Constructing and evaluating social software: lessons from Interaction Design

Christopher Douce
Institute of Educational Technology
Open University, Milton Keynes, MK7 6AA, UK

ABSTRACT
The process of developing interactive systems necessitates designers to have a comprehensive understanding of the needs of the user and the context in which a device or system is to be used. Interactive systems are often designed through a series of iterations, guided by a sequence of evaluations. This chapter describes how the research and development techniques used within the field of Interaction Design (ID), a successor to the field of human-computer interaction, can be used to inform the development and evaluation of social software systems. Particular attention is given towards the challenging area of end-user culture and how different evaluation paradigms and techniques can be applied. The chapter concludes by presenting pointers towards a number of international standards and highlighting a number of potentially useful research directions.

INTRODUCTION
Social software systems are interactive systems that enable users to share information and collaborate with each other. This chapter presents a very brief overview of a discipline called Interaction Design (ID). Interaction Design is a development of human-computer interaction. It represents a set of tools and techniques that can be used to guide the development of social software and interactive systems. ID draws upon a number of different fields, ranging from computer science and software engineering, through to psychology, sociology, anthropology and design, each contributing useful theories and knowledge that to guide the design and development of interactive devices, systems and environments.

This chapter begins with a description of what ID is and is then followed with a presentation of some of the processes and techniques that can be used to explore product and user requirements, drawing upon the powerful notions of user profiles and task scenarios. The 'culture' and 'understanding the user audience' sections provide additional source of useful guidance. The culture section asks designers to consider differences between user groups. The 'understanding the user audience' section calls upon designers to consider the importance of accessibility and inclusion. The design section that follows explores the process of creating a prototype and points to other important design approaches, such as participatory design.

One of the central tenets of effective ID and user-centred design is the notion of evaluation. Evaluating a system, whether it be a low fidelity paper prototype or the first release of an interactive system enables designers to explore whether a system works in the way that we expect it to and ask whether there is anything else that needs to be done to make the system more suitable for its target audience. The evaluation section outlines a number of useful techniques that can be applied by the designers of social software systems. A section that describes a number of technical standards offers useful guidance to facilitate the development of usable and accessible social software systems can be found towards the end

of this chapter. This is followed by a very brief example, taken from a current Interaction Design course that illustrates how some of the ideas can be applied. The chapter concludes by outlining a number of current and future research directions before summarising the main themes.

BACKGROUND

Interaction Design can be described as a set of subjects and techniques that can be used to facilitate the design of 'interactive products to support people in their everyday and working lives' (Preece et. al., 2002). Interaction Design has emerged from the field of human-computer interaction (HCI). HCI has a long history which encompasses diverse domains such as engineering psychology (Wickens, 1992), human visual and auditory perception, product design and typography. It could also be said to embody a number of different conceptual approaches; an approach that seeks to explain how aspects of interfaces are perceived by users, and another approach that considers the processes that can be used to design effective interfaces.

The emergence of HCI as a subject in its own right has been facilitated by the development of high resolution computer graphics, the creation of new ways to interact with computer systems and increasing levels of computing power. The continual increase in computing power has inspired the creation of new ways in which desktop computer systems can be used. The challenge of HCI was to explore not only how to harness new levels of power but also to create systems that are understandable and comprehensible to different user groups and communities without users having to undergo extended periods of training. User interfaces, in essence, should match the experience and needs of those who need to use them.

As computing and digital communication technologies have evolved, the computer is no longer a tool that is constrained to the desktop. Instead, the notion of an interactive product, or more specifically interactive software system, has expanded considerably due to the emergence of innovations such as mobile phones, personal digital assistants and the ability to embed microcomputer technology into a myriad of different devices. An interactive system can now be thought of as an application found on a desktop computer or an application running on mobile devices. The term can also encompass systems found within public spaces, such as bank teller machines, travel ticket dispensers and so on.

Interaction Design places significant emphasis on the user, drawing upon the notion of user-centered design. The user, it is reasoned, is likely to be the expert of their own needs. Approaching design with the objective of solving a functional problem without understanding the user may yield a product or system that may 'work' in a functional sense but may not be suitable for its intended users. A functional product may provide appropriate information from a set of inputs, but may be unusable in the sense that a user may not have sufficient background knowledge to understand which inputs to provide, or background knowledge to understand the outputs a system may produce. The needs of the user and the challenge of solving a problem through the creation of an interactive system need to be considered simultaneously.

Figure 1, from Preece, Rogers & Sharp (2002, p.186) represents an interaction design process. One of the most powerful principles of the ID process is the important notion of iteration; continual evaluation and re-evaluation has the potential to yield increasingly detailed designs that become increasingly suited to an intended user audience.

