The role of informal representations in early design

Sara Jones and John Sapsford

S.Jones, J.Sapsford-Francis@herts.ac.uk
Department of Computer Science, University of Hertfordshire,
College Lane, Hatfield, Herts, AL10 9AB, UK

Abstract. Early design activities are of critical importance to the success of a system, yet are fraught with difficulty. This paper presents the results of two small-scale studies which investigated the role of informal and semi-formal graphical representations in the early design of interactive systems. We argue that informal graphical representations may have an important role to play in early design in enabling designers to think creatively about possible design solutions. However, we demonstrate that reliance on informal diagrams as the primary means of communicating and recording design decisions is associated with a number of difficulties. We also identify a number of challenges in relation to the post hoc recording of design decisions using semi-formal notations. We end by discussing the way in which both informal and semi-formal notations may be used to maximum benefit.

1 Introduction

Early design activities are of critical importance to the success of a system, yet are fraught with difficulty. Decisions made in the design stage have far reaching, and sometimes devastating effects on subsequent development work. The design phase is particularly challenging in the development of systems employing new and fast-developing technologies such as multimedia and virtual reality where boundaries are constantly being pushed forwards and a wide range of expertise is required from different members of the development team [3]. In this paper, we focus on situations such as these where:

- novel or state-of-the-art systems are being developed and designers are required to think creatively about different possible approaches to solving a problem;
- design teams are made up of individuals from different backgrounds and with varying expertise so that developing an understanding of the system to be built which is shared by all members of the team is particularly challenging;
- there are no fixed roles within the design team and membership may vary over time so it is important that there should be unambiguous records of design decisions which can be understood by those joining the team later in the project.

We look, in particular, at design activities carried out during the course of the SPIRE project [1], whose aim was to develop a multimedia system that would provide information for staff and students about the integration of students with disability into the Higher Education environment. Development of the SPIRE system was begun in 1994 when the application of multimedia technology was much less common than it is

today. The team were unaware of any systems similar to what was thought to be needed, and creativity was therefore required in considering what system should be built. Five designers were involved during the early stages of the project, each with different levels experience of the application domain, of the chosen development platform, and of the development of information systems in general. Three of the designers had worked together on projects in the past, but two were new to the team and did not share this experience. Finally, different designers attended different meetings so membership of the design team effectively varied over time.

In design meetings for the SPIRE project, and in other design meetings associated with ongoing projects in our own department at the University of Hertfordshire (UH), we have observed that members of design teams often use graphical representations of various forms to communicate their ideas and form a basis for discussion. We believe that much can be learned from experience in these areas, about the process of design, about the role of graphical representations within it, and about what aspects of the process - and the corresponding use of graphical representations - may be improved.

This paper presents the results of two small-scale studies which investigated the role of informal and semi-formal graphical representations in the development of interactive systems. These studies were designed to help us gain insight into what various forms of graphical representation are used for, and what difficulties may be caused by the use of such representations. The following two sections describe each of the studies in turn. The results are discussed in section 4.

2 Study 1: Use of informal representations

2.1 Introduction

A range of different activities are involved at different stages of the design process. Early on in design, there is often a need to search for creative solutions. For example, in the SPIRE project, much effort was expended on trying to devise innovative yet appropriate ways of providing access to useful information about supporting students with disabilities.

If creativity is an important part of design, the tools and techniques used in design should, at very least, allow it to take place. One widely adopted means of encouraging creativity is brainstorming [7]. Although there is some doubt about the efficacy of brainstorming in groups, it is widely accepted as an effective approach for the generation of a large number of ideas [6]. Brainstorming involves generating many ideas whilst avoiding the premature evaluation of those ideas. Thus creativity appears to flourish through the fluent generation of ideas and a postponement of evaluative judgement. This suggestion is supported by a study of creativity in artists carried out by Getzels and Csikszentmihalyi [4]. These authors identified ‘problem finding’ as an important pattern of behaviour in successful artists. This behaviour is typified by:

- spending time exploring alternative approaches to work before settling on one in

particular
- remaining ready to change directions when new approaches suggest themselves
- not viewing a work as fixed even when it is finished

Thus we may suggest that to enable creative thinking, representations used in early design should not only maximise fluency and discourage premature evaluation, but also encourage explorations and allow changes in direction.

