Lecture 7: User-centred design research

Observation and task analysis

There are many techniques:

Structured interviews

Most software projects start with a series of meetings in which the system requirements are established. The agenda of these meetings is often concerned with many other matters than the user interface, however. In fact the people who will use the completed system may not even be present. Their requirements are defined by a representative (a system analyst for an internal project or a market researcher for a product) who may not have much experience of design for usability.

For this reason, user interface designers often conduct studies specifically to discover the requirements of the system users. One of the cheapest and most straightforward techniques is to conduct ***interviews*** with the users. Interviews must be carefully planned to be effective, however. They are generally more or less ***structured***, encompassing a selected range of users, and taking care to encourage cooperation from users who may feel threatened or anxious.

A structured interview is based around a set of questions that will be asked of every interviewee. This need not necessarily be a long list, but it helps to collect data into a common framework, and to ensure that important aspects of the system are not neglected.

Observational studies

Observational studies are a less intrusive way of capturing data about users' tasks, and can also be more objective. They involve more intensive work, however. An observational study of tasks that take place in a fixed location can be conducted by making video recordings which are transcribed into a ***video protocol***. This protocol can then be used for detailed analysis of the task - relative amounts of time spent in different sub-tasks, common transitions between different sub-tasks, interruptions of tasks and so on.

Audio recordings can also be used for this purpose in certain domains, but these are less likely to be useful for task analysis than they are in think-aloud studies.

If a task ranges over a number of locations, the investigator has no choice but to follow the

subject, taking notes or recordings as best as possible. An alternative is the user of diary studies, in which subjects take their own notes, but prompted to pay attention to specific times, events or categories.

Ethnographic field studies

Ethnographic study methods recognize that the investigator will have to interact directly with the subject, but while taking sufficient care to gain reasonably complete and objective information. An ethnographic study will attempt to observe subjects in a range of contexts, over a substantial period of time, and making a full record using any possible means (photography, video and sound recording as well as note-taking) of both activities and the artefacts that the subject interacts with.

Cognitive descriptions of human performance (often called ***human factors*** by engineers) tend to be most valuable in detailed assessment and critique of a proposed design. Descriptions of mental models can be helpful in elaborating a design concept. But ethnographic observation can help to understand technology and products in completely new ways, perhaps leading to innovative new concepts. In this respect, ethnography can be considered as a contribution to engineering ***requirements capture*** in a traditional technology company.

HCI researchers tend to have skills in all these techniques, but product designers generally want a simpler recipe that doesn't require them to spend a year or more doing fieldwork. Often the biggest problem they have is how to gain a perspective of what it is like to be a user, escaping the mindset of their own technical understanding and expectations of the product. A useful intermediate technique is to write fictionalized descriptions of the kind of person who will use the product, to help the engineer understand what sort of person they are based on his or her personal experience. These user ***personas*** might be derived from ethnographic fieldwork, or from conventional market research data. They are a particularly popular technique in Microsoft, where persona descriptions include photographs (presumably of actors), fictional biographies, and descriptions of why this person uses computers. Product design then proceeds on the basis that the designer tries to accommodate this range of fictional people.

Prototyping

Prototyping is becoming increasingly important as a software design method, particularly addressing the problems of developing user interfaces within a strict ***waterfall*** development model. Companies that use waterfall models have placed increasing emphasis on accurate portrayal of the user interface at the specification phase, after finding that the majority of specification changes arise from client not understanding the requirements for user functionality. In terms of mental models theory, this could be expected - clients who have no image of the interface that they will operate are unlikely to have a useful mental model of system behaviour.

If the system can respond in complex ways, it is difficult to appreciate this from static figures in a specification, so the specification phase of projects often uses ***rapid prototyping*** tools to construct a functional user interface. This prototype can be demonstrated to clients and used as a basis for discussion. If a spiral development model is adopted rather than a waterfall, the prototype can be ***refined iteratively*** until the full system functionality is achieved. Incremental prototyping requires that the rapid prototyping tool also meets the engineering requirements of the final system. If such a tool is not available, an alternative is ***deep prototyping***, in which one aspect of the system functionality is fully implemented before developing the rest of the interface.