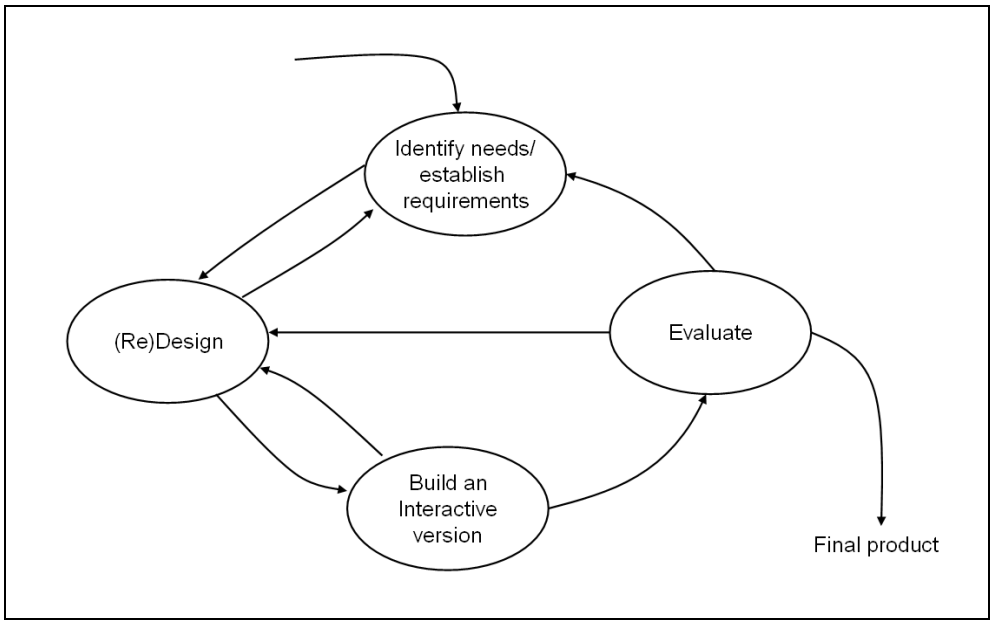

Figure 1. Interaction Design cycle

The phrase *social software system* is a term that can refer to a wide variety of different tools, ranging from instant messaging products and on-line discussion tools, through to web based utilities that enable virtual communities to be established. They can be defined as interactive systems that enable people to exchange information, potentially with to the objective of carrying out or supporting the completion of one of more activities or tasks. The precise nature of the information that is shared and the types of activities that are executed depends upon the users and the environment in which the systems are deployed. Different tools can be applied in different situations and social contexts. The requirements for users who wish to organise face to face meetings to support social activities, such as organizing a meal for friends at a local restaurant will be substantially different from users who use social software as a part of a well defined work task, for instance.

Ideally, interactive systems, whether they are social software systems used by many users or solitary applications used by a single user, should be considered 'usable'. The notion of usability differs between users due to variations in personal experience, knowledge, expectations and the physical environment in which interaction takes place. To try to understand or conceptualise usability, there are two sets of goals: usability goals and user experience goals. Usability goals relate to *functional attributes* of a system, such as whether a system is effective, efficient, safe to use, easily learnable and memorable. User experience goals, on the other hand, refer to different *attitudes and experiences* that a user may gather from an interactive system or device. User experience goals aim to consider whether a system is rewarding, fun to use, aesthetically pleasing, entertaining, enjoyable and so on (Preece, Rogers & Sharp, 2002). The challenge that interaction designers face is to find out which of these goals are appropriate given what is known about users and the tasks that they are likely to perform. Designers must then select and apply different tools and techniques to uncover potential designs whilst bearing in mind their chosen goals. When a design is complete, perhaps through the creation of a prototype, designers must then step towards choosing an evaluation method to uncover whether a design supports the chosen goals.

These challenges are addressed as designers move through an interaction design cycle that is similar to the one depicted in Figure 1. The following sections in this chapter expand upon some of the themes that the interaction design cycle encompasses, introducing the designers of social software systems to a number of tools that can be used to learn about the user, their tasks and the environment in which they are situated.

INTERACTION DESIGN THEMES

Gathering Requirements

Interactive products have users. Users differ in terms of their background knowledge and the attitudes that they hold towards a system, a product or problem. Users also differ in terms of the kinds of operations that they wish to perform using a system. Let us consider an example of a vending machine used to dispense train, metro or bus tickets. The vending machine is placed in a public space, perhaps in the concourse of a train station. One user, for example, may wish to purchase a ticket between two locations, whereas another user may wish to obtain some general information about a range of travel options that may exist. Different users may have different reasons to use a system.

Interaction designers can use a variety of tools to uncover assumptions about how a system should work, who it is designed for, and the tasks it should support. The differences between users of a system can be explored through the creation of sets of user profiles, or *persona*. A user persona is a rich description of a potential user. A persona enables the designer to consider the age and gender of a user, the motivations that they may hold, and be used to articulate the knowledge of similar systems that they are expected to possess. Persona descriptions are often named, enabling designers to share references to groups of user attributes quickly and efficiently. They enable hidden assumptions about potential users to be made explicit and invalid or erroneous ideas relating to user identity to be challenged. Designers are encouraged to create groups of personas to enable the boundaries between different users to be considered. One useful approach might be to create 'extreme' users, whose characteristics, values and objectives may be different to the norm (Moggridge, 2006), thus avoiding the possibility of creating systems that only cater for the 'average' user.

User personas can be used in conjunction with task scenarios. A scenario can be considered to be a rich description of an imagined situation, in which a product or system can be used by a particular user or group of users. A scenario, like the persona, allows implicit assumptions to be articulated and issues that need to be considered within the design phase to be highlighted. A scenario can be used to explore the environment in which a product or system is being used, for instance. It can also be used to consider the type of device or interactive mechanism that may be used in a particular product. A named user, for instance, may use a messaging tool to talk with a friend during a lunch break. Alternatively, the same user may use the same application through a mobile device whilst travelling home during the evening to communicate with a work colleague to confirm a meeting time that is to take place the following day. Scenarios enable the various contexts of use to be exposed and considered. Similarly, personas enable requirements and design challenges that relate to the user to be discussed and shared.

Scenarios and profiles are informal descriptions. To further understand the detail of a potential system, it may be useful to convert scenarios into a series of descriptive use cases. A use case can be described as a formal summary of the activities a system may perform, with emphasis placed on the sequence of tasks. There are a number of different types of use cases. Formal modeling languages, such as the Unified Modeling Language (UML) suggest a graphical notation. Other approaches include describing sequences of operations in a textual form, where alternative actions or deviations from the main task flow can be presented as alternative cases. The creation of use cases also permits the formation of higher level 'abstract' use cases, called 'essential use cases' (Preece et. al., 2002). These use cases allow designers to consider the pure essence of a series of interactions, devoid of any kind of implementation detail. One advantage of such an approach is that it has the potential to allow designers to consider different forms of interaction mechanisms through different types of devices or technologies.

An important question is this: how do designers find out about the users and the tasks that a system is required to support? A number of different techniques can be applied. These can include interviewing different stakeholders, observing how current systems are used, conducting surveys and questionnaires to identify potential areas for improvement, looking at data that is used and generated by existing systems and examination of associated documentation.

The development of requirements should be considered to be an iterative process, as suggested in Figure 1. Once a set of personas, scenarios and use cases are established, it is the responsibility of the designer to consider whether or not they accurately reflect the required essence of a system. To test their assumptions, designers have to design and build simulated or real artifacts that can be used to elicit opinions about the validity of the embodied requirements (as represented by a prototype) from a range of potential users.