In novel problem solving situations it may also be desirable to be able to introduce novel concepts. Furthermore, such concepts should be at an appropriate level for the task. One example of difficulties caused by working at an inappropriate level of abstraction is the problem of 'functional fixedness' which occurs when problem solvers have, perhaps unconsciously, introduced too many constraints. Looking for alternative ways of representing problems can help here. Glucksberg & Danks [5], for example, found that labelling of objects when solving problems does not always help, sometimes it hinders. Labelling an object as a screwdriver prevents us seeing it as a potential circuit connector. Using nonsense names apparently makes it significantly easier to see new uses for objects.

All of these arguments suggest that informal graphical representations or diagrams (doodles!) with no fixed semantics may have an important role to play in early design in enabling designers to think creatively about possible solutions to the design problems they are posed. However, from our earlier consideration of the context within which design often takes place (where there are inter-disciplinary design teams with changing membership), we can see that the representations used in design meetings may also need to fulfil other requirements: they may need to act as a medium for communication between different members of the design team, or even as a record of decisions made during design meetings.

The diagrams commonly used in meetings by designers at UH to help them to reason about particular system designs are often unstructured and do not use any one set of drawing conventions. The level of detail often varies in different parts of a diagram, depending on the difficulty (to the designers) of describing the corresponding parts of the system. Designers also introduce new diagrammatic conventions to describe problems encountered for the first time in new domains or with novel system architectures. In a close-knit design team, this fluidity of diagram semantics may not cause a problem, with the designers communicating effectively, even through changing and informal specifications. However, in larger design teams whose members have various levels of experience, misunderstandings may more easily arise. Backtracking from formal specifications or prototypes developed by individual members of the design team on the basis of such mistaken understandings is at best time-wasting, and can have more serious implications if customers have also been misled by the informal descriptions.

The study described in this section looks at the extent to which informal representations of a kind typically used in design meetings at UH may be expected to fulfil roles other than simply supporting creativity. It will focus on:

- the effectiveness with which they may be said to support an understanding of the system to be developed which is shared by the whole design team, and
- the extent to which they can be relied upon as accurate records of decisions made at particular design meetings.

2.2 Procedure

This study was based around a fairly difficult meeting of the SPIRE project design team at which 4 members of the team (J, S, M and F) were present. During the meeting the designers were trying to deal with a number of important but intangible design decisions to do with the conceptual design of the system. A number of diagrams were drawn.

After the meeting the diagrams were collected and the participants were interviewed later the same day. They answered questions about each of the diagrams. The diagrams shown were photocopied, although the originals were available to be viewed if necessary. Each participant was asked to:

1. Say who drew each diagram
2. Put the diagrams in order of their creation
3. Say what issue surrounded the creation of each diagram ("Why was this diagram drawn? What main issue does it relate to?")
4. Identify the parts of each diagram which were labelled on the photocopy ("What did this part of the diagram mean?")

Four diagrams had been produced during the original design meeting. These are referred to here as the 'layer' diagram, the 'local' diagram, the 'triangle' diagram, and the 'classes' diagram. Results relating to two of the diagrams (the 'layer' and the 'local' diagram - see fig.s 1a and 1b) will be discussed in detail.

Fig. 1a. The Layer Diagram (Showing labels 1 - 9)

Fig. 1b. The Local Diagram (Showing labels 1 and 2)

2.3 Results

There was broad agreement about who drew each of the diagrams and the order in which they were produced. A large number of statements were collected about the issues that surrounded the creation of the diagrams. For some of the diagrams there was considerable agreement amongst designers (including the creators of the diagram) about the issues they related to, but for other diagrams there was much less agreement.

