

Scenario-Based Design

Mary Beth Rosson and John M. Carroll

Department of Computer Science and Center for Human-Computer Interaction

Virginia Tech, Blacksburg VA

Chapter 53 in J. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Lawrence Erlbaum Associates, 2002, pp. 1032-1050.

1. The Basic Idea

Scenario-based design is a family of techniques in which the *use* of a future system is concretely described at an early point in the development process. Narrative descriptions of envisioned usage episodes are then employed in a variety of ways to guide the development of the system that will enable these use experiences.

Like other user-centered approaches, scenario-based design changes the focus of design work from defining system operations (i.e., functional specification) to describing how people will use a system to accomplish work tasks and other activities. However, unlike approaches that consider human behavior and experience through formal analysis and modeling of well-specified tasks, scenario-based design is a relatively lightweight method for envisioning future use possibilities.

A user interaction scenario is a *sketch of use*. It is intended to vividly capture the essence of an interaction design, much as a two-dimensional, paper-and-pencil sketch captures the essence of a physical design.

2. A Simple Example

Scenarios are stories. They consist of a setting, or situation state, one or more actors with personal motivations, knowledge, and capabilities, and various tools and objects that the actors encounter and manipulate. The scenario describes a sequence of actions and events that lead to an outcome. These actions and events are related in a usage context that includes the goals, plans, and reactions of the people taking part in the episode.

Table 1 presents three brief scenarios in which a member of a club uses different network tools to interact with club members. In all of these scenarios, the person's goal is to visit a club and interact with her friends at the club. The scenarios contrast three ways that such a goal might be supported by computer network technologies. Each is a potential "solution" to Sharon's needs, but the user experience varies from asynchronous text-based reading and posting, to a real-time graphical simulation of a meeting place.

Designers can quickly construct scenarios like these in order to make envisioned possibilities more concrete. The example contrasts three contemporary approaches to online interactions, but not as an abstraction, not as a list of features or functions. It contrasts three episodes of human-computer interaction and personal experience.

A. Science Fiction Club in a Web forum
<p>After three years at Virginia Tech, Sharon has learned to take advantage of her free time in-between classes. In her hour between her morning classes, she stops by the computer lab to visit the science fiction club. She has been meaning to do this for a few days because she knows she'll miss the next meeting later this week. As she opens a Web browser , she realizes that this computer will not have her bookmarks stored, so she starts at the homepage of the Blacksburg Electronic Village. She sees local news and links to categories of community resources (businesses, town government, civic organizations). She selects "Organizations", and sees an alphabetical list of community groups. She is attracted by a new one, the Orchid Society, so she quickly examines their Web page before going back to select the Science Fiction Club page. When she gets to the club page, she sees that there are two new comments in the discussion on Asimov's <i>Robots and Empire</i>, one from Bill and one from Sara. She browses each comment in turn, then submits a reply to Bill's comment, arguing that he has the wrong date associated with discovery of the Zeroth Law.</p>
B. Science Fiction Club in a Community MOO
<p>After three years at Virginia Tech, Sharon has learned to take advantage of her free time in-between classes. In her hour between her morning classes, she stops by the computer lab to visit the science fiction club. She has been meaning to do this for a few days because she knows she'll miss the next meeting later this week. As she starts up the Blacksburg community MOO, she can see that the last person using this computer must have been interested in orchids, because the welcoming text describes her location as an orchid garden, along with Penny and Alicia, who are discussing some new exotic varieties. The text description mentions an exit to Main Street, so she leaves the garden and starts moving south. Along the street she runs into George, who is working on a banner for the fair. She gives him a quick hello, and continues southward until she sees an eastward exit will take her to Eastenders Pub; this is where the Science Fiction Club meets. She enters the room and is told that Bill and Sara are already there, along with a pitcher of Newcastle Brown. She can tell from their current comments that they have been discussing the timeline from Asimov's <i>Robots and Empire</i>.</p>
C. Science Fiction Club in a Collaborative Virtual Environment
<p>After three years at Virginia Tech, Sharon has learned to take advantage of her free time in-between classes. In her hour between her morning classes, she stops by the computer lab to visit the science fiction club. She has been meaning to do this for a few days because she knows she'll miss the next meeting later this week. When she tries to start up the online collaborative environment, she finds that this computer does not have the client, so she waits for a minute or two while it is automatically downloaded and installed. After she logs in, she is taken back to her previous visit location, and sees the familiar panoramic view of her livingroom, her to-do lists and sketchpad, and the interactive map of Blacksburg. She positions and zooms in on the map until she can see downtown buildings She enters the Eastenders Pub subspace, where the science fiction club usually meets. She sees a panoramic image of bar, faces that show Bill and Sara are here, a food and drink menu, and various standard tools. The map updates to show a floorplan of the Pub—the dining room, the darts room, the office, and the main bar. Bill and Sara are using a chat tool and a shared whiteboard to sketch an event timeline for Asimov's <i>Robots and Empire</i>. Joining Bill and Sara in the chat tool, she types "Based on the Zeroth Law, I'm afraid I must drink some of your beer".</p>

Table 1: Three scenarios for a university student attending a club meeting online.

Scenarios of envisioned use can be successively detailed to discover and address finer-grained design issues. They serve as grist for group brainstorming, to develop further

alternatives, or to raise questions about the assumptions behind the scenarios. They can be used to analyze software requirements, as a partial specification of functionality, and to guide the design of user interface layouts and controls. They can be used to identify and plan evaluation tasks that will be performed by usability test participants.

3. Why Scenario-Based Design?

One reason that scenarios have become so popular in interactive system design is that they enable rapid communication about usage possibilities and concerns among many different stakeholders. It is easy to write simple scenarios such as those in Table 1, and takes only a little more effort to enrich it with a rough sketch or storyboard. When designers are working through ideas, they want to make progress quickly, so that they can obtain feedback and continue to refine their ideas. Scenarios are one way to do this.

The design of an interactive system is an ill-defined problem. Such problems tend to evoke a problem-solving strategy termed *solution-first* (Cross, 2001). In the solution-first strategy, designers generate and analyze a candidate solution as a means of clarifying the problem state, the allowable moves, and the goal. They exploit the concreteness of their own solution proposals to evoke further requirements for analysis.

Hazards of the solution-first approach	How scenario-based design can help
Designers want to select a solution approach quickly, which may lead to premature commitment to their first design ideas	Because they are concrete but rough, scenarios support visible progress, but also relax commitment to the ideas expressed in the scenarios
Designers attempt to quickly simplify the problem space with external constraints, such as the reuse of familiar solutions	Because they emphasize people and their experiences, scenarios direct attention to the use-appropriateness of design ideas
Designers are intent on elaborating their current design proposal, resulting in inadequate analysis of other ideas or alternatives	Because they are evocative and by nature are incomplete, scenarios promote empathy and raise usage questions at many levels

Table 2: Concerns stemming from the solution-first approach to design, and aspects of scenario-based design that address each concern.

A solution-first approach to design is energizing, effective, and efficient; it explains the popularity of contemporary system development approaches like rapid prototyping (Wasserman & Shewmake, 1982) and extreme programming (Beck, 1999). But this general strategy also entrains well-known hazards (Cross, 2001): Designers tend to generate solutions too quickly, before they analyze what is already known about the problem and possible moves. Once an approach is envisioned, they may have trouble abandoning it when it is no longer appropriate. Designers may too readily reuse pieces of a solution they have used earlier, one that is familiar and accessible, but perhaps not appropriate. They may not analyze their own solutions very well, or they may consider too few alternatives when exploring the problem space. In the next three sections we consider how scenario-based design may help to minimize these hazards of solution-first problem solving (see Table 2).

3.1 Scenarios are concrete but rough

Design analysis is always indeterminate, because the act of design changes the world within which people act and experience. Requirements always change (Brooks, 1995). When designs incorporate rapidly-evolving technologies, requirements change even more rapidly. The more successful, the more widely-adopted, and the more impact a design has, the less possible it will have been to determine its correct design requirements. And in any case, refinements in software technology and new perceived opportunities and requirements propel a new generation of designs every 2-3 years.

Design representations that are at once concrete but flexible help to manage ambiguous and dynamic situations. Analysts must be concrete to avoid being swallowed by indeterminacy; they must be flexible to avoid being captured by false steps. Systematic decomposition is a traditional approach to managing ambiguity, but it does not promote flexibility. Instead designers end up with a set of concrete sub-solutions, each of which is fully specified. Unfortunately, by the time the set of sub-solutions is specified, the requirements often have changed.

Scenarios of use reconcile concreteness and flexibility. A scenario envisions a concrete design solution, but it can be couched at many levels of detail. Initial scenarios are often extremely rough. They specify a possible design by specifying the tasks users can carry out, but without committing to lower-level details describing *how* the tasks will be carried out, or *how* the system will present the functionality for the tasks. The examples in Table 1 are at an intermediate level, with some detail regarding task flow, but little information about individual user-system interactions.

Concrete material is interpreted more easily and more thoroughly than abstract material. For example, people remember a prototypical example far better than they remember the abstract category to which it belongs (Medin & Schaffer, 1978; Rosch et al., 1976). Incomplete material tends to be elaborated with respect to personal knowledge when it is encountered. This process of elaboration creates more robust and accessible memories, relative to memories for more complete material (Wertheimer, 1938). The combination of concreteness and incompleteness in scenarios engages a powerful variety of constructive cognitive processes.

The fluidity of design situations demands provisional solutions, tentative commitments. Yet if every design decision is suspended, the result will be a design space, not a design. A scenario is a concrete design proposal that a designer can evaluate and refine, but it is also rough, so that it can be easily altered, and many details can be deferred.

3.2 Scenarios maintain an orientation to people and their needs

Designers need constraints; there are just too many things that might be designed. The current state of technology development makes some solutions impossible and others irresistible: On the one hand, designers cannot use technology that does not yet exist. On the other hand, designers are caught up in a technological zeitgeist that biases them toward making use of the latest gadgets and gizmos. They are likely to be biased toward familiar technologies, even when they are aware of limitations in these technologies.