Culture

In Interaction Design, the knowledge, experience and levels of motivation of different users is undeniably important when considering the requirements of an interactive system. Another attribute that should be considered is the culture of the user. Culture is a difficult term to define. In some respects, it can be used to refer to national culture, but it can also be considered in terms of other dimensions, such as profession, gender or socio-economic group. Some researchers have posited that culture can influence the design of an interactive system, its interface and the functions it may offer.

The work of Hofstede (1994) is considered to be influential. Hofstede posited a range of *cultural dimensions* that have been drawn from surveys distributed to employees of an international organisation, IBM. Hofstede suggested that, broadly speaking, citizens of different countries differ in their views towards acceptance of power, societal equality and tolerance of change. Power, for instance, is considered in terms of a cultural dimension named Power Distance. In a country with a high power distance score, power is held by a select few, whereas with a country with a low score, power is more equally distributed throughout a society. Similarly, Hofstede proposes a dimension called Uncertainty Avoidance that relates to whether change is likely to be embraced. Marcus and Gould (2000) describe how the cultural dimensions may influence the design of different web sites. Interactive products used in a country with a high power distance rating may present graphical symbols to which power can be attributed. Products used in a culture which has a low uncertainty avoidance rating may encourage users to explore different routes through a system and uncover new ways of working.

Hofstede proposed that different countries differ in terms of his dimensions. It should be stated that his dimensions remain controversial. Firstly, they represent broad generalisations and do not take into account that individuals and subgroups within a country may hold values that are at odds with a cultural norm. Secondly, the research that has been carried out comes from a particular context: a single international organisation and this may influence how the different dimensions have been conceptualized. Whether designers accept the validity of the Hofstede dimensions (or alternative dimensions proposed by other researchers) is perhaps a moot point. Rather than guiding the design and construction of interactive systems, they represent a set of conceptual tools that enables designers to consider the issue of culture and how it, as a concept, can influence the design of a product.

Culture is considered to be an important issue regarding the design of social systems, and one that requires further exploration. It is possible that a single social software system may be available to many different user groups, each of which may have differing values. The way in which the use of these systems can be utilised may be acceptable and desirable for one group but perhaps frustrating and discouraging for another.

Considering User Diversity

The requirements and design phases of ID should also take into consideration the functional diversity of a user population. Not only can users be considered in terms of their culture, background knowledge and experience of interactive systems, can also in terms of how different people can interact with systems. A system that is *accessible* is one that can be used by different people, regardless of whether they have any sensory, physical or cognitive impairments. People who have visual impairments, such as reduced vision or blindness, can overcome their impairments through the application of assistive technologies, such as screen magnifiers or screen readers.

Whether or not a system will work well with an assistive technology depends upon a number of factors. The first is whether an assistive technology is available on a system or device, whether or not a user is sufficiently skilled or able to make use of the tools that are available, and whether an interactive product has been designed in such a way that enables it to be used with different types of input and output mechanisms. Screen reader software, for instance, converts a visual representation of an application or web page to a linear audio stream, allowing people with visual impairments to listen to what is presented on a screen. Difficulties may arise if an application dynamically downloads and updates on-screen material without informing the user. Users of screen readers may therefore become disoriented or miss essential information. It is therefore essential that developers consider the diversity of not only the user audience, but also the diversity of the interface approaches that users may choose to apply.

Users of screen reader software can most effectively use web pages that use 'semantic markup'. If a web page or interaction screen has a clearly defined semantic structure, in terms of headings, visually impaired users, through the use of assistive technologies, will be more able to effectively navigate through the structure of an application, tool or web page. Designers and developers, therefore, should take effective steps to separate the presentation of content from the mechanisms that are used to present the material. Developers, for instance, should design web pages in such a way that their display can be controlled through cascading style sheets (CSS) rather than embedding presentation and formatting information within the body of a page, for instance. Enforcing this separation enables people who use assistive technologies to more efficiently navigate through interfaces and increase the likelihood that search engines can more effectively index and extract information from pages that are well structured.

The subject of web accessibility, and therefore the accessibility of social software systems, is a subject of continued debate and development. International standards bodies, such as the W3C, (Craig et. al., 2009) are considering the most effective approaches to take account of the dynamic nature of Web 2.0 applications whilst ensuring that new innovations in software development and interactive web system design do not create barriers for people who have physical, sensory or cognitive impairments.

The accessibility of interactive systems should *never* be considered as an afterthought. Instead, it should be considered as an integral part of the requirements process. Different users of a system need to be considered from the outset. Excluding certain user groups through poor design is not only undesirable from a moral perspective. The right to accessible applications is now considered as a part of equal opportunity legislation and those organisations that do not take account of user diversity will risk prosecution. Another argument for providing equality of access is an economic one. Excluding users, however accidental, will also exclude opportunities to provide goods and services. Furthermore, research has indicated that systems that are accessible may also achieve higher levels of usability. Increasing accessibility has the potential to benefit all users, regardless of how they interact or consume a social software service (Petrie & Kheir, 2007).

Design

The activity of design broadly refers to the process of constructing or creating an interactive system. In the context of Interaction Design, *design* can also refer to the process of software development. Before any detailed technical development can begin, designers should acquire a firm understanding of the processes users can adopt to request information, and how the information is presented to them. One way to achieve this understanding is to build a number of different prototypes (Lim, Stolderman & Tenenberg, 2008).

Prototypes can be divided into two broad categories: low fidelity and high fidelity. A low fidelity prototype is a 'low tech' presentation of the key elements of a system and can take the form of a 'paper prototype' which can consist of a number of rough pencil sketches. These include a storyboard, a series of task cards, and an interface sketch. A storyboard is used to present a graphical description of how a product is to be used. Users can be represented as simple stick figures and interactive devices can be drawn as crude blocks or squares. Important design influences, such as the weather (which may point towards physical design considerations) or crowds of other people (which may potentially point towards issues of security) can be represented, and can guide the development of a product.

Card sketches allow designers to quickly explore the sequence of actions that are carried out through an interface. A single card can be used to represent the information that is presented through an interface. Cards can be used to explore the question of whether the right information is presented on the screen, and whether a task should be split over a number of displays or pages. An interface sketch represents a development of a card sketch. It allows a designer to create a more elaborate representation of an interface screen, enabling the designer to begin to envisage the look and feel of an emerging design.