The 'layer' diagram. All four designers said that the layer diagram (see fig. 1a) was constructed to help decide where class descriptions of data objects should be presented to the user: whether they should be placed in the advice layer -essentially a hypermedium - or in the data layer -a structured database that the user can search or browse. The designers' responses to question 3 (about the purpose of the diagram and the issues surrounding its creation) are given below:

J: "We were discussing whether class descriptions should be part of advice or part of the database"

S: "Having problems to do with what is going in advice layer and data layer. Particularly where class descriptions should go"

M: "Talking about class descriptions and where they fit into our three level hierarchy. There was some argument about whether it went in the bottom layer or the middle. User's and designer's models: F said does it matter form the user's point of view? J argued for the bottom layer, F argued for the middle layer"

F: "I drew this after the other triangular diagram. Its about the three layer model. Whether classes of data should be in the advice layer or the data layer. We decided this confused user and designer views"

This agreement between designers over the issues surrounding the layer diagram suggests that informal diagrams can be an effective aid to communication (and hence to the development of a shared understanding) in contexts such as those of interest in this paper.

Broad agreement was also shown in the identification task (step 4 of the procedure above) carried out in relation to the layer diagram as shown in the table below, although some designers (J and S) had a more detailed memory of the meaning of various diagram components than others.

Table 1 Features of the 'layer' diagram identified by each subject
('•' means the relevant component of the diagram was correctly identified)

	J	**S**	**M**	**F**
1 Advice Layer	•	•	•	
2 Data Layer	•	•	•	
3 Class Description	•	•	•	•
4 Entities / Data Organisation	•	•	•	•
5 Other Class	•			
6 User Access	•	•	•	•
7 Data / Class Link		•		
8 Advice Class Link		•		•
9 Advice Layer Boundary	•	•		

The 'local' diagram. The observations described above contrast with those relating to the local diagram (see fig. 1b). Here it was harder to identify any one shared view about the purpose of the diagram. The designers' statements about the issues to which the local diagram related are given below:

J: "J describing contents list like a help system. Subtopics for local information. We discarded this...or did we?"

S: "J responding to F's prompting about class description "local". She was proposing that people should access the data via a hierarchical contents list."

M: "To do with things having lots of parents. Instances relating to multiple class descriptors. Something to do with an implementation issue to do with whether we should keep lists of classes attached to objects."

F: "J drew this to show that we could just have a list of local equipment rather than links. I felt we were assuming that data was just equipment at this point."

Identification of the features labelled in the 'local' diagram was also somewhat inconsistent as shown below:

Table 2 Names given by each subject to features labelled in the 'local' diagram

	Feature 1	**Feature 2**
J	topic heading	sub topics
S	class description (?)	class descriptions; indentations show 'is a subclass' relationships
M	local class	lot of local objects/equipment
F		class members

2.4 Conclusions

From what we know about creativity and the conditions under which it thrives, we have surmised that informal graphical representations may have an important role to play in early design in enabling designers to think creatively about possible solutions to the design problems they are posed. However, this study has demonstrated that reliance on informal diagrams as the primary means of communicating and recording design decisions is likely to lead us into difficulties. The lack of agreement between designers about either the significance (in terms of related issues) or meaning of some of the diagrams (for example, the local diagram) suggests that informal representations such as those discussed here may not be a reliable means of developing a shared understanding of the system to be developed, and are not suitable for use as permanent records of decisions made.

One solution to problems such as these which is proposed by the software engineering community is to translate decisions made into more formal (or at least semi-formal) notations such as data flow diagrams or entity relationship models. It is often assumed that this translation would be carried out by a requirements engineer who would be present at design meetings and would then record what was decided outside of the meetings.

The aim of the second study reported in this paper was to investigate whether this was likely to be a useful and reliable way of supporting the design process.

3 Study 2: Use of semi-formal representations

3.1 Introduction

This study takes up the question described above; whether semi-formal representations generated by a requirements engineer, who has been present at a design meeting, may be used as a reliable record of design decisions made during the meeting. The aim of the study was to investigate the use of three different kinds of 'semi-formal' graphical representations in recording decisions about high level dialogue design. We wished to investigate whether each designer's memory for the design decision reached was the same as that of the others by comparing the way in which decisions were recorded in each of the notations (see step 3). We were also interested in whether agreement between designers would apparently be greater when decisions were recorded using one notation rather than another.

3.2 Procedure

This study was done following a further SPIRE project meeting at which a high level design for part of the SPIRE system dialogue was discussed. Four designers were present at the meeting (S, M, P and J). Note that S, M and J had been part of the previous study, but that F was replaced by P in this meeting. At the end of the meeting, each of the designers' own graphical representations of the dialogue agreed upon were collected. These had been drawn during and after discussions in the meeting which related to a fifth drawing on the white board.