Scenarios are work-oriented design objects. They describe systems in terms of the work that users will try to do when they use those systems, ensuring that design will remain focused on the needs and concerns of users (Carroll & Rosson, 1990). Scenarios address what has been called the “representational bias” in human cognition—people overestimate the relevance of things that are familiar to them (Kahneman & Tversky 1972; Tversky & Kahneman, 1974). For instance, designers in a Web development company with years of experience in forms-based transactions will see this interaction paradigm as a solution to problems that might be better served by real-time interaction techniques. It is difficult to move beyond the familiar, but generating and sharing a vivid representation of exceptions to the status quo can promote innovating thinking. Scenarios describing unusual but critical circumstances can provide such a perspective.

The reuse of familiar ideas is just one type of constraint that designers may apply in their solution-first problem solving. Other constraints may arise from the organizational structures within which the design work is embedded. Design projects are often chartered with a priori commitments to follow a systematic decomposition process. This makes them easy to manage, but unlikely to succeed with respect discovering the real requirements of users and clients. Schedules and resources are often assigned in ways that create on-going conflicts between system designers and usability engineers. Usability engineers need to evaluate scenarios and prototypes at every stage of system development, but if schedules and resources do not provide for this, this work can conflict with software construction and refinement.

Constraints such as these can distract designers with ancillary factors so that they lose sight of what is essential in the design project, namely, the needs and concerns of users. The designer can become “unsituated” with respect to the real design situation, which is not the marketing manager’s projections, or the instructional designer’s list of steps, or the software engineer’s system decomposition. The real design situation is the situation that will be experienced by the user, and designers need to stay focused on that.

Scenarios can be made even more effective as work-oriented design objects when users are directly involved in creating them. Ackoff (1979) argues that the indeterminacy of design situations makes it imperative that *all* stakeholders participate directly. Scenarios support a simple and natural process of participatory design, where prospective users begin by enacting or relating episodes of current activities, then work iteratively with designers to transform and enrich these scenarios with the opportunities provided by new technologies (Carroll et al., 2000; Chin, Rosson, & Carroll, 1997).

3.3 Scenarios are evocative, raising questions at many levels

There is a fundamental tension between thinking and doing: thinking impedes progress in doing, and doing obstructs thinking. Sometimes this conflict is quite sharp, as when one must stop and think before taking another step. But frequently it is more a matter of trading off priorities. Designers are intelligent people performing complex and open-ended tasks. They want to reflect on their activities, and they routinely do reflect on their activities. However, people take pride not only in what they know and learn, but in what they can do and in what they actually produce.

Donald Schön (1983; 1987) discusses this conflict extensively in his books on reflective practice. For example, he analyzes a coach reacting to an architecture student's design concept for a school building, which included a spiral ramp intended to maintain openness while breaking up lines of sight (she calls the idea "a Guggenheim"):

"... when I visited open schools, the one thing they complained about was the warehouse quality — of being able to see for miles. It [the ramp] would visually and acoustically break up the volume." (Schön, 1987, page 129)

In this episode, the coach feels that the student needs to explore and develop her concept more thoroughly, noting that a ramp has thickness and that this will limit her plans to use the space underneath the ramp; he urges her to draw sections. However, he does not justify this advice; as Schön puts it, he does not reveal "the meanings underlying his questions" (Schön, 1987, page 132). Schön regards this as a hopeless confrontation in which no progress can be made on the particular design project, or on the larger project of understanding how to design. Both the student and the coach are willing to act publicly and to share actions, but they do not reflect enough on their own and one another's values and objectives, and on their interpersonal dynamics.

Reflection is not always comfortable; it forces one to consider one's own competence, to open oneself to the possibility of being wrong. Nonetheless, designers create many opportunities for reflection, for example organizing design review meetings, or building prototypes for formative evaluation. Such activities promote identification and integration of different perspectives; they raise concrete and detailed design issues to guide further work. In this way they help designers to reflect on the work they've already done. But they do not evoke reflection *in the context of doing design*. Design reviews and formative evaluations are ancillary activities that must be coordinated with design itself.

Scenarios help designers to reflect about their ideas in the context of doing design. The narrative is written to evoke an image of people doing things, pursuing goals, using technology in support of these goals. The story enables readers to empathize with the people in the situation, which in turn leads to questions about motivations, intentions, reactions, and satisfaction. For example, in the middle scenario from Table 1, is it valuable to Sharon to opportunistically encounter friends on her way to the club meeting? What effect does her recognition of the town's layout have on her success or experience at navigating the online environment?

Scenarios promote reflection and analysis in part because the human mind is adept at overloading meaning in narrative structures, both in generation and interpretation, as illustrated by the remarkable examples of dreams (Freud, 1900), myths (Levi-Strauss, 1967), and folktales (Propp, 1957). It is well-known that when people communicate, they rely on the *given-new contract* (Haviland & Clark, 1974): they assume or allude to relevant background information, then present what is novel. This normative structure eases both the generation and interpretation of narratives.

Schön (1983) describes design as a "conversation" with a situation comprised of many interdependent elements. The designer makes moves and then "listens" to the design situation to understand their consequences:

“In the designer’s conversation with the materials of his design, he can never make a move which has only the effects intended for it. His materials are continually talking back to him, causing him to apprehend unanticipated problems and potentials.” (page 101)

When a move produces unexpected consequences, and particularly when it produces undesirable consequences, the designer articulates “the theory implicit in the move, criticizes it, restructures it, and tests the new theory by inventing a move consistent with it” (page 155).

Scenarios often include implicit information about design consequences. Returning to the scenarios in Table 1, the archived forum in the first scenario allows Sharon to arrive at a discussion “at any time”, find out what has been said, and make a contribution. In the second scenario, it is Sharon’s navigation “down Main Street” that sets up her casual encounters with other community residents. At the same time these features have less positive consequences, for example the need to browse and read each of the comments in turn, or the requirement that she “walk” to get to the meeting place. These tradeoffs are important to the scenarios, but often it is enough to imply them (this is an aspect of the roughness property discussed above).

There are times, however, when it is useful to reflect more systematically on these relationships, to make them explicit. In another situation Sharon may find the archived discussion too long or disorganized to browse, or she may be distracted by friends on Main Street and never make it to the club meeting. These alternative scenarios present a failure with respect to the starting goal. To understand, address, and track the variety of both desirable and undesirable consequences of the original annotation design move, the designer might want to make explicit the relevant causal relationships in a scenario. Doing so provides yet another view of the envisioned situation (see Table 3).

Scenarios and analyses such as shown in Table 3 can help designers move more deliberately toward specific consequences. For example, it might be decided to provide discussion-summarization support in an online forum, so that the convenience of the forum interaction is obtained, but so that it also scales well to complex discussions. Alternatively, the opportunistic encounters with other residents might be considered desirable enough that efforts would be made to provide “presence: information in the Web forum. As each elaboration is envisioned and proposed, it too is explored for further consequences and interdependencies.

A. Science Fiction Club on a Web forum
<p>Accessing an online meeting through an Web discussion forum</p> <ul style="list-style-type: none"> + enables convenient browsing of an entire discussion at many points in time or place - but when a discussion becomes long or complex, it may be difficult to browse or understand
B. Science Fiction Club in a Community MOO
<p>Accessing an online meeting room by “walking” through a spatial model of the town</p> <ul style="list-style-type: none"> + allows fortuitous social encounters while moving around the MOO + evokes application and development of real world spatial knowledge about the town - but navigation to non-immediate sites may be tedious or awkward - but the overall spatial model may be poorly evoked by step-by-step navigation
C. Science Fiction Club in a Collaborative Virtual Environment
<p>Accessing an online meeting by clicking on a spatial map of the town</p> <ul style="list-style-type: none"> + simplifies navigation through direct pointing in an interactive map + suggests a town-oriented spatial context for the meeting within in the community - but a realistic map of the town may be perceptually complex - but a data-rich map may require a long time to download or update during use

Table 3: Design features of the scenarios presented in Table 1, each expanded to consider possible positive and negative consequences for users in the source scenario or closely-related alternatives. Each analyzed feature with its consequences is called a *claim*.

Scenarios of use are multifarious design objects; they can describe designs at multiple levels of detail and with respect to multiple perspectives. In this way they can help designers reflect on several aspects of a problem situation simultaneously. The scenarios in Table 1 provide a high-level task view, but can also be elaborated to convey the moment-to-moment thoughts and experiences of the actors in order to provide a more detailed cognitive view, or in terms of moment-to-moment actions to provide a more detailed functional view. Or, they might be elaborated in terms of the hardware or software components needed for implementing the envisioned functionality (Rosson & Carroll, 1995; Wirfs-Brock, 1995). Each of these variations in resolution and perspective is a permutation of a single underlying scenario. The permutations are integrated through their roles as complementary views of the same design object.

Using scenarios in this way makes them a more powerful design representation. They allow the designer the flexibility to develop and analyze key usage scenarios in great detail, for example to describe core application functionality, while merely sketching less critical scenarios. At the same time, designers are able to switch among multiple

perspectives, for example directly integrating usability views with software views. Such a flexible and integrative design object can help designers manage the many interdependent consequences implied by design moves (Rosson & Carroll, 2000).

4. A Framework for Scenario-Based Design

The concrete and work-oriented nature of scenarios make them an effective representation for human-centered design activities, particularly when these activities include participation by end-users or other stakeholders (Carroll et al., 2000; Chin, Rosson, & Carroll, 1997; Muller, 1992; Muller et al., 1995; Karat, 1995; Karat & Bennett, 1991; Rosson & Carroll, 2001b). Scenarios can be quickly developed, shared, and revised; they are easily enriched with sketches, storyboards or other mock-ups (Erickson, 1995; Kyng, 1995). A scenario of use can be directed at many concerns in system development, including documentation design and object-oriented software design (Carroll, 1995; Carroll 2000). Given these many virtues, it is no surprise that scenarios are pervasive in software design and development (Rosson, Maass, & Kellogg, 1989; Weidenhaupt et al., 1998). But here we expand this generally-accepted view of scenarios as a user-centered design representation. We offer a programmatic framework for employing scenarios of use in interactive system design (Carroll, 2000; Rosson & Carroll, 2001b).