To some, the notion of sketching may appear to be somewhat simplistic and unable to convey the true richness of digital interactive systems. Low fidelity prototypes represent the starting point in the development of a system. Sketches have the advantage of being quick and easy to create. Creating higher fidelity prototypes, perhaps by drawing designs in a graphical design tool, requires designers to commit to greater levels of time investment. Since sketches can be drawn quickly, there can be a lower sense of investment in the end result, and a greater inclination to change designs, should they be considered to be unsatisfactory. Furthermore, design sketches can be easily shared between designers, shown in communal spaces, cut out and torn apart and combined with sketches produced by others.

High fidelity prototypes can range from demonstration systems created using screen design tools, where sets of graphical representations of screen mock-ups can be imported into presentation packages, such as Microsoft PowerPoint. Alternatively, high fidelity prototypes or may be operational systems developed tools such as Visual Basic. This type of prototype can differ wildly in terms of how much functionality they implement. A horizontal prototype, for instance, may implement most of the key items of functionality at a relatively superficial level, presenting the salient menu options, for example. A vertical prototype, for instance, may demonstrate, in detail, how a small number of functions operate.

By applying the principle of iteration rigorously it is possible to incrementally develop a prototype into a fully functional product. A low fidelity sketch could be translated into a design that may guide the development of a high fidelity equivalent. The skill of the interaction designer does not only lie with the ability to understand and imagine the user and their tasks, but also their ability to construct useful and effective evaluation activities. An interaction designer must also examine the results of evaluations and listen to potential users and make decisions about what feedback will most effectively guide the design and development of the next version.

Significant questions that need to be asked include: other than following a process of iteration, what occurs within design? Secondly, what can be done to facilitate the generation of new ideas that may find their way into an interactive system or social software system. In his book Designing Interactions, Moggeridge (2006, p.729-732) presents an alternative list of stages (or principles) that may comprise a design process. Each of these principles outlined in table 1 could be used in sequence, but it is entirely possible that each stage or activity yields new information or insights that takes the designer towards parallel periods of discovery and exploration. Moggeridge writes, 'the fastest progress towards a successful design will be made when these elements are used quickly and repeated frequently, but not usually in the same order!'

Principle	**Description**
Constraints	This relates to the starting point of the product or the system. It may emerge in the form of a briefing or specification, either from an individual or a design team.
Synthesis	This is the phase where different ideas collide to produce new ideas after 'all of the relevant issues have been absorbed'. Synthesis comes when a design problem is understood in its entirety, and connections between different themes and issues can be created to form new ones.
Framing	Framing relates to the concept of adding a framework to the activity of thinking about a problem. By framing a problem (or describing it in a particular way) new ideas can be considered.
Ideation	Ideation is the process of creating new ideas. One approach is to hold brainstorming sessions with designers who each have an understanding of the problem. Ideas and related concepts can be sketched or drawn. One approach is to continue to create new ideas and to defer judgement on all ideas until the end of an ideation session.
Envisioning	'Ideas are like dreams until they are visualized into some concrete representation' (Moggeridge, 2006, p.733). The concept of envisioning echoes with the principle of storyboarding, to explore context and consider individual users and their actions through user profile and scenario descriptions.
Uncertainty	Some ideas that are produced may not be clear. After ideation, many ideas may be uncovered but it may be unclear which deserve further attention or consideration. A sense of uncertainty leads to the selection of those which may be worth exploring further.
Selection	This is the phase of culling those ideas that are not considered to be 'good enough' to warrant further exploration.
Visualization	This phase simply consists of taking the idea and drawing sketches. A visualization in this sense can be qualitatively different to the categories of sketches that were described earlier and make up the prototype phase.
Prototyping	'Prototyping is about testing any aspect of the way a design is expected to work. You can create a prototype that represents an idea that has been selected and visualized' p.734. In this phase you can create (low-fidelity) prototypes using index cards, scissors, glue, pencils and pens in a way that is much quicker (and cheaper) than had designs be created using a computer mediated application. Plus, different designers can be directly involved without tools getting in the way.

Evaluation	This may include taking your sketches and presenting them to users or other designers. These sketches may inspire the designer to consider new alternatives. The feedback that is gathered may point the way towards modifications to a prototype, or reapplication of any of the previously described phases.

Table 1. Design phases.

Design is an exceptionally creative activity. It relies on the ability of designers to absorb requirements, consider the perspective of others and imagine the circumstances in which products or tools may be used. Designers are also free to use analogies or metaphors to think about and consider designs. Not only do metaphors have the potential to enable potential users to draw upon existing expertise, helping them to understand unfamiliar interactive products, metaphors also have the potential to point designers towards new ways of interacting with a system.

Preece et. al. (2002) introduce the notion of a *conceptual model*. A conceptual model is a 'set of integrated ideas and concepts about what it [a product] should do, behave and look like', p.40. A related concept is the notion of an *interaction mode*. This is the way in which interaction with a system or product may take place. Broadly speaking, an interaction may relate to actions carried to an object (such as a file located within a desktop metaphor), or by carrying out a process or activity. Activity based interactions may include instructing a system to do something (issuing it a command), conversing with it in some way (in the form of an interactive dialogue), manipulating and navigating items within a part of an activity, or exploring and browsing aspects of a system or environment. Metaphors, analogies, conceptual models and interaction modes all represent useful and important concepts that allow designers to consider how to create interactive systems.

Another approach that could be used to facilitate design is to ask users to become directly involved in the design process. Designers may not have all the knowledge, insight and understanding about how to specify a product for a particular user audience. The principle of adopting co-designers, or potential users who are unskilled in the process of design is known as *participatory design*. This approach has the advantage that it is possible to establish a direct dialogue between the user and the designer. Issues such as terminology and functionality can be uncovered and discussed. One important disadvantage is that there may be professional cultural differences between the designer and the co-designer; the cognitive distance between the knowledge of the designer and the user has the potential to cause communication difficulties and yield misunderstandings. Furthermore, it should always be remembered that the co-designers may be tempted to propose designs that are appropriate for users who are similar to their own, potentially disregarding other equally important user groups. It is important that the interaction designer attempts to maintain a degree of objectivity whilst at the same time trying to effectively harness the insights and knowledge the co-designers may be able to offer.

