthand for explaining the behaviours that the software should support, such as zooming. Essentially, they created the rudiments of a visual language to play with design ideas more effectively. Their Creation helped their Communication. In this example, team members reflect on this activity and how it was helpful:

"J: And we've developed, actually, some mnemonics for different things happening on the computer screen. We've been using what P just drew, a pair of lines with a larger gap at one end than at the other end to mean *zoom*, because that's standard

– you know the thing with the two circles and *that one.* And is there somewhere where we've drawn somebody dragging something?

B: This idea? [pointing to the sketch]

J: Yeah, somewhere or another we've drawn somebody dragging something and drawn an arrow and that kind of thing.

P: There, because that's been dragged over to that window, the "Sandbox" window.

J: So, it may be interesting for future attempts to do this by you guys to teach the clients, us, a vocabulary for drawing computer screens on pieces of paper.

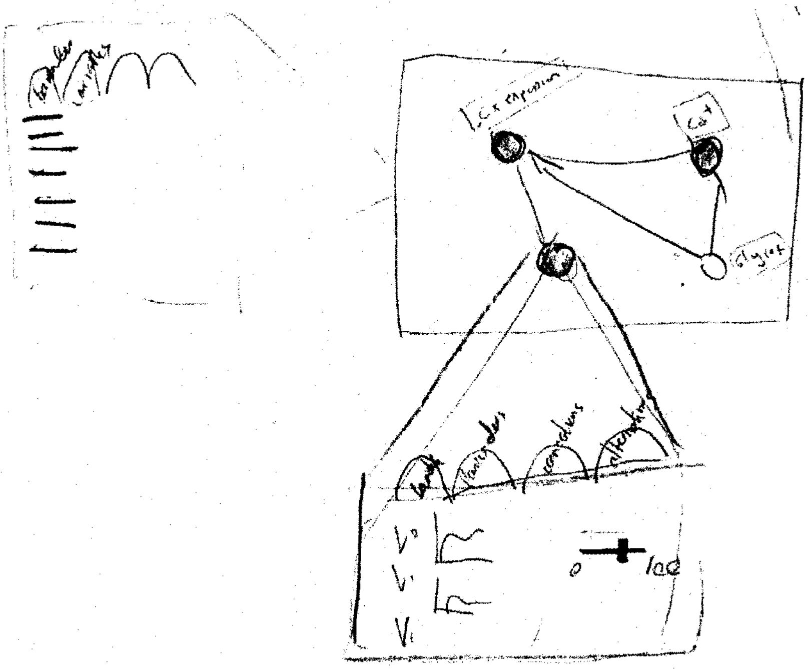

Fig. 8. Use of an abstract representation of zooming to depict the intended interaction of the software. This depiction was subsequently adopted for other parts of the design process and other kinds of zooming.

Figure 8 shows the use of this zooming abstraction, represented by converging lines, as sketched by the participants early in the design process. It is not a novel visual abstraction for the purpose of representing zooming. What is creative here is that the case study participants used the representation as a mnemonic for reminding them how the interface was intended to behave and used this for making future decisions about how the software would work. The also applied this technique in representing different kinds of zooming, both visual and semantic.

Another example is the use of an abstract visual metaphor to constrain user input. One component of the visualisation allows 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 designers 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 9 shows an example of this novel representation, which was created by the participants early in the course of exploring this problem.

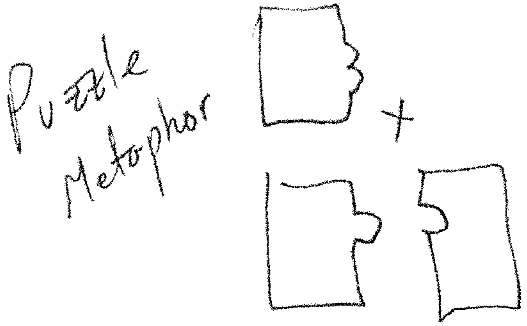

Fig. 9. A visual metaphor for software behaviour.

Creating solutions to problems and novel widgets. In the next example, three participants use a sketch to work out how to solve a problem that the visualisation should show what data ranges are supported by the biology literature. Working together, they create and embellish the novel idea of a slider widget that interactively displays thumbnails of academic papers that support data values in a parameter range. See Figure 10.

"J: I'll just put a little note next to the parameters. One of the things the parameters need to have is: when you drag a slider for a parameter to the point which corresponds to a value with a known supporting text, then it flashes and then it allows you to click and to bring up the paper.

P: That would be really cool, actually! You can imagine a situation where you've got the slider that looks like this and when you're in the middle you've got a list of five papers that support that value and as you get to the end it goes down to one paper and then you go off the end and it says "This is a guess. You're making this number up."