The framework summarized in this section incorporates scenario-based analysis and design into all phases of system development, from requirements analysis through usability evaluation and iterative development. We exploit the general advantages of scenario-based design described in Section 3, but at the same time show how to make the impacts of scenario-based reasoning comprehensive and systematic. The overall process is one of usability engineering, where the scenarios support continual assessment and elaboration of the system's usefulness, ease of use, and user satisfaction. The aim is to develop a rich understanding of current activities and work practices, and to use this understanding as a basis for activity transformation.

Figure 1 provides an overview of the scenario-based design framework (hence SBD). We assume that system development begins with an initial vision or charter, even though this may be quite sketchy or non-binding. This vision motivates a period of intense analysis during which the current situation is examined for problems and opportunities that might be addressed by available technologies. The analysts' understanding of the current situation is communicated in problem scenarios and claims. Problem scenarios describe prototypical stakeholders engaged in meaningful activities; the claims enumerate features of the current situation that are understood to have important consequences—both positive and negative—for the scenario actors.

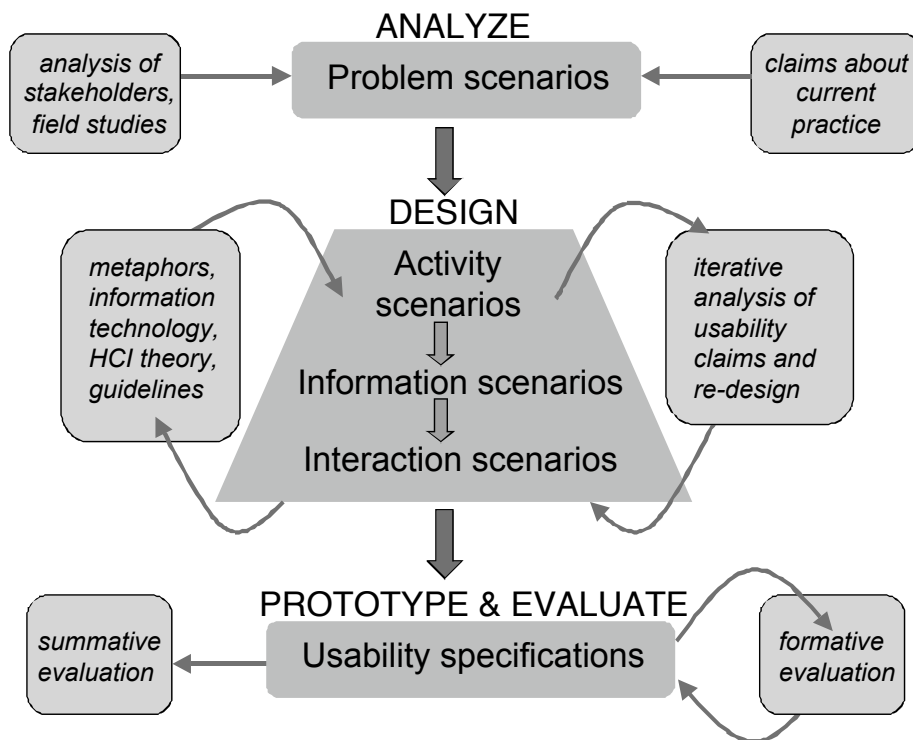


Figure 1: An overview of the scenario-based design (SBD) framework. Scenarios serve as a central representation throughout the development cycle, first describing the goals and concerns of current use, and then being successively transformed and refined through an iterative design and evaluation process (from Rosson & Carroll, 2001b).

The problem scenarios are transformed and elaborated through several phases of iterative design. Design envisionment is inspired by metaphors and technology options, but at the same time is constrained by the designers' knowledge of interactive system design. Each set of scenarios is complemented by claims that analyze the possible positive and negative consequences of key design features. Claims analysis leads designers to reflect on the usage implications of their design ideas while the ideas are being developed.

Scenario-based design is guided by usability evaluation throughout development. Each narrative serves as a test case for analytic evaluation; each claim hypothesizes usability outcomes for one or more test cases. Design scenarios are also evaluated more directly through empirical usability studies. In these the claims analysis structures a mediated evaluation process, wherein the hypothesized usage impacts are operationalized and tested explicitly (Scriven, 1967). The empirical findings are interpreted with respect to the ongoing claims analysis, refining or redirecting the design efforts. We turn now to a brief example illustrating the key elements of the framework.

4.1 Requirements Analysis

A challenge for any software development project is identifying the complete and correct set of requirements (Brooks, 1995). Many system requirements are functional, addressed by the actual services and information provided by the final system. Other requirements are nonfunctional, for example the measured quality of the software implementation or user

interactions, or pragmatic features of the system development process like schedule, cost, or delivery platform (Rosson & Carroll, 2000; Sommerville, 1992; Sutcliffe & Minocha; 1998). In SBD we express an initial analysis of requirements as a *root concept* (Table 4). The root concept enumerates key aspects of the team’s starting vision; it is used to guide further analysis and elaboration of system requirements.

Component	Contributions to the root concept
High-level vision	Club members interact anytime, anywhere; develop shared resources
Basic rationale	Network-based interaction overcomes barriers of place and time Digital media are convenient to archive, organize, and retrieve over time
<u>Stakeholder group:</u> Club officer Club member Prospective member	Convenient scheduling and posting of shared events and information Ongoing access to club activities, persistent recognition of contributions Self-paced exploration of club vision, history, and membership
Starting assumptions	Open-ended participatory design process Members have pervasive access to personal computers and network connections Community computing development accomplished via volunteer efforts

Table 4: A root concept for developing online activities for a Science Fiction Club

Table 4 contains a root concept for the science fiction club example that we use to illustrate the framework. The starting vision and rationale in this case are quite straightforward: there are obvious advantages to meeting with associates in person, but the constraint of same-time, same-place limits frequency and/or length of such meetings. Moving some of the club’s activities online increases interaction opportunity and flexibility. At the same time, a side-effect is that digital interactions can be stored and for other purposes.

The root concept also documents the team’s shared beliefs about the project’s major stakeholders. A *stakeholder* is any person or organization who will be affected (either positively or negatively) by the system (Checkland, 1981; Muller, 1991). It is important to take a broad view of stakeholders, particularly early in requirements analysis, so that appropriate individuals and groups will be represented in analysis activities. In the example, we consider several different types of club members because they will have distinct goals with respect to system use—an officer who might find an online system convenient for scheduling and posting information; a “regular” club member who will now have more options for participating, and a prospective member who can learn about the club and its activities in a customized, self-directed fashion.

Although the emphasis of SBD is on analyzing and developing usable functionality, there may be a range of nonfunctional concerns that will constrain development. These are documented as starting assumptions in the root concept. For our example project, we assume that the design of the online club software will involve considerable participation by stakeholders, that club members already have the computing resources needed to access the system, and that use and maintenance of the final system will take place through members’ volunteer efforts.

The root concept lays the groundwork for analyzing the club's current activities. This might involve fieldwork, for example a visit to a meeting where notes, photographs, or videotaped records are made. It may be more modest, perhaps a survey of current members aimed at eliciting descriptions of activities, or a series of semi-structured interviews with club officers, members, and prospective members. A rich source of requirements are activity artifacts—for instance a club newsletter or calendar, reports or other shared products created by the group. Such artifacts are an excellent source of implicit information about stakeholders' values and activities (Bodker, 1992; Norman, 1988; Rosson & Carroll, 2001).

Field studies of current practices generate rich and diverse data about needs and opportunities. In order to direct these data productively toward design efforts, a more abstract representation of themes and relationships is required. An affinity diagram (group members post and organize individual observations; Beyer & Holtzblatt, 1998) is very helpful in discovering these themes. Other useful techniques include diagrams of the stakeholder relationships, hierarchical task analysis of central activities, and summaries of how collected artifacts support group activities (see Rosson & Carroll, 2001; Chapter 2).

In SBD a key result of requirements analysis is a set of problem scenarios and claims. A problem scenario is a narrative of current practice that synthesizes actors, themes, relationships, and artifacts discovered in the field work. These scenarios are not design-neutral however. Even during early analysis activities, we assume that the team has a vision of how technology might enhance current practice. The fieldwork and related analyses will inevitably be colored by this vision (Carroll et al., 1994). If the team fails to establish a vision, or creates inconsistent or contradictory visions, this too will influence requirements analysis, but in a less positive fashion.

An effective technique for generating problem scenarios is to first describe a set of hypothetical stakeholders—individuals who will represent the different sorts of people studied during the fieldwork. It is important to create a rich but realistic view of these individuals, as they form the basis for describing and later transforming current activities and experiences (the technique is similar to the *persona* concept in Cooper, 1998). Our examples focus on the experiences of Sharon, a busy third-year student at a large state university; one of Sharon's interests is science fiction, and she pursues this interest through reading and discussion with friends in a science fiction club.

The scenario in Table 5 conveys some aspects of the club's current practice; it enacts a typical activity and simultaneously communicates issues uncovered during fieldwork. Problem scenarios like this may be based directly on an observed episode, or they may be entirely synthetic. The goal is to introduce and contextualize the themes and relationships that will guide subsequent design work. This particular story combines our concept of a typical student club member with issues related to real world meetings—for example, the need to arrive in a particular place at a particular time, the protocol for being greeted, for entering into a conversation, for proposing new topics, and so on.