Evaluation

With iteration being a fundamental principle of ID, the activity of evaluation is one of its foundations. An evaluation aims to determine if a system that is being proposed is going to be useful, understandable or appropriate for a certain group of users. Evaluations can be initiated throughout the ID cycle. User feedback can be solicited from rough sketches or high fidelity prototypes to gather information about the effectiveness of an emerging design. An interaction designer may create several versions of a design and carry out a number of different evaluations.

Since iteration is considered to be both necessary and useful, designers can legitimately ask the question: 'how many alternative designs or prototypes are necessary to build an effective system?' The ultimate

answer depends on the product that is being constructed and how much time or budget is available. Different evaluation techniques differ significantly in terms of how much time and resources they require. It is important to remember that the cost of evaluating prototypes and making changes early within a design cycle is likely to be slight in comparison to the cost of making substantial changes to high fidelity prototype or operational system.

Returning to the principle of a 'card prototype', a designer can solicit feedback relating to an early design by 'simulating' the actions of an imagined computer. Mouse clicks or button presses can be simulated by presenting questions to evaluation participants akin to, 'what would you do now?'. When a user has performed an action, the evaluator could then change one interface card for another. Any confusing terms or difficulties with the interaction can then be noted and used as input to a further design iteration.

When faced with a high fidelity prototype or an existing system, perhaps in the form of a web site it is possible to apply a *discount usability* method, such as heuristic evaluation (Nielsen & Molich, 1990). Using a number of well known usability guidelines, a small number of experts can be called upon to assess the operation of a system to determine whether improvements can be made. The advantage to such an approach is that it can provide relatively quick (and low cost) feedback about a proposed design. The disadvantages include the possibility that evaluators may discover faults that are not present, misunderstand the context in which the system is used, or fail to take account of the attributes of the user.

Another type of evaluation method that could be used is that of laboratory based studies, where users are required to carry out tasks. Using the usability testing methodology, difficulties are noted and analysed with a view to potentially making improvements to the interface of a system. Other human-computer interaction techniques that could be applied include cognitive walkthroughs (Nielsen & Mack, 1994), and predictive evaluations (Card et. al., 1983). These techniques, like heuristic evaluations, have the advantage of not requiring end users. They do, however, suffer from the disadvantage of not being able to fully take into account the complexities that may be present in a real-world setting.

Evaluation method	**Attributes**
Heuristic evaluation	Usability experts review a system and comment on usability using a set of heuristics (rules) that embody good design practice. More than one expert should be used. Following independent review, the experts then compare their results and arrive at an agreement on their findings. Results are then shared with interaction designer.
Cognitive walkthroughs	Experts simulate users as they step through or use an interactive system with the intention of uncovering whether users will have an understanding of what to do next, given their goals and the information that has been presented to them. No users are required for this approach.
Predictive evaluation	Experts examine an interface to a system and analyse how long a user will take to complete one or more tasks, based upon estimates it takes to carry out certain actions. This approach is used to evaluate performance of one design over another. No users are required, but will not take account of individual differences (such as fatigue) or environment effects (such as environmental noise) on task performance.
User testing	Users are asked to complete a number of tasks (often within a laboratory) to facilitate gathering of evidence in the form of video of user behaviour and device display. Alternatively, with low fidelity prototypes, facilitator can simulate operation of device to gather early feedback. Costly in terms of time. Requires a number of users.
Ethnographic observation	Can be used to gain a rich understanding of how a system 'fits in' with

	the culture and environment for which it was designed. Requires in-depth observation of user action and gathering of information about usage 'in the field'. Participant observation can be used by an evaluator to immerse him or herself into patterns of use. Most expensive evaluation approach.

Table 2. Summary of evaluation approaches.

There may be many different users of social software systems, each of whom may have their own patterns of usage formed by their own individual needs and requirements. Techniques such as interviewing and observation (and participant observation) enable evaluators to gain an understanding of how a software system is consumed by different user communities. Ethnographic techniques enable evaluators to uncover whether terminology is appropriate for a given audience, or whether an interactive system ultimately disrupts or facilitates communication between different user groups.

These techniques have the advantage of being able to illustrate problems that other techniques may be unable to show. Whilst the user evaluation and cognitive walkthrough techniques may be able to highlight which elements of a user interface might be confusing or difficult to understand to some users, such approaches will not take into consideration the context in which a device is used. The user, for example, may wish to use a product on a bright sunny day on the side of a noisy street, for instance, and discover that audible alerts do not give appropriate feedback and the screen does not have sufficient contrast. Furthermore, through an observational study, for instance, the information storage capacities of a device may be used in ways that were unimagined by the original designer. Such findings may either indicate a shortcoming of a design, or perhaps point the way to new ways in which a system might be used or enhanced. In many respects, there are similarities between the methods that can be used to gather requirements for new systems and the approaches that can be used to evaluate existing systems. Interviews, for instance, can tell us what is required in a new system. They can also be used to tell us how a new system has been used and the opinions users hold about a particular system.

Choosing an effective evaluation strategy depends upon the nature of the product or system that is the focus of an evaluation. The DECIDE framework (Basili et. al., 1994, cited in Preece et. al., 2002) offers some useful guidance. The 'D' in DECIDE refers to the need to determine the goals of an evaluation, i.e. what issue it is trying to explore. An interaction designer might have an objective to explore whether the steps in a task sequence are understandable to a chosen user group, for instance. 'E' refers to the exploring of questions which are to be asked as a part of an evaluation. Each evaluation goal may yield a number of sub-questions. 'C' refers to the *choice* of the evaluation paradigm or approach, such as one of the approaches that are presented in table 2. The identification of practical issues, such as who is to be involved, where is it to take place and how long it is to take is represented by the 'I'. 'D' represents the need to decide upon how to deal with ethical issues. When conducting user testing, it is necessary to obtain *informed consent* to protect both the evaluator and the participant. When ethnographic studies are performed, the issue of consent is something that may need to be re-negotiated during the period of a study, especially if the number of participants affected by a study widens (Hammersley & Atkinson, 1996). Finally, the 'E' refers to the act of the evaluation itself. The results, of which, should be presented in a form that can guide the on-going development of a product or system.