B: That's brilliant yeah. Because then you can know when your stuff is supported and when it's not supported.

J: [Sketching] And you can see, here's a little window. So here's our slidebar. Here's the 'Paper Names' window. Here is – I'm fantasizing now, this would be really hard to implement – a photograph of the cell plate that was used in each of the papers.

P: [Indicating on J's sketch] So when you're here, what you get is this list of five. And when you're *here*, what you get is a much smaller list, you know, two. And then when you're right on the end, then it just says: 'guess'."

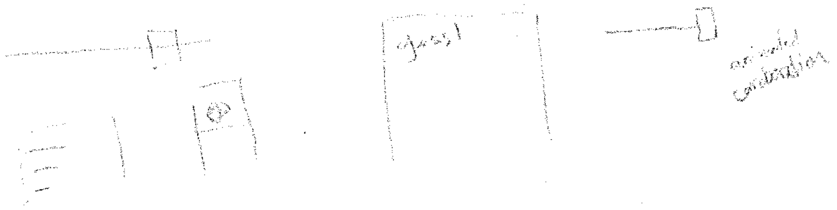

Fig. 10. Creating solutions using novel widgets: the "Provenance Slider".

Sketching also helped participants to overcome the intimidation presented by a stack of requirements specifications with no clear design solution. The prospect of having to overcome a great challenge often creates so-called "writer's block" where designers are unable to begin to work through the design problems. By talking together about each of the design problems and simultaneously sketching out ideas on paper, these blocks were overcome, and creative ideas began to emerge. At the end of the case study, during a reflective discussion, it was the universal opinion of the participants that the sketching helped to overcome inertia and to get the creative process started.

2.3.3 Collaboration

The third major way that sketching facilitated the design process was by helping people to work together. Each of the participants brought his or her unique ideas to the design process, and needed to share them with others. By actively sketching these ideas with others, group input and ideation was made possible. Indeed, there were many points when two or more participants were actively sketching on a common work area together as they played with design ideas. Sketching further supported Collaboration by allowing team members to create scenarios from which to elicit feedback and participation of others. Sketching also triggered understanding among designers which allowed them to build on each other's ideas.

It is important to note that the quality and quantity of collaboration was very significantly affected by the design setting. In this case study, we sought to create a situation conducive to collaboration by ensuring that the environment was comfortable, that the groups were small (no more than 6 people), and that everyone had an opportunity to participate. As facilitator, the UCLIC researcher actively encouraged other team members to try to sketch out ideas for others and supported collaboration in the sketching process.

Creating a scenario and acting it out. There were many occurrences of this. In this example, it is interesting to note that as J creates the scenario, he uses onomatopoeic sounds to emphasize the activity that would occur over time, whilst gesturing on the sketch with his pencil. His portrayal of the scenario conveys to the group what he suggests might happen as users engage with the visualisation. This also allows others to give their input into that behaviour. Refer to Figure 6.

"J: So what happens here is, if I've already plugged those together and then I plug that in, these two bricks come apart and a pair of these [drawing] they go [making onomatopoeic sounds] and these are actually new software entities that the Orchestrator will have to spawn at runtime.

B: And not only are they doing that, but they will have to do that as you did just now, verbally, as an *animated thing*.

J: Of course. Absolutely."

Helping people can build ideas together. In this example, one of the participants, P, explains an idea for a "sandbox" where biologists can perform their own experiments on data. He uses a sketch created in a previous design sessions to describe what activities would be possible in the visualisation, indicating on the sketch as he describes it. He points out different objects in the sketch saying, "this" to help communicate the

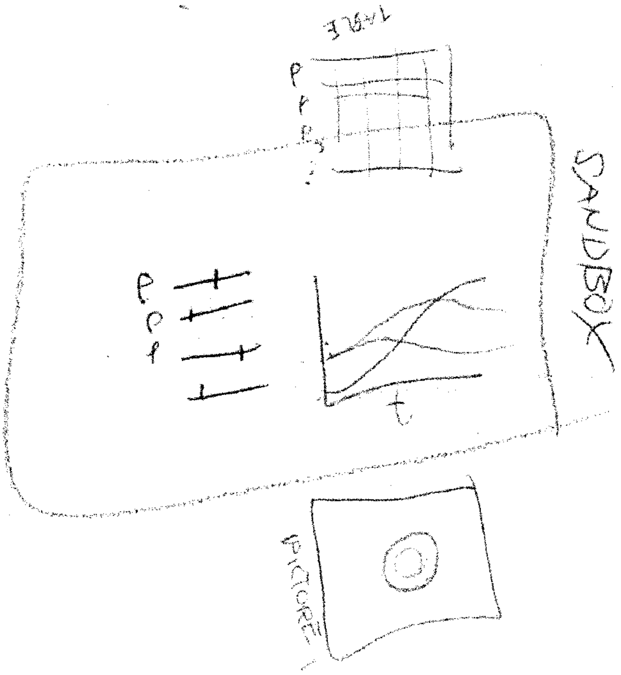

Fig. 11. Facilitating collaboration through sketch sharing.