Sharon joins an ongoing Science Fiction Club discussion

Sharon is a busy third-year psychology student at Virginia Tech. Even though she has a biology exam tomorrow morning, she has been looking forward to her science fiction club meeting for several days, so she decides to go and stay up late to study when she gets back. She remembers that they were planning to talk about Asimov’s Robots and Empire, and she has a new theory about the timeline for first detection of the Zeroth Law.

The meeting is scheduled for 7pm at their usual room in the town library. But she is late getting back from dinner with her room-mate, so she misses her regular bus and arrives 15 minutes late. The meeting is already underway; she notes that they have a relatively small group tonight, but is happy to see Bill and Sara, who are the real experts on Asimov. She is even more delighted to see that these two are already having a heated discussion about the Zeroth Law. But she is cannot immediately tell what points have been made, so she sits back a while to catch the drift of the conversation. At a break, Bill greets her and asks her what she thinks about Faucian’s insight. She replies that she isn’t sure about how central he is to the plot, but that she has a new theory about the timeline. They promise to hear her proposal in a few minutes, then resume the argument.

Table 5: A problem scenario describing Sharon’s visit to the science fiction club meeting.

Face-to-face interaction with club members at a meeting
<ul style="list-style-type: none"> + ensures that both nonverbal and verbal communication contribute to the interaction + leverages many years of experience with communication protocols and conventions – but may introduce distracting or irrelevant personal information about partners – but inhibits parallel communication activities (i.e., among multiple parties at once)
A regular physical space used for club meetings
<ul style="list-style-type: none"> + promotes a feeling of familiarity and intimacy among established members + simplifies the planning and execution process for arriving at meetings – but requires members to travel to the site for interaction – but physical locations are valuable resources that must be shared among organizations

Table 6: Two claims analyzed from the club meeting problem scenario. The feature of interest appears in the shaded area; hypothesized positive consequences are prefaced with a plus sign and negative consequences with a minus sign.

The themes and relationships implicit in a scenario can be made more explicit and open for discussion by analyzing them in claims (Table 6). Problem claims are analyzed by identifying features of the scenario that have notable consequences for the actors’ experience. This is an instance of analytic evaluation and as such is clearly guided by the expectations and biases of the evaluator. A more systematic evaluation can be obtained by asking questions of the scenario that are guided by cognitive or motivational theories of human behavior (Carroll & Rosson, 1992; Polson et al., 1992). The first claim captures a key aspect of Sharon’s experience in the scenario—meeting in-person with other club members creates a rich communication bandwidth; both verbal and non-verbal content can be shared, which helps Sharon know when to jump in with her new topic.

Once an activity feature has been associated with a claim, the analysis is extended through “what if” reasoning that explores other possible positive or negative consequences. For instance, what if Sharon arrives in a T-shirt with a distracting political message? What if the argument goes on for so long that Sharon is never able to raise her topic? Scenarios both evoke and support this sort of questioning; the concrete setting of the scenario invites reasoning about partially-overlapping alternatives and outcomes.

An important characteristic of claims analysis is that it considers both positive and negative consequences. During requirements analysis, there is a tendency to focus on the difficulties or concerns of current practice, as apparent in activity breakdowns or contradictions (Bødker, 1992; Kuutti, 1995; Kyng, 1995; Nardi, 1996). In contrast, designers tend to focus on the likely benefits of proposed features. Claims analysis imposes a balanced view of both problems and opportunities. From the perspective of requirements analysis, it ensures that we build on aspects of the current situation that are already working well. From the perspective of design envisionment, it forces us to consider the side-effects or other undesired impacts of changes to a situation.

Problem scenarios and associated claims are the central product of requirements analysis in SBD. Note though that they are *not* requirements in the traditional sense of the term—that is, they are not a specification of features required of a system. Instead they serve as requirement criteria; design solutions are expected to address the positive and negative consequences conveyed by the scenarios and claims. For instance, we will “require” that the online club environment reinforce the advantages of in-person meetings as much as possible, but at the same time address the disadvantages. This is quite different from specifying that it will have a synchronous communication channel. Individual features of the solution will be identified, elaborated, evaluated, revised, or discarded in an extended iterative process.

4.2 Activity Design

Requirements emerge and are refined throughout system development (Brooks, 1995). But at some point a team understands enough about project stakeholders and their needs that they are ready to make specific design proposals. Indeed some projects may be so over-determined that system functions are specified in advance, and requirements analysis consist simply of analyzing user characteristics and preferences. In SBD, the initial step toward specifying a design solution is made by envisioning how current activities might be enhanced or even completely transformed by available technologies. We deliberately minimize attention to the concrete steps of user interaction at this point, emphasizing the basic goals and motivations of the new activities (see also Constantine & Lockwood, 1999).

SBD is *activity-oriented*—we analyze current practice at the level of meaningful activities, and build from this to new activities (Kuutti & Arvonen, 1992). A danger in this is that the designers will focus too much on how goals are pursued in the current situation, and on understanding and responding to people’s current expectations about their tasks and about technology. To encourage consideration of new options and insights, we deliberately expand the “design space” prior to envisioning the new activities. By design space, we mean the array of possible concepts and technologies that might be relevant to the problem

domain, along with some analysis of what these options might bring to the design solution (MacLean, Young, & Moran, 1989; Moran & Carroll, 1996).

Table 7 exemplifies two techniques useful in exploring design alternatives. The upper part of the table shows how different conceptual metaphors evoke contrasting views of stakeholder activities. Metaphors are often used deliberately in user interface design, with the hope that users will recruit them in reasoning by analogy about what a system does or how it works (Carroll & Thomas, 1982; Carroll, Mack, & Kellogg, 1988; Madsen, 1994). Here we emphasize the role of metaphors in helping designers “think outside the box” (Erickson, 1990; VerPlank, 1988). Addressing real world activities and concerns is crucial to effective system design, but it is often metaphoric thinking that promotes the insights of truly creative design.

Activity design features suggested by metaphors for an online meeting	
Reading at the library	Self-paced, individual access to structured information
Hearing a lecture	Large audience; prepared materials; one-way communication
Visiting a museum	Array of artifacts, small groups or individuals examine, discuss
Going to a cocktail party	Friends forming subgroups; social exchange and mingling
Activity design features suggested by information technology for an online meeting	
A hierarchy of Web pages	Mix of text and graphics, category names and links
An email distribution list (listserv)	One-way “push” communication, possibly large audience
A shared whiteboard	Informal sketches
Meeting groupware	Explicit agenda, support for floor control, meeting records

Table 7: Using metaphors and available information technology to reason about activities

An analysis of available information technology provides a complement to the metaphoric thinking. In a sense, the technology provides another set of metaphors for thinking about the activities to be supported, but in this case the analogy is with classes of software and devices that already exist (e.g., Web information systems, email or database packages). At the same time, a technology-oriented analysis such as this directs the design back to many of the pragmatic concerns of software development, by enumerating possible technology and how it might contribute to the solution. This analysis may also be very influenced by the projects starting assumptions (Table 3), for instance if the development organization already has developed a shared whiteboard or has considerable experience with Web information systems.

The exploration of metaphors and technology does not generate a new activity design. Rather it provides a set of lenses for discussion. The team might consider what it would be like if the online science fiction club was designed to be like a cocktail party versus a lecture; they can argue about the relative advantages of using a structured

groupware framework versus an open-ended Web site. These divergent discussions form a backdrop for the convergent process of scenario writing.

The generation of activity scenarios is a creative process influenced by many factors. The problem scenario provides a starting point—a realistic context and set of goals for meetings among club members. The associated claims motivate some basic design moves—the general heuristic is to maintain or even enhance the positive consequences for the actors while minimizing or removing the negatives (Carroll & Rosson, 1992). The metaphors and technology exploration provide solution ideas—how our concept of meetings or discussions might be transformed and how existing technology might support these activities. Of course, the designers' knowledge of human-computer interaction and of interactive system design broadly speaking also provide important guidance—for example, knowing the relative affordances of different computer-mediated communication channels, understanding the motivational challenges of discretionary-use software, and so on.

Two contrasting scenarios for Sharon's interaction with her club interaction in Table 8. Both activities address the goals of the actors in the problem scenario—joining a club discussion and introducing a new topic of personal interest. Both respond to claims analyzed for the problem scenario, for example attempting to maintain the familiarity and intimacy of a real world meeting in a club, while making participation more flexible, and enabling parallel conversations. However, they address these goals and concerns with rather different views of what constitutes an online discussion and what network-based technology might be used to support it.

The first example was influenced by the lecture and library metaphors. Contributions to the online club materials are automatically distributed to group members; this is analogous to sitting in a room and listening to a lecture. The online material is organized into topical categories that can be browsed in a self-paced fashion, just as Sharon might browse stacks of books in a library. These metaphors are easily supported by a combination of email and Web pages.

A. Sharon visits the Science Fiction Club online forum

Sharon is a third-year psychology student at Virginia Tech, and after three years she has learned to take advantage of her free time between classes. In her hour between her morning classes, she stops by the computer lab to visit the science fiction club, because she heard from a friend that they are discussing her favorite book, Asimov's *Robots and Empire*, and she wants to share her new theory about the timeline for the Zeroth Law.

When she logs onto a computer, she first checks her email, and sees that as she hoped there are several emails from club members proposing and responding to views on this book. But rather than read each email, she follows the convenient link to the Club's Web site, which takes her right to the ongoing discussion. As always, the reviews are first, then the discussion topics, where she finds the new discussion thread started by Sara and Bill. She reads the new thread before adding her theory about the Zeroth Law, and notes that Bill is also fascinated by this piece of the story. She summarizes her theory, and because she wants the group to focus on this issue she makes it a first-level topic but links it to Bill's post to acknowledge the relation. When she submits it she is reminded that an email has been sent to the club listserv with her contribution.

Before leaving, Sharon backs up to the homepage and browses the book categories to look for new books and discussions. Right underneath her favorite category of "Artificial Intelligence" (where the Asimov series is placed), she discovers an intriguing new entry, "Brain Evolution". She doesn't recognize any of the authors in this category, so sends herself a reminder to track down a couple of books from the category later that day.