Standards, Guidelines and Legislation

International standards can provide interaction designers with a rich source of information, particularly regarding definitions of common terms, guidance about design and evaluation processes and information about best practice. ISO 9241-11 is a standard which describes the 'ergonomic requirements for office work with visual display terminals: guidance on usability' defines usability as: 'the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and

satisfaction in a specified context of use'. This standard presents a usability framework that considers the user, their task, the equipment they use, and the environment in which a product is used. A number of product goals are defined, namely, effectiveness, efficiency and satisfaction. The standard goes on to present some examples that describe how these various goals can be measured or assessed by way of an evaluation.

International standards do not stop at presenting sets of useful definitions that can help to facilitate communication between designers. ISO 13407, for instance, outlines the stages of human-centered design that can be applied for the development of an interactive system. These stages, it should be stated, are similar to those that are depicted in Figure 1. This standard outlines the principles and rationale of human-centred design, presents the importance of iteration and emphasises the importance of evaluation.

ISO represents one of many international organisations that are offering advice about how to best design and implement interactive systems. Other organisations, such as the W3C is an important source of information about emerging internet standards and systems. As suggested earlier, the web content accessibility guidelines (WCAG) is particularly important, offering guidance relating to the construction of systems that can be used by all users, regardless of whether or not they have a disability (Caldwell et. al., 2008). In the United Kingdom, the WCAG guidelines feature in the British Standard PAS 78:2006, which is entitled 'Guide to good practice in commissioning accessible websites' (BSI, 2006).

Other domain specific standards and specification communities can also provide useful and practical advice about the construction of social software systems. In the world of e-learning technologies, IMS, for instance, can guide the development of useful and accessible applications and systems (IMS, 2002).

One of the advantages of paying attention to the work produced by standard bodies is that they have often been created by a group of experts who are often from a number of different domains. The published documents can represent distilled expertise and may present examples of good practice. Adhering to standards also exposes the possibility of creating interactive systems that are interoperable with other systems, allowing them to be used in new and imaginative ways. It should be added that the process of creating useful standards is one that is very slow. Useful ideas can emerge many years before they find their way into standards documents. This said, it will be interesting to follow the emergence of new standards and guidelines that facilitate the design of new and exciting social software systems.

A final point is that interaction designers and designers of social software systems need to be aware of the important of international legislation. Designers need to be aware of their obligations in terms of conforming to anti-discrimination legislation. Similarly, designers need to be aware of data protection and privacy legislation in the countries in which they are situated. Since social software provides people with means to communicate and share information, issues of security and data integrity are just as important as developing appropriate interaction mechanisms that are suited to the needs of users.

Applying Interaction Design

Let us consider a hypothetical problem; a collection of products which enable users to share recipes with each other. Firstly, there will be a phase of requirements gathering. This may include a series of interviews with potential users and gathering of useful or related documents. Once a set of requirements have been gathered an interaction designer can then begin to consider of a product in terms of user persona and scenarios. The next stage of a design may be to *contextualise* the problem, to allow it to be discussed and further explored by others. As suggested in an earlier section of this chapter, one approach is to create a product storyboard. A storyboard may be a very rough sketch which simply presents a product and the environment in which it may be used and describes the broad interaction paradigm that a particular design is to adopt. A very simple example of a storyboard is presented in Figure 2.

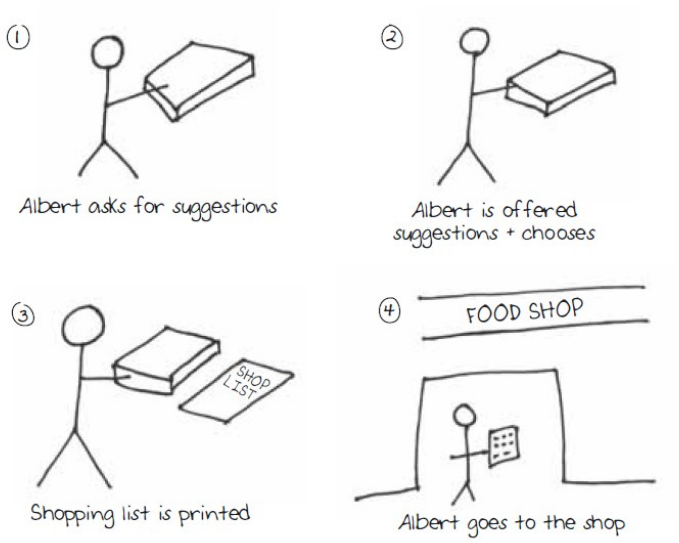

Figure 2. A example storyboard

Once the context and interaction paradigm is understood, a designer (or group of designers) may then begin to consider and compose aspects of an interface. Not only must designers consider the actual components of a screen or display, but also how different screens are conceptually connected or linked to each other. One approach to explore the sequence of screens or interactions is to create what is known as a card based prototype. An example of card sketch is presented in Figure 3.

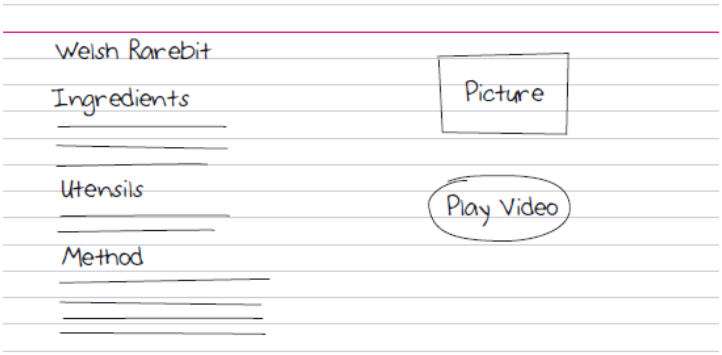

Figure 3. Example of a card from a card-based prototype

Using card sketches, designers can quickly evaluate their designs by presenting them to potential future end users. Designers can efficiently simulate the operation of a computer or system with the intention of evaluating the sequence of actions, the language of the items that are presented on the cards, and whether the interpretation of the requirements is correct. When the design of the cards have been agreed upon, the designer then may move onto a further design iteration by creating a more detailed design, such as interface sketch, perhaps similar to the one shown in Figure 4. Note the increase in the fidelity (or detail) as an increase in the number of design decisions have been made