In this study, each participant was approached on the day after the meeting and was asked to:

1. Describe in general terms what his/her own diagram was about.
2. Label all important parts of his/her own diagram.
3. Draw a representation of what was decided upon in the meeting using:

    - a Flowchart
    - a State Transition Network (STN)
    - a Jackson Structure Diagram (JSD)

    using the prompt sheets provided to find out what symbols to use for each notation, and what those symbols should mean. Drawings could be done in any order.

4. After each drawing, describe any problems with using the notation ("Was there anything you wanted to represent but couldn't using this notation?" "Did you have any difficulties with understanding how to use the notation?").

5. Note any further comments e.g. about how the notations compare.

(Note that the purpose of steps 1 and 2 was mainly to remind participants of what had been discussed in the meeting in preparation for attempting step 3.)

3.3 Results

No consistent patterns were observable in the orders in which designers chose to tackle drawing diagrams using the three notations requested, or in the times they took to produce those diagrams.

The characteristics of diagrams drawn using each of the notations were as follows.

Flowcharts. An analysis of the diagrams in terms of the occurrence of symbols given to the participants on the prompt sheets shows that they were used in the following way.

Table 3 Numbers of flowchart processes, conditions and arrows drawn by each designer

	Total Processes	**Total Conditions**	**Total Arrows**
S	7	2	13
M	7	6	18
P	7	5	15
J	8	2	10

Some processes (such as an initial Start or Enter process, and database and advice browsing processes) were identified by all designers. Others (such as Quit, Set Profile, and Browse Tasks) were only identified by some. The setting of user profiles was treated differently by different designers: for example, one (P) broke this down into three separate processes, while others treated profile setting as a single process; also two designers (P and M) indicated that the user could return to setting the profile from other points in the dialogue, whereas two (S and J) showed the setting of the user profile as a one-off activity at the beginning of the dialogue.

Table 4 Flowchart processes drawn by each designer
('•' means the process indicated was included in the relevant subject's flowchart)

	S	**M**	**P**	**J**
Start / Enter	•	•	•	•
Quit	•	•	•	
View Subject Index	•			

Access / Browse Database	•	•	•	•
Browse / Read Advice	•	•	•	•
Set Profile	•	•		•
Select User Type, Disability and Division				•
Set User Type			•	
Set Disability			•	
Set Division			•	
Set Default Profile	•	•		•
Browse Tasks		•		•
Refined View of Database				•

The comments recorded by the designers after producing flowcharts were as follows:

- Couldn't show details of 'setting profile' procedure. (S)
- Couldn't show that advice given depends on profile. (S)
- Ended up filling in bits of design I didn't feel were decided on in the meeting. (S)
- Not sure if I'm using flowchart syntax properly (horizontal arrows). (S)
- Hard to fit ideas into this sort of framework. (M)
- OK for this decision as it lent itself to simple Y/N questions, but I wouldn't use it for preference. (J)
- The flowchart was OK but suffers from the same drawbacks as JSD [see below] as well as being too procedural. (M)

State Transition Networks (STNs). An analysis of the STNs produced by the four designers reveals that the numbers of states specified by all designers was the same (though what these states varied considerably - see below), but that differing numbers of transitions were recorded.

Table 5 Numbers of STN states and transitions drawn by each designer

	Total States	**Total Transitions**
S	7	9
M	7	14
P	7	12
J	7	8

Again, some states (such as an initial Start or Enter state, and database and advice browsing states) were identified by all designers. Others (such as Quit, Access Mode Choice, Viewing the Subject Index, and Viewing Task Hierarchy) were only identified by some. The setting of user profiles was also treated differently by different designers: some gave a more detailed description of the states involved in this process than others.