B. Sharon goes to the Science Fiction Club's online room

Sharon is a busy third-year psychology student at Virginia Tech. But after three years she has learned to take advantage of her free time between classes. In her hour between her morning classes, she stops by the computer lab to visit the science fiction club because she heard from her friend that they are discussing her favorite book, Asimov's *Robots and Empire*, and she wants to share her new theory about the timeline for the Zeroth Law.

When she logs onto a computer, she first checks her email, and sees that as she hoped there are several emails from club members proposing and responding to views on this book. But rather than read each email, she follows the convenient link to the Club's online room. She is taken to their regular discussion spot, the bar of a local pub. As she arrives, she sees that Sara, Bill, and Jennifer are already there. She reviews their conversation, and notes that they are discussing Jennifer's new review of Asimov's *Robots and Empire*. Before she joins in, she quickly opens and browses Jennifer's review. She agrees with Jennifer, so she eagerly jumps in to take her side against Bill and Sara. In a few minutes, the chat moves on to plan a group outing that night. She has to study, so she drops out of the conversation to create a new discussion with her theory about the Zeroth Law. She sees that an announcement is sent to all the club members when she has finished creating the object.

Sara keeps an eye on the others' conversation, and when there is a break, she invites them to visit her new topic. They discuss the Zeroth Law for a while, but leave it open for others to visit. On her way out, Bill tells her he has a new "Brain Evolution" grouping he is working on. She hasn't heard of the titles he mentions, so she sends herself a reminder to track down a couple of books from the category later that day.

Table 8: Two alternative activity designs for the online club meeting

The second scenario shows an influence of the cocktail party, museum, and library metaphors. It emphasizes social exchange and informal conversation, as well as responses to an assortment of club-specific artifacts in the space. Sharon is able to see which of her fellow members are around, and can follow the conversation but also carry out her own exploration in parallel. She jumps in and out of the conversation as her interest in the topic increases or decreases. The club members are engaged in activities that refer to artifacts on display in their room—discussion topics and a bookshelf that displays book titles categorized by theme.

Both of these scenarios have attractive consequences for Sharon and her friends. Both are possible solution approaches, so how do the designers choose? Again, many pragmatic factors contribute to this decision—the kinds of software currently in use, the team’s design expertise, development resources, organizational priorities, and so on. But assuming that both solutions are genuine possibilities, the designers must also evaluate them with respect to their usage implications. One way to do this is with participatory design sessions (Carroll et al., 2000; Chin, Rosson, & Carroll, 1997; Muller, 1992; Kyng, 1995) that focus on how well the alternatives suit stakeholders’ needs. In SBD we also evaluate scenarios through claims analysis, where the positive and negative implications of design features are considered through “what if” discussions.

A sample claim analyzed from each scenario appears in Table 9. In this illustrative example, the scenarios were intentionally written to be very similar in many respects; the claims capture one of the basic design contrasts built into the alternative designs. The Web site offers a convenient hierarchical listing of topics, whereas the online room holds a number of different “objects” that people discuss in real time. The analysis helps the designers to see the relative advantages and disadvantages of an organized asynchronous interaction and a more ad hoc synchronous exchange. Such an analysis may or may not be enough to mandate one alternative over the other. But at the least it begins to lay out usage issues that will be the topic of continuing design.

Discussion archives organized by topic and content submitters
+ leverages people’s familiarity with categorical hierarchies
+ emphasizes the central and permanent recognition of individuals’ contributions to the archive
– but browsing extensive stored archives may be tedious or complex
– but people may be disinclined to contribute more transient and informal content to an organized archive
Real-time conversation organized by the people present in a space
+ leverages people’s familiarity with real world conversational strategies
+ encourages a combination of topic-specific and ad hoc informal exchange
– but requires that conversation participants be present at the same time
– but newcomers may find it hard to interrupt an ongoing conversation

Table 9: Activity claims that help to contrast the implications of competing scenarios.

It is important to note how much progress can be made even at this very early level of envisioning activities. The narratives in Table 8 are quite concrete and evocative; designers or their clients can readily understand what is being proposed and begin to consider the relative pros and cons of the design ideas. Yet the scenarios are “just talk”; indeed if they are shared and discussed over an interactive medium, they can easily be extended or revised as part of a real time design review and discussion.

4.3 Information Design

As design continues, tentative decisions are made about the design direction. For this continuing example, we will elaborate the second of the two activity scenarios in Table 9. We have opted for the “online room” concept because it strongly reinforces important strengths of current real world practice (e.g., familiarity, intimacy, well-learned conversation skills; see Table 6), while at the same time addressing some of the problems identified in requirements analysis (e.g., the need to travel to a specific place, the possible distraction of irrelevant personal information). We now begin to elaborate the underlying activity with information and interaction details—the user interface.

As for activity design, we first explore the design space with metaphors and technology options. The metaphors applied at to information design may even overlap with those used in activity design, but with the emphasis shifted to what the users will see and understand about the system; the concerns of this phase are similar to those in the “gulf of evaluation” discussed by Norman (1986). For instance, the three metaphors of library, museum, and cocktail party suggest these information design ideas:

- *library*: documents look like pages in books with title-bearing covers; objects are arranged in alphabetical or category order on shelves; there are desks and chairs for browsing and note-taking
- *museum*: the space is broken into relatively small topic-specific rooms; objects of interest are mounted on the walls; there is space around each object enabling a group of interested parties to form; descriptive titles and text are attached to each display object
- *cocktail party*: there are a number of attractive “seating areas”, perhaps including a table and chairs; visitors are organized into groups and emit a “buzz” of conversation; new arrivals are greeted with waves or smiles

Technology options explored at this phase might include hyperlinks (icons or other controls are used to navigate or access content objects), MUDs (multi-user domains based on a spatial model), other graphical techniques for rendering a room (a static photograph, a panorama, a three-dimensional model), as well as techniques for representing the people and objects in the space (avatars, buddy lists, texture-mapped forms and objects).

Ideas such as these are discussed and combined with the design team’s experience in information design. Information design possibilities are “tried out” in the activity design scenarios, with attention directed toward claims analyzed in earlier phases. Does the information design further promote intimacy among club members? Will newcomers feel welcome and will they be able to “catch up” and participate? Does the design allow for parallel activity? Design inquiry such as this—and the scenarios that provide a real world context for the reflective process—is a hallmark of scenario-based design.

A. Sharon goes to the Science Fiction Club’s room in the community MOO

<Sharon’s background and goal to share her Zeroth Law theory>

When she logs onto a computer, she first checks her email in-box, and sees several emails marked with a dot indicating that they are new; a quick read of the senders confirms that they are from club members proposing and responding to views on this book. She opens the first one, knowing that it will contain a convenient link to the Club’s online room. She is taken to their regular discussion spot, and she skims the familiar description of the bar. She grins to see a new seating option someone has added, a snailshell-toadstool combination, and seats herself at this spot; she is told that she is “reclining in a luxurious curl” and ready to join in the activities. She also notes a new exit leading to a “Fractal Immersion Room”, and makes a mental note to visit later. Finally, the welcoming text stream concludes with its usual status report, informing her that Sara, Bill, and Jennifer are in the pub, that Jennifer has just added a new review to the bookshelf, and that this review is currently in the possession of Bill.

Text messages from her friends begin appearing, including a quick interleaved hello from Bill, before he comments on a point Jennifer made in her review. Sharon thinks Bill might be mistaken, but before joining in, she asks Bill if she can pick up the review so that she too can read Jennifer’s comments. She finds the issue Bill is debating and sees that she agrees with Jennifer, so she eagerly jumps in to take her side against Bill and Sara. In a few minutes, the chat moves on to plan a group outing that night. She has to study, so she drops out of the conversation to create a new discussion with her theory about the Zeroth Law. After she issues the commands to instantiate the discussion and then types in a provocative starting premise, the system reports to the group that a new discussion has been added, and that an email announcement has been sent to the club mailing list.

<The discussion of Sharon’s proposed new topic, her resolution to read the “Brain Evolution” titles>

B. Sharon goes to the Science Fiction Club’s room in the collaborative environment

<Sharon’s background and goal to share her Zeroth Law theory>

When she logs onto a computer, she first checks her email in-box, and sees several emails marked with a dot indicating that they are new; a quick read of the senders confirms that they are from club members proposing and responding to views on this book. She opens the first one, knowing that it will contain a convenient link to the Club’s online room. She is taken to their regular discussion spot, and she sees the familiar panoramic image of Eastenders Pub, with the mirror and bar prints on the wall, the wooden brass-trimmed bar, and the red canvas bar stools. Miniature images of her friends Bill, Sara, and Jennifer are also there, positioned in a close group at one end of the bar. On the club’s special bookshelf, she sees all of the reviews and discussions contributed recently, organized as usual by name of author. As usual, the reviews appear as simple text documents, while discussions appear as indented lists. One review on a middle shelf is highlighted in yellow, telling her that this is new since she last visited. She guesses that this review may be what the others are discussing in the chat area, so she opens it to see what all the excitement is about. When she does, she can see that the other three also have the review open, in fact she can see from their named pointers exactly the passage that they are discussing. She finds that she agrees with Jennifer in this case, and eagerly jumps in to take her side in the argument.

In a few minutes, the chat moves on to plan a group outing that night. She has to study, so she drops out of the conversation to create a new discussion with her theory about the Zeroth Law. She uses the room toolbox to create a new discussion object, indicates that this concerns Asimov’s *Robots and Empire*, and adds a provocative opening premise about the Zeroth Law. When finished, the discussion object is positioned automatically on the top shelf and given a bright yellow color. She is also provided feedback indicating that an announcement has been sent to the club mailing list.