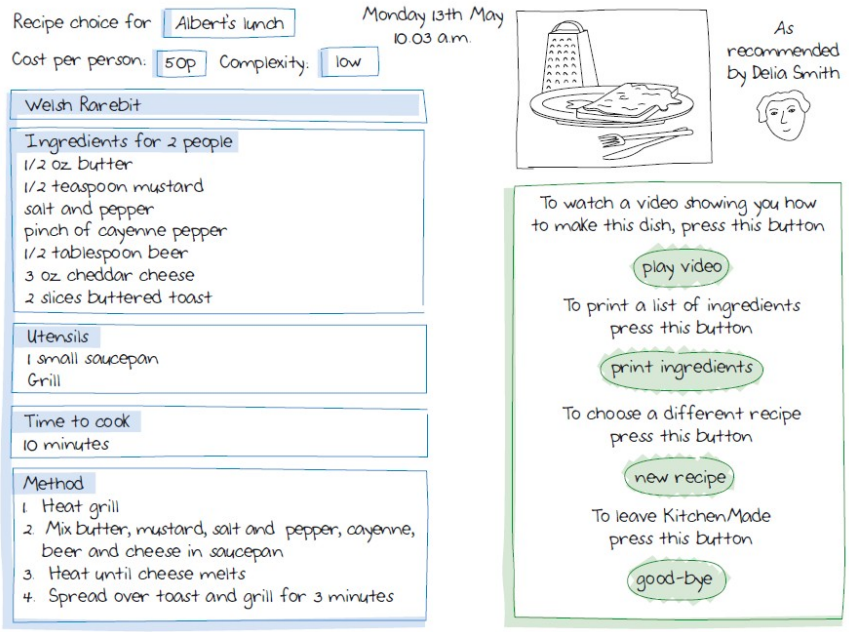

Figure 4. Design of an interface sketch

The process of interaction design and user centred design suggests that designers create a number of different prototypes through a number of different iterations with the intention of creating increasingly detailed designs which increasingly match the needs of all those who comprise the target user group. The designer (or designers) must always be mindful to consider the important issues of user culture and accessibility. The issue of product accessibility is especially important, since issues that relate to the interoperability of assistive technology may only become apparent when more detailed prototypes are considered.

FUTURE RESEARCH DIRECTIONS

Historically, interaction design has been predominantly associated with human-machine interaction. With the advent of increasingly mobile computing devices and near pervasive internet access in many developed countries, it is expected that the focus of HCI will move towards how machines or technology can support human-to-human interaction (Harper, Rodden & Rogers, 2008). The move towards human-to-human interfaces raises a number of important issues. Interactive systems will increasingly have the ability to facilitate different kinds of communication between different groups of people. Network technology will increase the levels of connectedness between people, their interests, their organisations and institutions. Due to the differences between groups and the types of interaction tasks that need to take place, the notion of a static 'one size fits all' interface will change. This change may see the emergence of ways of understanding how to present ambient information (information that is always present or available) in a way that is customised to the task that a user is completing. Just as the issues relating to task and information consumption will remain significant, issues relating to individual and group culture will also remain significant.

The Harper et. al. (2008) report describe a world that has changed from a situation where a person may have used a single computational device, such as a personal computer, to a situation where we may currently make use of a number of different computational devices; a PC at home, a sophisticated mobile phone, and a PC at work. A vision for the future is that a user may use hundreds of different computer systems located on a network 'cloud'. A challenge for the future is to ensure that the usability and user experience of using many different systems is not compromised. Developments in terms of

interoperability and standardization have the potential to help facilitate compatibility between different systems.

The issues of increased levels of connectivity between different systems directly point towards the increasingly important challenge of security and privacy. The emergence of new social software systems may see the development of new higher level concepts or abstractions that can potentially guide the development of new systems. A single user may wish certain types of information to remain private, whereas other information categories may be widely shared. Users will face challenges of how to best control information shared between different groups such as family, friends, work colleagues or a wider group of known associates such as former co-workers, for instance.

The emergence and application of social software may see the increased use of ethnographic methods to gain understanding of whether social software systems can adapt effectively to changing situations and uses. In a situation, where human-to-human communication may change over an extended period of time, longitudinal studies may be the only way to evaluate how successful, powerful and interoperable a particular set of social software utilities are to a chosen group of users. Similarly, just as the development of design principles and heuristics has guided the development of 'discount usability' evaluation for interactive product, the lessons learnt from building and evaluating social software systems may give way to the development of a new generation of social software heuristics.

The notion of usability has given way to the broader concept of user experience. It is prudent to ask whether the processes of ID may eventually take into account wider sets of goals. Whilst the user experience goals that were introduced earlier ask designers to consider whether a system is 'emotionally fulfilling' or whether an interface is 'satisfying', it might be possible to consider user experience in terms of quality of service and the degree to which a social software system permits users to share an emotional experience. The term 'emotional bandwidth' has been used to refer to the extent to which a system permits individuals to either project or detect the emotional expression of others. A challenge for ID theorists and practitioners is to create the tools that enable the expressivity of social software systems to be evaluated.

It is also worth considering the question of whether social software systems themselves might support the design and development of other interactive systems. The process of user-centered design relies on the cycle of producing alternative designs, presenting results to users and making changes based upon the feedback that has been gathered. These processes, given their richness in terms of necessitating communication between different groups of experts is one that has the potential to extensively benefit from the construction of interactive social software.

CONCLUSION

Interaction Design is an interdisciplinary subject that is often situated within the broad boundaries of computer science. There are many themes within interaction design that can be drawn upon to guide the design, development and evaluation of social software systems. Particular attention has been given towards the topics of requirements gathering and evaluation. The techniques used in both situations can be considered to be similar and emphasis has been placed on the potential of ethnographic techniques. These techniques aim to understand the different perspectives of users and the tasks they wish to carry out over periods of time. Standards, guidelines and legislation are considered to be important sources of information for an interaction designer. Standards are considered to be the summary of extended debates about best practice and they have often been created by stakeholders from many different disciplines.

