Some Notes about Design

“Design is concerned with how things ought to be in order to attain goals and to function.” — Herbert Simon

Why Design?

Design tasks, such as architectural design, mechanical engineering design, software design, writing, or music composition, require human creativity and judgment in indecisive circumstances. Because of the complexity and the large amounts of information needed to do design, there is great interest in developing computer programs that can help people do design.

Many communities of workers and researchers have investigated design, each from their particular perspective. The Artificial Intelligence community has attempted to formally model design processes in order to automate them. The human-computer interaction community, on the other hand, has viewed the problem from a more "human" perspective, asking questions about how humans and computers can work together to do design. Software engineering and architecture, have looked into the problem of how to improve the way design is done in their respective fields. Despite all those efforts, however, great progress has yet to be made. Why so? What is the problem? Is anything wrong with those approaches?

This course will try to convince you that design tasks are not as simple as they were initially thought. Design is not merely a search through an well-defined space to find a best solution. Design tasks involve cognitive, social, organizational, and cultural issues. And computational support for design is not merely a function of computational power. Computational support for design must itself be designed with an understanding of the fundamental problems that design problems present.

Let’s take a concrete example to illustrate what we mean by design, what problems we are talking about, and what challenges exist in tackling the problems.

A Design Problem: An Example

Making a travel plan for vacation is a design task. Given a limited amount of resources such as your holiday period and budget, you have to make decisions for detailed plans while taking into account your preferences over places, transportation and accommodation.

Let’s say you have holidays between November 10th and 25th, you would like to visit Australia (because a friend of yours said it was a nice place to vacation), and you can spend up to $900 for flight tickets—these are design requirements. A design solution would be a travel itinerary listing which flights to take and where to stay. Solving the design problem requires knowledge about making a vacation plan, about Australia, and about flights going to and from Australia. We can think of two different scenarios, one is by using an expert system, and the other is by interacting with a human travel agent.

Suppose there is an expert system that designs a vacation plan for you. You input your design requirements into the expert system and the system returns a solution to you, or maybe a few alternatives. Your input consists of the destination, departure date, returning date, and your budget constraint. After a minute or two, you will be presented with a display showing two options of flights, leaving in the morning or early afternoon. You click on the option for leaving in the early afternoon, and you will get a pretty printed travel itinerary.

As the second scenario, suppose you visit a travel agent instead of interacting with the expert system. You describe to her that you are interested in going to Australia between November 10th and 25th. She starts asking you questions such as whether you prefer early morning departure, and tells you that you have to change a flight in New Zealand. When you ask her what you can do in New Zealand, she tells you a plenty of nice things about New Zealand. She also tells you that you can save $100 if you fly back on 26th instead of 25th. After fifteen minutes of the conversation with her, you get a ticket to New Zealand, which brings you back to town on the 26th.

The above two scenarios are similar to situations we have observed in our work and to situations that occur in everyday life. Why does one end up with two different design solutions? Is it simply sloppy design of the expert system that led to a different solution from the one created with a human travel agent? Of course one could design an expert system that always asks a customer whether the customer is interested in going to New Zealand instead of the original destination. But it would not solve the problem because there will always be questions that might lead the customer to change the original requirements!

Why does one change the problem requirements? How can one be so indecisive as to change the destination—"weren't you interested in going to Australia in the beginning?

This is the fundamental problem of design: the requirements (the problem itself) is not known precisely before attempts at a solution are made. And only through attempting to solve the problem do the requirements come into focus. The travel planning problem was presented to illustrate this point. Now consider a more complicated design problem, such as the design of a home. During the design process, many initial requirements must be changed and reprioritized, and new requirements emerge. This is not due to a lack of expertise on the part of the architect or the client, but rather it is due to the fact that neither party can understand the problem until they attempt to devise the solution.

Nature of Design

Problems are not given. Design is a ubiquitous activity practiced by numerous professionals. Designers solve problems. But apart from problems in school, most problems in real life are not given. Understanding the problem is the problem. The predominant activity in designing complex systems is the participants teaching and instructing each other. Because complex problems require more knowledge than any single person possesses, communication and collaboration among all the involved stakeholders are necessary. Domain experts understand the practice and system designers know the technology. Rittel termed this situation "symmetry of ignorance" — none of these carriers of knowledge can guarantee that their knowledge is superior or more complete compared to other people's knowledge. To overcome the symmetry of ignorance, as much knowledge from as many stakeholders as possible should be activated with the goal of achieving mutual education and shared understanding.