<The discussion of Sharon’s proposed new topic, her resolution to read the “Brain Evolution” titles>

Table 10: Alternative information designs for the online club meeting scenario

Two alternative information designs are presented in Table 10. For simplicity just the central actions making up the discussion have been elaborated. Both scenarios offer a view into a virtual room that contains club members and documents. Both assume an

information model that is spatial, applying the concepts of rooms, furniture, and so on as a pervasive metaphor. Both also borrow from the other more specific metaphors described above—for example there is a bookshelf that organizes reviews and discussions, there are notices and other artifacts posted on the walls, the participants form groupings and are in conversation. However, the technology supporting each scenario is quite different, in one case, consisting of a traditional text-based MUD, and in the second, providing a graphical rendition of the underlying spatial model.

Either of the information designs could be used to represent an online club space, but the two proposals have rather different implications for how club members are likely to experience the space. In the first case, all attention will be directed at a sometimes complex stream of descriptive text. The experience is rather like reading a book or a play, with different people and objects providing the content, but much of the mental experience under the control of the reader. In contrast, the graphical view offers a concrete rendition of the space, and the attention of the participant is instead directed toward a specific activity, in this case a shared discussion of a new review.

Textual descriptions of people and objects present in an online space
<ul style="list-style-type: none"> + focus participants' attention on a single source of information about the situation and events + leverages club members' experience and enjoyment with fantasy-producing textual imagery – but the interleaving many sorts of descriptions and communications may become quite complex – but it may be impossible to integrate individual text-based fantasies into a coherent mental model
Two-dimensional visual depictions of people and objects present in an online space
<ul style="list-style-type: none"> + leverages club members' familiarity and habits with real world places and objects + enables parallel processing of spatially-distributed information + allows participants to use spatial cues in organizing activities (e.g., position in review, location of chat area) – but club members may feel restricted by the constraints of a two-dimensional space – but objects or people distributed in the space may distract from the activity in focus

Table 11: Claims contrasting usage implications of the alternative information designs

These general implications are captured in the claims presented in Table 11. As for the earlier design claims, these arguments do not mandate one choice over another, rather they provoke discussion of each alternative's pros and cons. In this illustrative example we elect to pursue the graphical view rather than the text-based view, largely because it simplifies the comprehension and participation process. However, we note an important negative consequence, that the “real world” view of the pub architecture and contents may dampen the creativity or fantasy of members' contributions. This may become an issue as design continues, for example we may search for ways to suggest a hybrid approach, inviting both real world and fantasy content (Cherny, 1995).

Information design comprises all aspects of how the task information is organized and rendered during users' activities. In most design projects, this will include paying special attention to the needs of new or inexperienced users. For instance, suppose that this was Sharon's first visit to the online club. How would she know that the bookshelf held

recent reviews and discussions, that she could open them? In the graphical case, the visual cues provided, along with general experience in using graphical user interfaces, might be enough to cue the behavior described in the scenario. But would she know to use the “toolbox” to create a new discussion object? It is often useful to create “documentation design” versions of other design scenarios, where the hypothetical actor(s) are assumed to have little or no experience using the system. In SBD, the design of supporting documentation (help texts or other guidance) is also inspired by appropriate metaphors (e.g., a coach, a policeman) and technology (e.g., online tutorials, animated demonstrations). More detailed examples of scenario-based documentation design can be found in Rosson and Carroll (2001).

4.4 Interaction Design

User interaction design becomes even more detailed when the concrete exchanges between the user(s) and the system are specified. Of course, to some extent this has been foreshadowed by decisions already made—a graphical array of people and objects implies some way of indicating or selecting among them; a text-based chat record implies a mechanism for character or word input. We have also noted several cases of system feedback, a critical element of any interaction design. But many details are still open, and the scenarios serve as a usage context for considering interaction options.

As in other aspects of design reasoning, metaphors can be used to inspire interaction design. For instance in thinking about how people navigate to the science fiction club space (i.e., the Eastenders Pub), we contrasted the metaphor of walking down a street and through a door, with that of using a map to point at and access a spatial location directly. In the first case, the metaphor emphasizes the three-dimensionality of the real world, and opens up the possibility that other people or objects might be encountered “along the way”. The second emphasizes a more two-dimensional view of the world, but leverages people’s familiarity with maps, as well as relieving the possible tedium of step-based navigation. In this case, we decided that the convenience and familiarity of map-based access was more desirable than the suggestion of three-dimensionality in the underlying model. But we left as an open issue the possibility of adding other information to the map, that would enable and prompt opportunistic encounters and exchanges.

With respect to our continuing example, we have simplified it even more than in the discussion of information design, addressing a single question of user interaction design—how should club members access and interact with the review and discussion objects (and presumably any other objects that are available for use). This is a central interaction that will be repeated many times, so it warrants special attention. The abbreviated scenarios in Table 12 show the influence of two familiar technologies for interaction with graphical objects—direct manipulation (double-click to open) versus a less direct command-oriented manipulation (select and apply a menu command).

A. Sharon goes to the Science Fiction Club’s room in the collaborative environment
<p><Sharon’s background and goal to share her Zeroth Law theory></p> <p><Sharon arrives in the bar, sees that her friends are talking about a new review></p> <p>Sharon wants to open the review to see what all the excitement is about. She moves her mouse pointer to the bright yellow icon and clicks twice. A separate window entitled “Jennifer’s Review” opens to the side, covering the other icons on the bookshelf. Sharon is automatically positioned at the same location as Sara in the review text; she knows from experience that this means that Sara made the last comment in the chat area. This irritates her for a minute, because she wanted to see what Bill was talking about, not Sara, but she quickly finds out where Bill is positioned via his colored rectangle in the scroll bar, and moves to share his view.</p> <p><Sharon participates in the argument, then creates her new discussion object></p> <p><The discussion of Sharon’s proposed new topic, her resolution to read the “Brain Evolution” titles></p>
B. Sharon goes to the Science Fiction Club’s room in the collaborative environment
<p><Sharon’s background and goal to share her Zeroth Law theory></p> <p><Sharon arrives in the bar, sees that her friends are talking about a new review></p> <p>Sharon wants to open the review to see what all the excitement is about. She moves her mouse pointer to the bright yellow icon and clicks the right button to bring up the menu. She recognizes the list of review-specific action choices, and selects the second item (join) rather than the first one (browse). A separate window opens to the side, covering the other icons on the bookshelf. Sharon is automatically positioned at the same location as Sara in the review text; she knows from experience that this means that Sara made the last comment in the chat area. This irritates her for a minute, because she wanted to see what Bill was talking about, not Sara, but she quickly finds out where Bill is positioned via his colored rectangle in the scroll bar, and moves to share his view.</p> <p><Sharon participates in the argument, then creates her new discussion object></p> <p><The discussion of Sharon’s proposed new topic, her resolution to read the “Brain Evolution” titles></p>

Table 12: Alternative interaction designs for opening the review as part of the online club meeting scenario.

The two scenarios convey some of the tradeoffs in deciding whether to provide menu-based interaction with task objects. The support of direct manipulation builds on the pervasive use of such interaction techniques in modern interactive software. But the what-if reasoning applied to this scenario pointed to some usage situations that are not well-supported by this simple technique. What if Sharon did not know the others and did not want to join in on the discussion? What if she wanted to get a quick (private) sense of the issues were before joining the group? A command-based interaction (i.e., through a customizable menu system) offers more possibilities for user control, but at the cost of adding more steps to the detailed interaction (see Table 13). Developing and analyzing these alternatives helped us as designers to think through the relative benefits of ease of execution versus user control and flexibility.

Double-clicking to open the object represented by a visual icon
+ leverages users' general experience with graphical user interfaces + promotes a feeling of direct interaction with the object represented by the icon – but the semantics of the double-click may be hidden, and vary as a function of the object that is opened
Selecting and then requesting a menu to open the object represented by a visual icon
+ makes explicit the command that is being addressed to the object represented by the icon + creates an opportunity to choose among multiple object-appropriate actions – but the selection-then-choice operation may seem tedious for frequent actions – but the menu list of options must be perceived and interpreted and slow down the interaction – but selecting and addressing a command creates a level of indirection (thus distance) in goal mapping

Table 13: Claims that capture some of the tradeoffs associated with alternate interaction techniques

Note that this concrete and specific interaction design detail has broad implications for what users are able to do with this system. Direct manipulation techniques are simple, familiar, and pervasive, but can limit functionality: different science club documents may implement different meanings of “open”, but they will not offer any other options for interaction. In contrast, a menu system is flexible and extensible, even to the extent of admitting new kinds of function for club artifacts not yet invented. We recognized these issues by exploring this scenario and its variants—during claims analysis we considered scenario variants in which other object-specific functions could be useful, ultimately leading us to choose the more complex menu-based interaction technique. This example demonstrates how even small details can have important consequences for the *activities* being envisioned and supported; in SBD, the scenario context ensures a continuous focus on activities, even during detailed interaction design.

In the science fiction club example, we have considered and incorporated standard user interaction technology—the familiar WIMP paradigm of windows, icons, menus, and pointing. However, SBD can also be used to envision and analyze the implications of more novel user interaction paradigms and devices. For example, we might consider a role for intelligent agents as part of a new user scenario, and contrast this to a scenario involving community-generated FAQ (frequently asked questions) repository. Or we could explore the implications of gesture or speech recognition in lieu of (or as a complement to) conventional keyboard and mouse input devices. A key advantage of exploring these ideas within a scenario context is that designers are less likely to be caught up in the new technologies for their own sake; the method leads them to try out their new ideas in usage situations that at the least are believable, and that are analyzed explicitly with respect to usability consequences. More detailed examples of SBD activities focused on emerging interaction paradigms are discussed in Rosson & Carroll (2001b).