Table 6 STN states drawn by each designer

	S	M	P	J
Start / Enter	•	•	•	•
Quit			•	
Access Mode Choice	•			
Viewing Subject Index	•			
Database Access / Browsing	•	•	•	•
Advice Browsing	•	•	•	•
Viewing Task Hierarchy	•	•		•
Profile Setting	•	•		•
User Profile Setting		•		•
Default Profile Setting		•		•
Set Student			•	
Set Disability			•	
Set Division			•	

The comments recorded by the designers after producing their STNs were as follows:

- Assume can quit system at any point. (S)
- States are always linked with what's on the screen. (S)
- Could put some information about transitions on arrows but not enough e.g. couldn't describe that advice displayed depends on profile set and task chosen very easily. (S)
- Filled in some bits not decided in meeting. (S)

- Not sure whether the bits circled in red are valid system states/transitions - possibly this is just one transition? (M)
- I keep wanting to put more interface detail in re: where do you go after Select Task or DB Access. This was not discussed at meeting and could be misleading if added. (J)
- Of the three notations, the STN was the most expressive for the ideas I wanted to capture. (M)

Fig. 2. STN produced by designer P for study 2

Jackson Structure Diagrams (JSDs). The following numbers of procedures were used:

Table 7 Numbers of JSD procedures drawn by each designer

	Total Procedures
S	19
M	12
P	9
J	10

Some procedures (such as the overall Using / Accessing Information in SPIRE, and Direct Database Access procedures) were again identified by all designers. Many more components were identified only by some designers. The setting of user profiles was again tackled differently by different designers.

Table 8 JSD procedures drawn by each designer

	S	M	P	J
Using / Accessing Information in SPIRE	• b	• b	• b	• b
Start	•			
Quit	• *			
Choose Access Route	•	•		
Direct Database Access	• ° b	• ° b	• ° •	• °
Guided Information	• ° b	• ° b		• ° b
View Subject Index	•			
View Database	• * • *	•• °		
Set Profile	• b	• * b	• b	• b
Select Task		• °		• *
View Advice	• *	• °		• *
User Sets Profile	• ° b	• °	•	• ° b
System Sets Profile	• ° b	• °	•	• ° b
Choose Profile				•
Default Profile				•
Choose User Type	• *		•	
Choose Disability	• *		•	
Choose Division	• *		•	
Set User = Student	•			
Set Disability = All	•			
Set Division = All	•			
Browse Data		•		

•• : procedure occurs twice in the diagram
b : procedure is broken down further
° : procedure is defined as optional
* : procedure is defined as iterative

The comments recorded by the designers after producing their JSDs were as follows:

- Can't show relation between two 'view info from db' boxes. (S)
- Can't show swapping between advice and data layer or subject index and data layer (but not sure this decided in the meeting anyway). (S)
- Can't show that advice given depends on profile. (S)
- Good to be able to break down functions. (S)
- Diagram got unwieldy. (S)
- Not sure if this is the right way to represent choices. (S)
- Doesn't seem very suitable for designing flexible dialogues. (S)
- In general I found it harder to do this than the other two methods. I think this was because in the meeting we talked about conceptual design rather than implementation and I have always seen JSD as more biased towards implementation issues. (M)
- Found this harder than flow chart. (J)
- I don't think it makes the selection decisions very obvious. (J)
- Wasn't sure where to put the detail of profile selection. (J)

- Numbering looks cumbersome. (J)
- How do I show select one or the other? (J)
- I found the JSD too low level for high level design ideas and too restrictive. (M)

3.4 Conclusions

A number of challenges were identified in relation to the post hoc recording of design decisions using semi-formal notations.

The same designer may record design decisions differently using different notations.

One example of this is the specification of a quit option. The possibility of quitting from the program under design was not actually discussed during the meeting, and no explicit decisions regarding quitting had been made in any previous meeting. Thus, the fact that J never specified how a user would quit from the program may reflect the fact that she was faithfully following the instructions of the study to specify only what she felt had been decided in the meeting. However, the other designers decided to incorporate a quit into their graphical specifications. Interestingly, each incorporated it (explicitly) in only some of their diagrams. While S included it explicitly in both the flowchart and JSD representations, she did not include it in the STN, but noted that she was assuming it would be possible to quit at any point. M included quit only in his flowchart representation, and P only in the flowchart and STN.