sandbox idea (Figure 11). This helps biologist M to understand how she might be able to use such a tool and she suggests a possible application. In later design sessions, this allowed her to elaborate on the sandbox, and add her own suggestions for functionality.

P: Finally, we have this idea of a 'sandbox mode'. The idea is, you start off with a blank screen and you say, "Okay in order to look at this really big complex thing that I don't understand, what I really want is *this* parameter here, that I've plucked from over there, and *this* parameter here, and *this* parameter here that I – these are just the ones that I happen to be interested in. And then I want a graph of *this*. And that gives you an interface which just has those things on it and nothing else. And you can add and you can take away from this as much as you want to make it as simple or as complex as you want. If you want to, you can adjust everything.
M: Uh-huh.
M: Like, now I'm going to remove this pathogen, and see what happens if I remove it?
P: Yeah.
M: Or glucagon, or whatever."

Supporting collaborative decisions. Here, the facilitator explores with participants practical limitations of using one visual metaphor over another, by sketching out the alternatives. As a result of expert knowledge of the mathematical constraints, J was able to rule out the use of the puzzle-piece metaphor in Figure 9 and argue for the patch-panel metaphor in Figure 12. By this means, the participants collaboratively identified the better of the two visual metaphors for this application.

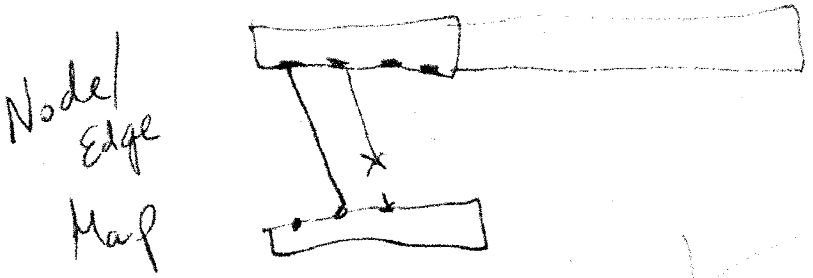

Fig. 12. Choosing among sketched options through collaborative decisions. This "patch-panel" metaphor is a better alternative than the jigsaw puzzle-piece metaphor for the interaction under consideration by the group.

"B: So, here's the puzzle-piece metaphor. Okay another metaphor would be a patch-panel metaphor where you have a piece, [and] a piece here. And they have, essentially, you know, links and receptors.
J: Yeah, this is an old fashioned telephone exchange.

B: It's exactly, the exact metaphor I'm thinking of. And so this visually – you can tell that these two pieces can fit together, whereas these two pieces cannot fit together, right?

P: Uh huh.

B: But the only space you have to work with [within the puzzle piece metaphor] is the perimeter of the visual object in that case. Whereas with these [patch-panels], the perimeter itself doesn't actually encode whether it fits or not, but whether you, say, can draw a link here – it'll either be possible or it won't be possible."

2.4 Outcomes

The final design session included a discussion wherein participants specified what software projects should be produced and prototyped. This was the point at which participants synthesized their knowledge gained during the sketching and design activities and produced a specific set of software specifications. Whereas they had begun the design sessions with no clear solutions in mind, by the end of the process, they were able to articulate exactly what tasks the Information Visualisation software should support, what visual representations should appear on-screen, what interactions should be possible, and how this would support end-users' work processes in computational biology. They then produced a document describing eight sub-projects for which formal specifications could be written and for which code authoring could be started.

3 Discussion

Perhaps the most encouraging result was that the members of the Beacon Project developed eight new software specifications that they did not have before the activity occurred. By encouraging communication, creativity, and collaboration, the use of sketching during the design process allowed participants to evolve new ideas about how the software could solve their most pressing problem on the Beacon Project, namely, allowing both biologists and mathematicians to collaborate in the development of computational models which describe liver function. Moreover, as both biologists and mathematicians were included in the design activities, they were able to more clearly understand the communication challenges among them and to sketch solutions that the software might provide.