The focus of human-computer interaction has been to understand how people use desktop bound personal computers and how to design effective and usable applications. Computing is now mobile, and through near pervasive web access, users are able to access a myriad of information services quickly and efficiently. Social software systems aim to facilitate human-to-human communication. Interaction design represents a development of HCI which contains sets of techniques and tools that can be used to

create useful, exciting and engaging social software systems that can be used by all members of society. It will be interesting to observe how interaction design changes as new ways of interaction and communication emerge.

REFERENCES

Basili, V., Caldiera, G. & Rombach, D. H. (1994) *The Goal Question Metric Paradigm: Encyclopedia of Software Engineering.* New York: John Wiley & Sons.

British Standards Institute (2006*) PAS 78:2006 - Guide to good practice in commissioning accessible websites*

Card, S. K., Moran, T. P. & Newell, A. (1983) *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Carroll, J.M. (Ed.) (1995) *Scenario-Based Design: Envisioning Work and Technology in System Development*. New York: John Wiley & Sons.

Caldwell, B., Cooper, M., Reid, L.G. & Vanderheiden, G. (2008) *Web Content Accessibility Guidelines (WCAG) 2.0*. Retrieved August 11, 2009, from http://www.w3.org/TR/WCAG20/

Craig, J., Cooper, M., Pappas, L., Schwerdtfeger, R. and Seeman, L. (2009). *Accessible Rich Internet Applications (WAI-ARIA) 1.0*. . Retrieved August 11, 2009, from http://www.w3.org/TR/wai-aria/

Hammersley, M. & Atkinson, P. (1995) *Ethnography : principles in practice, second edition*. London:Routledge.

Harper, R., Rodden, T. & Rogers, Y. (2008) Being Human: human-computer interaction in the year 2020. Cambridge: Microsoft Research Limited, Retrieved August 11, 2009, from http://research.microsoft.com/en-us/um/cambridge/projects/hci2020/

Hofstede, G. (1994) *Cultures and Organisations: Intercultural Cooperation and its Importance for Survival*. HarperCollins.

IMS (2002) IMS Guidelines for Developing Accessible Learning Applications version 1.0 white paper. Retrieved August 11, 2009, from http://www.imsproject.org

International Standards Organisation (1998) *9241-11: Definition of usable software*.

International Standards Organisation (1999) *13407: User-centred design process for interactive systems*.

Kirah, A., Fuson, C., Grudin, J. & Feldman, E. (2005) Ethnography for software development. In Bias, G. & Mayhew, D. J. (Eds) *Cost-justifying usability: an update for the internet age*. (pp.415-446) San Francisco: Morgan Kaufman.

Lim, Y-K, Stolterman, E. & Tenenberg, J. (2008) *The Anatomy of Prototypes: Prototypes as Filters, Prototypes as Manifestations of Design Ideas*. ACM Transactions on Human-Computer Interaction, 15(2).

Moggridge, B. (2006) *Designing Interactions*. Cambridge, MA: MIT Press.

Nielsen, J. & Mack, R. L. (1994) *Usability Inspection Methods*. New York, NY: John Wiley & Sons.

Petrie, H. & Kheir, O. (2007) *The relationship between accessibility and usability of websites*. Proc. SIGCHI conference on Human factors in computing systems, 397-406.

Preece, J., Rogers, R. & Sharp, H. (2002) *Interaction design: beyond human-computer interaction*. New York, NY: John Wiley & Sons.

Marcus, A. & Gould, W. E. (2000) Crosscurrents: cultural dimensions and global web user-interface design. *Interactions*, 7(4), 32-46.

Nielsen, J., & Molich, R. (1990, April) *Heuristic evaluation of user interfaces*, Proc. ACM CHI'90 Conf. (pp. 249-256)

Nielsen, J. (1992, May) *Finding usability problems through heuristic evaluation*. Proceedings ACM CHI'92 Conference, (pp. 373-380).

Wickens, C. D. (1992) *Engineering psychology and human performance: second edition*. HarperCollins.

KEY TERMS & DEFINITIONS

Accessibility: The subject of designing services and physical objects in such a way that they can be used by people with disabilities, often through the application of assistive technologies. In the context of software systems, an accessible system is one that does not present barriers to end users.

Card-based prototype: A series of sketches that can be prepared during to design of a low-fidelity prototype. The operation of a card-based prototype can be demonstrated through the user of a facilitator. Cards are used to explore what should be presented on the surface of an interface and whether the split between different displays are appropriate.

Conceptual Model: A set of integrated ideas about what a product should do and how it should behave. Related to the concept of an interaction mode.

Evaluation: The activity of determining whether a product conforms to a particular set of criteria, for example, whether a product or system is understandable given a particular user audience.

Interaction Design: A multi-disciplinary subject that presents a set of tools and techniques that can be used to design an interactive system.

Interaction Mode: A way of interacting or working with a system. An interaction may be based around an object or based around an activity.

Interaction Paradigm: A way of thinking about how to interact with a system. The most common interaction paradigm is the desktop computer. Alternative paradigms may embody the use of mobile systems and pervasive internet access. Other paradigms include ubiquitous computing, wearable computing and augmented reality, for example.

Persona: A description of a potential user of a system. A persona may include a name, information relating to the users' background knowledge, experience of other similar interactive systems and underlying motivations.

Scenario: A rich textual description of how a particular user (as described within a persona) may use system, device or product. Scenarios can be used as tools to elicit and explore user requirements for interactive products.

Storyboard: A high level sketch that depicts how a product or system is to be used. It contains information that enables designers to understand the environment or the context in which a system is used.

Usability Goal: Design objectives that relate to the functional attributes of a system, such as whether a system effective, efficient, safe to use, offers an appropriate level of functionality, is easily learnable and memorable

Use Case: A textual summary of the interactions between a user and a system, with the focus being upon the normal course of action. Use cases represent one of the ID design tools, and can be created following the production of a scenario description.

User Experience Goal: A wider set of design objectives which relate to how a user may feel about a system.

ACKNOWLEDGEMENTS

Figures 1, 2, 3 and 4 © Copyright The Open University, Fundamentals of Interaction Design course, M364 (2005).