Table 9 Occurrences of the 'Quit' option in each designer's representations

	Flowchart	**STN**	**JSD**
S	•	Assumed from anywhere	•
M	•		
P	•	•	
J			

Another example of this problem is that in her flowchart representation, J specified that users should be able to move from one part of the system (where they were viewing advice) to another (where they would be able to see a refined view of the database). In her STN, she did not specify that this should be possible.

It is too early to say what features of the notation are significant here, though we might speculate that the sequential view of interactions which flowcharts and JSD force on the designer are more likely to lead to the specification of a quit option as the final procedure in a chain of interactions. Specifying the fact that the user should be able to quit from the system at any time is never easy in a graphical notation and quickly leads to diagrams becoming too complex to be helpful.

Different designers may use the same semi-formal notations in different ways.

With flowcharts, some designers used a condition to denote a point where the system asks the user for input and then acts according to the input received (e.g. J's condition: 'Do you want to?'). Others used a condition to describe something which the system does internally without the user being aware of it (e.g. S's condition: 'Profile set?').

In the State Transition Networks, there were differences in emphasis about what different designers thought should be represented as states. S's states corresponded to what might be on the screen at a particular point in the interaction. M's and J's were more task-related. Furthermore, some people put in states what others put in transitions. For example, S used a state to represent the point where users could choose between the two available access modes, whereas other designers simply specified two possible transitions from the previous state. Some people gave more detailed descriptions of transitions than others. An example from S's diagram is: 'Profile set: use user type to choose task hierarchy'. Examples from J's diagram are: 'Yes', 'No', 'Select DB', 'Select Guided Information'.

Three of the designers used the JSD notation in much the same way. The diagram produced by the fourth was much sketchier.

Designers who are not experienced or trained in the use of a semi-formal notation may use it incorrectly.

One of the participants in the study was confused over whether they had used the flowchart notation correctly, some designers did not label all transitions in the State Transition Networks, and three of the designers used the JSD notation incorrectly, despite having access to a prompt sheet which gave a general description of how to use it.

Some differences in designers' views and recollections of design decisions remain constant when decisions are recorded in different notations.

Sometimes, designers included in their diagrams things which were not discussed at the meeting but were assumed to follow from previous meetings. Not all designers agree on these issues. For example, S recorded the fact that users should access one part of the system (the database) in a particular way (via a subject index), but no other designer mentioned this. The use of a subject index to access the database had been discussed at a previous meeting but not explicitly decided upon. As a result of that meeting, S apparently assumed that it was to be used in the way specified, but other designers did not.

As another example, P had recorded a different and more detailed view of a particular part of the interaction (that in which a user's 'profile' is set) than other designers in each of his diagrams. It is interesting to note that the diagram drawn by P during the meeting was the only one to have been drawn 'as a discussion tool' rather than as a record of the design decision reached, so that his memory for what was decided might not have been so precise as that of other designers who recorded the decision in the meeting.

If diagrams using particular notations become too complex, designers may forget to put some things in.

For example, S recorded the fact that users would see a particular part of the system (the task hierarchy) in her flowchart and STN diagrams, but not in the JSD (which was drawn third). Her JSD representation had already become quite complex by the time she came to specifying the relevant part of the system - it practically filled the page already - and she may have forgotten to specify the use of the task hierarchy because of the size and apparent complexity of the diagram.

4 Discussion

The results of study 2 suggest that caution should be exercised in assigning the job of recording design decisions, or specifying design solutions, after design meetings to a single individual. For each of the semi-formal notations investigated, the decisions recorded by different designers after the meeting were different. Some of the differences in view were common across notations - these differences between the designers' memories for decisions reached were constant no matter what notation was used to record them. Other differences in view were apparent only when certain notations were used.

Each of the semi-formal notations used in the study appeared to have its own drawbacks. Comments about the difficulties experienced in using the three notations can be broadly categorised into three main areas:

- difficulties in expressing particular design decisions (different notations make different things difficult)
- use of semi-formal representations encouraging specification of more detailed design than was decided
- concern over diagram complexity

It seems likely that restrictions imposed by the notations used (and possibly also by the designers' lack of familiarity with the notations) could have lead designers to adapt and accommodate their memories of the design decisions made to the constraints of the notations, thereby tending to increase the differences between the decisions recorded by different designers.