These common approaches to prototyping are quite different to the prototyping techniques that have been found to be successful in developing novel user interfaces. Many product designers believe that creativity in the product design process is directly related to the number of prototypes produced. HCI research similarly emphasises techniques for developing a large number of prototypes, exploring different possible solutions, and evaluating the usability of alternatives. This is in contrast to incremental prototyping techniques, which encourage cost-saving by using the first solution regardless of its usability properties.

Investigation of multiple prototypes requires low cost techniques for producing prototypes. Rather than implementing realistic system functionality, these often use generic graphic design tools with some scripting functions: early HCI research often used Apple Hypercard,

and more recent work uses tools like Flash. Simulations of user interfaces are often presented as a ***click-through*** prototype – a sequence of simulated screens typically loaded into Powerpoint, where a demonstrator moves the mouse pointer to a position on the screen and clicks there, pretending that the system is responding to that action (in fact, it always advances to the next slide – the demonstrator has to remember where to click to make it appear realistic). An even more radical proposal is ***low-fidelity*** prototyping, in which the prototype user interface is made using controls built from glue and paper.

The objective of building multiple prototypes is to investigate design alternatives through evaluation with actual users. This might involve simple discussion in a participatory design workshop, a more structured interview, or use of think-aloud to study the mental model that the user develops when interpreting the prototype.

Theories of computers and social actors

When we are designing computer systems that will be closely integrated into a social context, we need to consider a number of important properties of social contexts. The theoretical concerns are, as follows:

1. The things that people say relate to the current situation (***indexicality***). In natural human conversation, it is not usually possible to interpret an utterance without knowing the context in which it appeared. Social situations are constructed from words, but the same word can be used to make very different situations.

2. Most social actions are made in response to the other people around you and the things they do (***contingency***). It is seldom practical to make a detailed plan in advance, and then carry out an exact sequence of actions without change. But plans are necessary and useful; so a plan must be flexible. Alternatively, a plan might be an orientation rather than a specification of 'what to do'. Indeed, most plans are in fact of this latter kind.

3. In social situations, the things that people say and do are organized in a way that makes sense to other people (***accountability***). If someone does something that does not make sense within this framework, this makes him appear irrational, disruptive or mad (whether or not his actions seemed logical in his own mind).

4. If you want to understand why someone said or did something, it is the context they are

in that gives their actions meaning; their point if you will. Abstract theories or explanations about social actions are often too abstract to help explain the details of social actions (and so may not help in design). Nevertheless, and perplexingly, abstract theories are often used by people themselves to help give meaning and sense to what they do in any situation.

All four of these aspects of human behaviour are different to the behaviour of computer systems. The semantics of computer languages determines that a particular language element should always have the same effect (they are not indexical). Computer plans can be derived exactly from a statement of requirements, but it is seldom possible for a computer to improvise, other than replanning from scratch (they are not contingent). Computers do not behave in the same way as people, participating in social situations, with the result that they often do things that seem arbitrary, rather than accountable contributions to a conversation. Finally, computers always act according to rules (programs) constructed to follow computational theories – the computer itself is not uniquely adequate to explain its behaviour.

If we are designing computer systems that will be used within social situations, we need a set of research methods that can provide a 'remedial' perspective that makes the computer slightly less incompetent as a social actor. We can do this by observing real social actions in context (***ethnography***), writing about what we see at a level that not only what happened but why things were done, in a way recognizable to the participants themselves (***thick description***), and making detailed analysis of recordings in order to understand the patterns and rules of the situation (***work analysis***).

Methods for analyzing qualitative data

There are some rigorous approaches to analysis of video data, oriented toward these concerns. However, these require specialist training and are probably too time consuming for routine design work. In cases where it is necessary to take an open-minded approach to qualitative data such as interview transcripts or think-aloud protocols, many HCI researchers use a technique known as ***grounded theory***, in which individual statements are coded and categorized in response to the data itself, rather than framed by a prior hypothesis or assumptions about how users ought to interact.