4.5 Usability Evaluation

In SBD, we assume that usability evaluation takes place early and throughout the design and development process. Any representation of a design can be evaluated. Figure 1 emphasizes a phase of evaluation that takes place after detailed user interaction scenarios

have been developed. This is relatively formal usability testing—users are recruited to carry out representative tasks on early operational prototypes. Such evaluation does require sufficient progress on a design to enable construction of a prototype, though of course such prototypes may be built using special-purpose languages or tools, or may even be a low-fidelity prototype constructed of cardboard and paper (Muller, 1991; Virzi, Sokolov, & Karis, 1996).

Early in design, user feedback may be obtained in rather informal settings, for example a participatory design session (Chin, Rosson, & Carroll, 1997; Muller, 1992). Users can be included in discussion and envisionment of activity scenarios. The design ideas are also subjected constantly to analytic evaluation through claims analysis and other design review activities (e.g., usability inspections or cognitive walkthrough; Nielsen, 1995; Nielsen & Mack, 1994; Polson et al., 1992). All of these activities yield formative evaluation feedback that guides changes and expansion of the design vision.

SBD implements Scriven's (1967) concept of *mediated evaluation*. In mediated evaluation, empirical data are collected (Scriven calls this "pay-off" evaluation), but the materials and methods used in the empirical test are guided by prior analytic evaluation. The analytic evaluation may have many different components, for example an expert-based inspection, or perhaps a cognitive model constructed for a particularly complex or critical interaction sequence (Gray, John & Atwood, 1992; Kieras, 1997). In SBD, the primary method for analytic evaluation is claims analysis. The claims written during scenario generation and discussion analyze the usability issues most likely to influence the system's success or failure; they are used as a skeleton for constructing and administering empirical usability tests. One way to view a claims analysis is as a series of usability hypotheses that can be assessed empirically; claims also help to explain *why* a design feature has an observed impact on users' experiences.

Scenarios and their associated claims are combined to create *usability specifications* (Carroll & Rosson, 1985; Good et al., 1986). A usability specification is a representative task context that has been analyzed into critical subtasks, with each subtask assigned target usability outcomes. In SBD, the design scenarios provide a realistic task context, and the associated claims provide an analysis of the scenario that is parameterized with expected or desired usability outcomes. When the prototype has sufficient functionality that it can be tested with representative users, the specified tasks are tested and the results compared to the target outcomes.

A sample usability specification developed from the science fiction club scenario appears in Table 14. The scenario serves as a usability specification in two ways. Early (or at any point) in design, representative users may be asked to simply explore a rough prototype while pursuing the open-ended goals stated in the task context. Because the actual experience of users would vary considerably in this case, it does not make sense to establish performance outcomes; indeed these tests are likely to include instructions to think aloud, so as to provide as much feedback as possible about the user experience (Mack, Lewis, & Carroll, 1983). Nonetheless, the system is considered successful in these cases if it satisfies the goals of a user enacting the scenario; collecting general ease of use or satisfaction ratings at the end of each episode can provide a measure of this.

Task context: Sharon is a busy university student and a regular member of the science fiction club. During a few free minutes she sees from her email that new discussions have begun at their online club room. She joins them, planning to share her new idea about her favorite Asimov novel when there is a break in the conversation .

Overall scenario outcome:

Average rating of at least 4.0 (of 5) on *ease of use* and *satisfaction*

	Performance Targets	Satisfaction Targets
Subtask 1: Navigate to the online club room	20 seconds, 0 errors	4 on <i>convenience</i>
Subtask 2: Identify present club members	5 seconds, 1 error	4.5 on <i>presence</i>
Subtask 3: Identify and open Jennifer’s review	10 seconds, 0 errors	4 on <i>directness</i>
Subtask 4: Locate and join Bill in review	15 seconds, 1 error	0.5 on <i>confusion</i>
Subtask 5: Create new discussion object	60 seconds, 1 error	4.5 on <i>feedback</i>

Table 14: Usability specification developed from the science fiction club scenario

When the prototype is robust enough to measure subtask times, more detailed usability specifications guide empirical testing. In the example, a set of five simple subtasks has been analyzed from the user interaction scenario. These tasks are directly related to claims that have been developed for key design features (only some of these claims have been documented here). Performance measures are established, based either on the designers’ own (expert) experiences with the prototype, or on benchmark data collected from comparable systems. Satisfaction measures are constructed to assess one or more of the specific concerns raised in the claims. For example, a negative consequence of menu-based interaction is that it may reduce feelings of directness. The usability specification tracks this issue by requiring that users’ perception of this quality be at an acceptably high level (as operationalized by a Likert-type rating scale with a range of 1-5). The satisfaction qualities specified for the other subtasks were similarly derived from advantages or disadvantages hypothesized by claims.

Usability specifications developed in this way have two important roles in evaluation. First, they provide concrete usability objectives that can be serve as a management tool in system development—if a product team accepts these targets, then the team’s usability engineers are able to insist that redesign and improvement continue until they are met (Carroll & Rosson, 1985; Good et al., 1986).

Second, the specifications tie the results of empirical evaluation directly to the usability issues raised during design. For instance, our interaction design scenario specified that Sara determines Bill’s position from his colored rectangle in the scroll bar (see Table 12). A positive consequence is that awareness of others’ and their activities is enhanced; a negative consequence is that the display becomes more crowded. The time it takes to locate Bill is specified as one measure of the feature’s impact, but this performance target is complemented by users’ subjective reactions to the feature. For example users might indicate level of agreement to a statement such as: “The scroll bar with rectangles indicating others’ positions was confusing”. Problematic results with respect to either of these usability targets raise specific issues for redesign.

As development continues, more complete prototypes or initial working systems are constructed. At this point, new usage scenarios are introduced, with new actors, goals, and activity contexts. These scenarios are also subjected to claims analysis, and the resulting claims used to develop additional usability testing materials. This is an important step in the evaluation, because it ensures that the design has not been optimized for the set of design scenarios that has guided development thus far. Near the end of the development cycle (or at pre-defined milestones), a carefully orchestrated *summative* evaluation is carried out, with the goal of assessing how well the system performs with respect to its usability specifications.

4.6 Scenario-Based Iterative Design

The SBD framework described in this chapter is highly iterative. Although we emphasize the grounding of new activities in a thorough analysis of current practice, we assume that many new possibilities will not be realized until design ideas are made concrete and exposed to actual use. At times entirely new activities may be envisioned, when breakdowns in the technical or social environment demand a radical transformation of current practice. Even in these cases, an SBD process is valuable—problem scenarios help designers predict how stakeholders will need to evolve their goals and expectations if they are to learn and adopt the new activities (Orlikowski, 1992).

More commonly, the development process will involve a more gradual co-evolution of computing technology and the users' activities, where the technology raises new possibilities for action, and people's creative use of the technology in turn creates new requirements for support (Carroll, Kellogg, & Rosson, 1991). In these cases, user interaction scenarios provide a central activity-centered thread in iterative design, serving as intermediate design products that can be generated, shared, and revised, as part of an overarching envisionment-evaluation-refinement cycle.

5. Scenarios throughout the System Life Cycle

The SBD framework is aimed at the iterative development of activities that people may pursue with computing support. Our example has focused on the central processes of requirements analysis, design, and usability evaluation, but one of the great strengths of scenario-based methods is that they support a diverse range of system development goals (Carroll, 1997; see Figure 2). Product planners present day-in-the-life scenarios to managers as design visions (Dubberly & Mitsch, 1992); requirements engineers gather workplace scenarios through direct observation and interviews, and analyze scenarios as primary data (Antón, McCracken & Potts, 1994; Holbrook, 1990; Hsia et al., 1994; Kaindl, 1997; Kuutti, 1995; Potts, 1995).

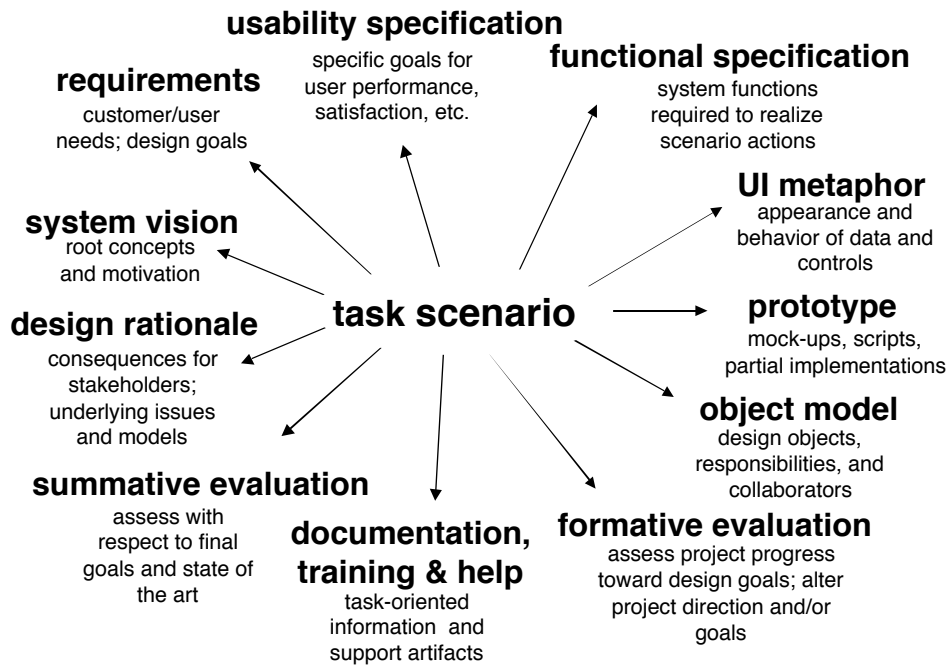


Figure 2: Scenarios have diverse uses throughout system development life cycle

Even if scenarios are not developed and transformed as described in the SBD framework, they may be used at many points along the way. For instance, task-based user documentation is often structured by scenarios. Minimalist help and training provide many examples of this, such as a “training wheels” system that blocks functions that are not relevant to a paradigmatic novice-use scenario (Carroll & Carrithers, 1983), or a “view matcher” that guides new programmers through a predefined scenario of debugging and modification (Carroll, Singer, Bellamy, & Alpert, 1990; Carroll & Rosson, 1991; Rosson & Carroll, 1996).