The use of informal representations such as those considered in study 1 has the advantage that it does not impose any conceptual constraints on those drawing them. This is very useful, particularly during the early stages of a project where creative thinking is required. Such diagrams are commonly used in design meetings when new ideas are being discussed, and designers creating and viewing such representations can develop a reasonable degree of shared understanding, particularly in relation to informal representations which are relatively well-developed (such as the 'layer' diagram). However, understanding in relation to sketchier representations (such as the

'local' diagram) is less reliable. While the sense of agreement generated by the discussion of common artefacts during a meeting may be useful in terms of strengthening the team, it can have undesirable effects if it is, in fact, not based on proper understanding. It is worth noting here that the members of the design team who are likely to mis-understand each other most are those who do not share a great deal of experience - for example new comers to the design team (who will often be the programmers responsible for fine tuning the design!) are likely to mis-understand something which is agreed between senior members of the team on the basis of shared experience from previous projects.

One solution to the problems of relying on informal representations which is proposed by the software engineering community is to have requirements engineers translate decisions made into semi-formal notations outside of the meetings. This solution has its own problems. If no real agreement between designers has been generated during a meeting, then the view recorded by an individual requirements engineer may easily not be representative of all designers' views. The constraints imposed by the use of a semi-formal notation are likely to lead to further deviation from the views of those present in the initial meeting.

We suggest that designers should be encouraged to continue sketching out informal representations during design meetings held early in a project, and also during the early stages of later meetings, both for general discussion and for their own personal use. We have already discussed the way in which using informal representations in general discussion enables creativity and the development of a feeling of shared understanding. We also note that in study 2, where three of the designers (S, M and J) drew their own (informal) representations of what was being agreed during the course of the meeting whereas the fourth (P) did not, the semi-formal representations drawn by P after the meeting showed consistent differences from those drawn by S, M and J. Thus drawing even informal representations of decisions during a meeting appears to assist designers in appreciating and accurately remembering what is discussed.

In addition to using such informal representations, we suggest that as the project progresses, it is worthwhile to focus discussions by attempting to generate semi-formal representations during the later stages of a meeting with all designers present. This facilitates the development of a more detailed, but still shared, understanding of the design solutions proposed, and means that the representations produced serve as a reliable record of what was actually agreed during the meeting. Semi-formal representations can be used as a basis for discussion, negotiation, and specification of agreement during the meeting - effectively playing the role of a joint contract agreed by all designers present. Diagrams drawn using semi-formal notations can be composed by one or more of the designers present in a public way and taking account of the views of all designers present - conflicts being publicly resolved as necessary. We suggest that this might be done using shared physical or computer-based (CASE) tools which support the creation of appropriate diagrams - for example allowing designers to easily manipulate symbols used in notations appropriate for describing the relevant design decisions. We aim to continue our research in this area in order to investigate the feasibility of this approach.

References

1. Bearne, M., Hewitt, J., Jones, S. and Sapsford-Francis, J. "Providing public access to information in complex and weakly structured domains: a 3-layered model for hypermedia information systems", *New Review of Hypermedia and Multimedia, Applications and Research*, vol 2 (1996).

2. Bransford, J.D, Sherwood, R.D., and Sturdevant, T. "Teaching Thinking and Problem Solving" in Baron J.B. and Sternberg R.J. (Ed.s) *Teaching Thinking Skills: Theory and Practice*. Freeman (1987).

3. Britton, C. Jones, S. Myers, M. and Sharif, M. "A survey of current practice in the development of multimedia systems", *Information and Software Technology*, vol 39, 695-705 (1997).

4. Getzels J. and Csikszentmihalyi M. *The creative vision: A longitudinal study of problem finding in art*. New York, Wiley (1976).

5. Glucksberg, S. and Danks "Effects of discriminative labels and of nonsense labels upon availability of novel function" *Journal of Verbal Learning and Verbal Behaviour*, vol 7, 72-6 (1968).

6. Meadow, A., Parnes, SJ. and Reese, H. "Influence of brainstorming instruction and problem sequence on a creative problem solving test" *Journal of Applied Psychology* , vol 43, 413-416 (1959).

7. Osborn A.F. *Applied Imagination*. New York, Scribner (1953).