Lack of Integration of Problem Framing and Problem Solving. Many problem-solving methodologies assume that problems can be clearly defined a priori — before any attempt at a solution is made. The design of artifacts is in most cases an "ill-defined" or "wicked" problem, creating the following dilemma:

- one cannot gather information meaningfully unless the problem is understood;
- one cannot understand the problem without having a concept of the solution in mind; and
- one cannot understand the problem without information about it.

In real life, as opposed to the classroom, problems and tasks themselves are moving targets requiring an integration of problem framing and problem solving, such that the work in progress suggests ways to proceed, and the development of a solution causes the understanding of the task to grow and change. The research needs to be concerned as much with problem definition as problem solutions by stressing the importance of externalization enabling designers to represent both problems and solutions.

Lack of Integration of Action and Reflection. Designers are engaged in a cyclic process of action and reflection. Action is governed by a nonreflective thought process and proceeds until it breaks down. A breakdown occurs when the designer realizes that nonreflective action has resulted in unanticipated consequences — either good or bad. Schön describes this feedback by saying that designers engage in a conversation with materials and solution attempts in which they simultaneously create solution attempts and listen to the "back-talk of the situation". In many cases the back-talk of the situation is insufficient. Argumentative information, such as design rationale, manuals, and help systems, are provided to help such reflection processes. However, without an integration of action and reflection, argumentation is unable to serve our actions and large argumentative spaces will remain unused. It is necessary to contextualize argumentation to the designer's task at hand.

Knowledge Communication in Design Communities

Design is a rich setting in which to study computer-mediated communication. Large and complex design projects cannot be accomplished by any single person, and they often cut across different established disciplines, requiring expertise in a wide range of areas. Software design projects, for example, involve designers, programmers, human-computer interaction specialists, marketing people, and user participants. Design projects are unique, and therefore each design project requires learning and produces new knowledge in the form of understanding as well as artifacts. Complexity in design arises from the need to synthesize stakeholders' different perspectives of a problem, the management of large amounts of information relevant to a design task, and understanding the design decisions that have determined the long-term evolution of a designed artifact. Successful projects must overcome many barriers to communication and shared understanding.

Media change the nature of learning and communication in design. Ideally, new media will improve both individual and collaborative design by augmenting the cognitive abilities of designers and allowing them to transcend some of the barriers that have limited knowledge creation and sharing in design.

Design

Design is a ubiquitous activity that is practiced in everyday life as well as in the workplace by professionals. It is not restricted to any specific discipline, such as art or architecture, but instead is a broad human activity that pursues the question of "how things ought to be", as compared to the natural sciences, which study "how things are". It is a fundamental activity within all professions: architects and urban planners design buildings and towns, lawyers design briefs and cases, politicians design policies and programs, educators design curricula and courses, writers design novels and technical documentation, psychologists design experiments, and software engineers design computer programs.

Designers solve problems. But apart from problems in school, most problems in real life are encountered, not given. For these problems, understanding the problem is the problem. Real-life problems must be framed, a process in which the important objects are determined and desired outcomes are defined.

Emphasizing the integration of problem framing and problem solving casts design as a search for a problem space rather than just within a problem space. It brings into question all design methodologies that are founded on a separation of analysis and synthesis. Furthermore, it emphasizes the importance of problem owners (those for whom an artifact is designed) as stakeholders in the design process because they have the authority and the knowledge to reframe the problem as the problem space becomes better understood.