Interestingly, as none of the team members were particularly skilled in sketching, they were initially apprehensive and uncertain as to how it would help them to arrive at design solutions. Early in the design activity, the facilitator was more likely to need to prompt participants to the drawing pad and pencils. In contrast, by the end of this process, participants were drawn to use the tools without any encouragement. Their apprehension about sketching had disappeared. People were eager to use the tools to help them communicate and to assist in creating new ideas. They turned to sketching to collaboratively make decisions and rehearse proposed solutions so that colleagues could asses the idea. In discussion during the follow-up interviews, people agreed that they would not have expected this result and that they found it extremely fruitful.

The 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 their research, these ideas tended to remain "silo-ed". They had not yet shared them with one another on the project. Sketching these ideas together aided in this process and allowed people to modify solutions to more tightly address project needs. Moreover, the participants reported that sketching enhanced their effectiveness in a way that was not possible through verbal dialogue alone. The sketching had prompted changes in thinking about and exploring design problems which would not have otherwise occurred.

These changes in communication, creation, and collaboration are activities that are common to many design disciplines. We believe that these categories of activity would not have emerged so successfully without the use of sketching as an integral part of the design process. What is also novel in this case is that sketching proved uniquely suited to supporting the Information Visualisation design – a highly visual domain. As sketching can support any of these activities at any point, it is appropriate for a generative design process where the novel representational and interaction solutions are unlikely to be known in advance. Also, because they are interdependent, they are non-hierarchical. Each activity supports the others and can occur at any time. Better communication helped the participants in this study to collaborate effectively and stimulated their creativity. Likewise, collaboration improved communication and sharing of ideas. Enhanced creativity resulted in mnemonics and visual that helped to facilitate further communication and collaboration. This non-hierarchical quality is very appropriate for problem solving design activities involving brainstorming and collaboration.

This perhaps highlights what makes the research most promising. Combined with theory development from other areas, further understanding of how sketching supports design in groups may emerge and prove useful to practitioners. While this does not mean that the findings from this case study are generaliseable to other projects in strictly positivist terms, the generic nature of the sketching activity and the successes that are already known to other design disciplines makes it likely that using sketching would be fruitful for others who are engaged in designing highly visual, interactive software. In an area where new theory about design methodology is relatively sparse as compared to more established disciplines such as architecture and graphic communication, any new theoretical knowledge can prove useful to others and can spur further exploration.

4 Future Work

As with any qualitative method, the orientation of this research is to develop theory about the phenomena under scrutiny. This does not necessarily mean that findings will be universally applicable. Yet, we believe that additional research into the effectiveness of sketching will confirm the ways in which it was useful in this case study and yield additional understanding about how it might be helpful to others. We also believe further study would clarify how sketching is related to other low-fidelity prototyping techniques such as paper mock-ups, demonstrate when sketching is most useful, and suggest reasons why. We are optimistic that future practitioners will find sketching useful in to their efforts to surmount Information Visualisation design challenges.

Acknowledgements

We gratefully acknowledge the members of the Beacon Project for their assistance and collaboration in this research. We'd particularly like to thank Peter Saffrey and James Hetherington for their valuable contributions.

References

1. Tufte, E.R. (1983). *The visual display of quantitative information.* Graphics Press: Cheshire, Connecticut.
2. Shneiderman, B. (1994). Dynamic Queries for Visual Information Seeking. *IEEE Software,* 11(6), pp 70-77.
3. Card, S., Mackinlay, J., and Shneiderman, B. (1999). *Readings in Visualization: Using Vision to Think.* Morgan Kaufman: San Francisco, California.
4. Lawson, B. (1997). *How designers think: the design process demystified.* Architectural Press: Oxford, UK.
5. Schön, D.A. (1983). *The Reflective Practitioner: How Professionals Think in Action.* Temple Smith: London.
6. Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison Wesley Longman: Reading, Mass.
7. Baskerville, R. (1999). "Investigating Information Systems with Action Research," Communications of The Association for Information Systems, (19) Article 2.
8. Wilkins, Barry (2003). *MELD: A Pattern Supported Methodology for Visualization Design,* PhD Thesis, University of Birmingham.
9. Alexander, C, et al. (1977). *A Pattern Language: Towns, Buildings, Constructions.* Oxford University Press: New York.
10. Gamma, E., et al. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley: Boston.
11. Borchers, J. (2001). *A Pattern Approach to Interaction Design.* John Wiley & Sons: New York.
12. Preece, J., et al. (1994). *Human-Computer Interaction.* Addison-Wesley: Essex, UK.
13. Hackos, J. and Redish, J. (1998). *User and Task Analysis for Interface Design.* John Wiley & Sons: New York.