Our research in design integrates the task of problem framing with that of problem solving by stressing the importance of externalizations that enable designers to represent both tasks. In this sense, externalizing ideas is not a matter of emptying out the mind but of actively reconstructing it, forming new associations, and expressing concepts in external representations while lessening the cognitive load required for remembering them:

> "Externalization produces a record of our mental efforts, one that is 'outside us' rather than vaguely 'in memory'. ... It relieves us in some measure from the always difficult task of 'thinking about our own thoughts' while often accomplishing the same end. It embodies our thoughts and intentions in a form more accessible to reflective efforts." ((Bruner, 1996), p. 23)

Differentiating Design Approaches

Design processes involve stakeholders (often coming from different disciplines) who create artifacts. For many design activities, one can distinguish between *design time* (when the artifact is being designed) and *use time* (when the artifact is being used). At design time, a major challenge is to imagine how users will experience artifacts, whereas at use time the users are actually experiencing the artifacts.In *professionally dominated design*, professional designers (such as architects, software developers, urban planners, and teachers) engage in design methodologies founded on the belief that they understand the users' needs. At design time, they create artifacts with which users "have to live" at use time. In professionally dominated design, the "experts" see the creation of artifacts as their primary tasks (e.g., architects build buildings, software developers create software systems, urban planners design cities, and teachers develop courses); and understanding and communicating with other stakeholders are seen as secondary tasks representing extra work and thereby taking resources away from the primary task.

Participatory design approaches seek to involve users more deeply in the process as co-designers by empowering them to propose and generate design alternatives themselves. Participatory design supports diverse ways of thinking, planning, and acting, thus making work, technologies, and social institutions more responsive to human needs. It requires the social inclusion and active participation of the users. It is a response to the theoretical argument that design problems are ill-defined and wicked and therefore cannot be delegated to experts. Instead, all stakeholders who are *owners of problems* must have a voice in the design process and they must participate in the framing of the problem.

Communication processes between designers and clients in participatory design face two barriers:

(1) clients may not know exactly what they want; and

(2) stakeholders lack a common language that allows them to educate each other, propose new visions, understand and critique these proposals, and come to a shared understanding of how things should be.

Developers are often biased toward working in their own language and formalisms, which is a barrier for users, who are forced to express their knowledge in the developer's vocabulary. Communication breakdowns occur when developers and users do not have a shared context. The challenge for communication is to establish a shared context that allows for communication and the accumulation of shared understanding.

Despite the best efforts at design time, designed artifacts need to be evolvable at use time to fit new needs, account for changing tasks, and incorporate new technologies. However, design approaches (whether done for users, by users, or with users) have traditionally focused primarily on activities and processes taking place at design time and have given little emphasis and provided few mechanisms to support systems as *living* entities that can be evolved by their users (see Table 1).

Meta-design approaches characterize objectives, techniques, and processes for creating new media and environments that allow the owners of problems to act as *designers*. A fundamental objective of meta-design is to create socio-technical environments that empower users to engage in creating knowledge rather than being restricted to the consumption of existing knowledge.

Meta-design extends the traditional notion of system design beyond the original development of a system to include an ongoing process in which stakeholders become *co-designers*—not only at design time, but throughout the whole existence of the system. A necessary, although not sufficient, condition for users to become co-designers is that software systems include advanced features that permit users to create complex customizations and extensions. Rather than presenting users with closed systems, meta-design approaches provide them with opportunities, tools, and social reward structures to extend the system to fit their needs. Meta-design shares some important objectives with user-centered and participatory design, but it *transcends* these objectives in several important dimensions and it changes the processes by which systems and content are designed. Meta-design shifts control over the design process from designers to users and empowers users to create and contribute their own visions and objectives at use time as well as at design time. Meta-design is a useful perspective for projects for which 'designing the design process' is a first-class activity, meaning that creating the technical and social conditions for broad participation in design activities (in both design time and use time) is as important as creating the artifact itself.

Table 1 summarizes the role of the user in professionally dominated, participatory, and meta-design approaches. Only meta-design views the users as active participants and designers throughout the lifecycle of a designed artifact.

Table 1: The Role of Users in Different Design Approaches

Design Approach	Design Time	Use Time
Professionally-dominated des	Users have no voice	Users have to live with artifacts designed by othe
Participatory design	Users are active participants; systems are desig as complete systems artifacts	Users are consumers of artifacts designed with th input, but artifacts cannot be evolved to serve unforeseen needs
Meta-design	Users are active participants; systems are desig as seeds; design is focused on design for participation, as well as use	Users can act as designers and evolve the artifact fit new needs