Usage scenarios have also come to play a central role in object-oriented software engineering (Jacobson, 1995; Jacobson, Booch, & Rumbaugh, 1998; Rubin & Goldberg, 1992; Wirfs-Brock et al, 1990). A *use case* is a scenario written from a functional point of view, enumerating all of the possible user actions and system reactions that are required to meet a proposed system function (Jacobson, et al. 1992). Use cases can then be analyzed with respect to their requirements for system objects and interrelationships. Wirfs-Brock (1995) describes a variant of use case analysis in which she develops a “user-system conversation”: using a two-column format, a scenario is decomposed into a linear sequence of inputs from the user and the corresponding processing and/or output generated by the system. Kaindl (2000) extends this analysis by annotating how scenario steps implement required user goals or system functions.

Scenarios are promising as a mediating representation for analyzing interactions between human-centered and software-centered object-oriented design issues (Rosson & Carroll, 1993; 1995). As we have seen, scenarios can be decomposed with respect to the software objects needed to support the narrated user interactions. These software objects

can then be further analyzed with respect to their system responsibilities, identifying the information or services that should be contributed by each computational entity (Wirfs-Brock & Wilkerson, 1989; Beck et al., 1989; Rosson & Gold, 1989). This analysis (often termed *responsibility-driven design*, Wirfs-Brock et al., 1990) may lead to new ideas about system functionality, for example initiatives or actions taken by a software object on behalf of the user or another object. Scenarios and claims analysis are useful in describing these new ideas and considering their usability implications *in the context of use* (Rosson, 1999; Rosson & Carroll, 1995; Rosson & Carroll, 2001a). For example, a calendar object may be given the responsibility to notify club members of upcoming events; this helps to keep the members informed, but individuals may find the reminders annoying if they are very frequent, or they may come to rely on them too much.

The general accessibility of scenarios makes them an excellent medium for raising and discussing a variety of competing concerns. Software engineers are concerned about issues such as code reuse, programming language or platform, and so on; management is concerned with project resources, scheduling, and so on; a marketing team focuses on issues such as the existing customer base and the product cost. These diverse concerns are *nonfunctional requirements* on system development—concerns about “how” a system will be developed, fielded, and maintained rather than “what” a system will provide (Sommerville, 1992). Usability goals are often specified as nonfunctional requirements, in that they typically focus on the quality of the system rather than its core functions (Mylopoulos, Chung, & Nixon, 1992). The low cost of development, content flexibility, and natural language format of scenarios and claims, make them excellent candidates for contrasting and discussing a range of such issues throughout the software development life cycle (Rosson & Carroll, 2000; Sutcliffe & Minocha, 1998).

6. Current Challenges

When we design interactive systems, we make use. We create possibilities for learning, work, and leisure, for interaction and information. Scenarios—descriptions of meaningful usage episodes—are popular representational tools for making use. They help designers to understand and to create computer systems and applications as artifacts of human activity, as things to learn from, as tools to use in one’s work, as media for interacting with other people.

Scenario-based design offers significant and unique leverage on some of the most characteristic and vexing challenges of design work: Scenarios are at once concrete and flexible, helping developers manage the fluidity of design situations. Scenarios emphasize the context of work in the real world; this ensures that design ideas are constantly evaluated in the context of real world activities, minimizing the risk of introducing features that satisfy other external constraints. The work-oriented character of scenarios also promotes work-oriented communication among stakeholders, helping to make design activities more accessible to many sources of expertise. Finally, scenarios are evocative, raising questions at many levels, not only about the needs of the people in a scenario as written, but also about variants illustrating design tradeoffs.

Scenario-based methods are not a panacea. A project team who complains “We wrote scenarios, but our system still stinks!” must also report how their scenarios were

developed, who reviewed them, and what roles they played in system development. If a user interaction scenario is not grounded in what is known about human cognition, social behavior, and work practices, it may well be inspiring and evocative, but it may mislead the team into building the wrong system (Carroll et al., 1998). Scenarios are not a substitute for hard work.

At the same time, *any* work on user interaction scenarios directs a project team to the needs and concerns of the people who will use a system. It is in this sense that scenarios can provide a very lightweight approach to human-centered design. Simply writing down and discussing a few key expectations about users' goals and experiences will enhance a shared vision of the problems and opportunities facing system users. Adopting a more systematic framework such as described here adds control and organization to the creative process of design, and at the same time generates work products (scenarios and claims) that can serve as enduring design rationale during system maintenance and evolution (McKerlie & MacLean, 1994; Moran & Carroll, 1996).

Where are scenarios taking us? The current state of the art in the design of interactive systems is fragmented. Scenarios are used for particular purposes throughout system development, but there is no comprehensive process (Carroll, 1995; Jarke, Bui, & Carroll, 1998; Weidenhaupt et al., 1998). Scenario practices have emerged piecemeal, as local innovations, leading to a considerable variety of scenario types specialized for particular purposes (Campbell, 1992; Young & Barnard, 1987). A detailed textual narrative of observed workplace practices and interactions, a use case analysis of an object-oriented domain model, a day-in-the-life video envisionment of a future product, and the instructions for test subjects in an evaluation experiment could *all* be scenarios. Recognizing this, and cross-leveraging the many different views of scenarios, is a potential strength of scenario-based design. But much work remains in developing overarching frameworks and methods that exploit this potential advantage.

It is important for us to be ambitious, skeptical, and analytic about scenarios and scenario-based design. Forty years ago, Herman Kahn (1962) expressed puzzlement that scenarios were not more widely used in strategic planning. In the 1990s, scenarios have become so pervasive in interactive system design that younger designers may wonder what the alternative is to scenario-based design! But there is yet some strangeness to scenarios. We are not much farther than Kahn was in understanding how scenarios work as tools for planning and design, or in understanding how to fully exploit their unique strengths as aides to thought.

References

- Ackoff RL, (1979) "Resurrecting the future of operations research," *Journal of the Operations Research Society*, 30(3), pp. 189-199.
- Antón A, McCracken WM, and Potts C, (1994) "Goal decomposition and scenario analysis in business process reengineering," *Proceedings of CAiSE'94: Sixth Conference on Advanced Information Systems Engineering* (Utrecht, the Netherlands, June 6-10), Springer-Verlag.
- Beck K, (1999) *Extreme Programming Explained: Embrace Change*. Reading MA: Addison-Wesley. Beyer H, and Holtzblatt K, (1998) *Contextual design: Defining customer-centered systems*, Morgan Kaufmann, San Francisco.

- Beck K, and Cunningham W, (1989) "A laboratory for teaching object-oriented thinking," in N. Meyrowitz (Ed.), *Proceedings of Object-Oriented Systems, Languages and Applications: OOPSLA '89* (pp. 1-6), ACM, New York.
- Beyer, H and Holtzblatt K, (1998) *Contextual Design: A Customer-Centered Approach to System Design*. San Francisco, CA: Morgan Kaufmann.
- Bødker S, (1991) *Through the Interface: A Human Activity Approach to User Interface Design*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Brooks F, (1995) *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, Reading, Massachusetts (Anniversary Edition, originally 1975).
- Campbell RL, (1992) "Will the real scenario please stand up?" *SIGCHI Bulletin*, 24(2), 6-8.
- Carroll JM (ed.), (1995) *Scenario-based design: Envisioning work and technology in system development*, John Wiley and Sons, New York.
- Carroll, J.M. 1997.. Scenario-based design. In M. Helander & T.K. Landauer (Eds.) *Handbook of Human-Computer Interaction, Second Edition*. Amsterdam: North Holland, pp. 383-406.
- Carroll JM, (2000) *Making use: Scenario-based design of human-computer interactions*, MIT Press, Cambridge, Massachusetts.
- Carroll, J.M. & Carrithers, C. 1983. Blocking errors in a learning environment. Psychonomic Society 24th Annual Meeting (San Diego, CA, November 17). Abstract in Proceedings, page 356.
- Carroll JM, Chin G., Rosson MB, and Neale, DC. (2000) "The development of cooperation: Five years of participatory design in the Virtual School," *Proceedings of DIS 2000: Designing Interactive Systems* (Brooklyn, NY, 17-19 August). New York, ACM pp. 239-251.
- Carroll, J. M., Karat, J., Alpert, S. A., van Deusen, M., & Rosson, M. B. 1994. Demonstrating Raison d'Être: Multimedia design history and rationale. In C. Plaisant (Ed.), *CHI'94 Conference Companion* (pp. 29-30). New York: ACM.
- Carroll, J. M., Kellogg, W. A., & Rosson, M. B. 1991. The task-artifact cycle. In J. M. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge, UK: Cambridge University Press (pp. 74-102).
- Carroll, J.M., Mack, R.L., & Kellogg, W.A. 1988. Interface metaphors and user interface design. In M. Helander (Ed.) *Handbook of Human-Computer Interaction* (pp. 67-85). Amsterdam: North Holland.
- Carroll JM, and Rosson MB, (1985) "Usability specifications as a tool in iterative development," *Advances in Human-Computer Interaction*, HR Hartson, ed., Ablex, Norwood, New Jersey.
- Carroll JM, and Rosson MB, (1990) "Human-computer interaction scenarios as a design representation," *Proceedings of the 23rd Annual Hawaii International Conference on Systems Sciences* (Kailua-Kona, HI, January 2-5), IEEE Computer society Press, Los Alamitos, California, pages 555-561.
- Carroll JM, and Rosson MB, (1991) "Deliberated evolution: Stalking the View Matcher in design space," *Human-Computer Interaction*, 6, pp. 281-318.
- Carroll JM, and Rosson MB, (1992) "Getting around the task-artifact cycle: How to make claims and design by scenario," *ACM Transactions on Information Systems*, 10, pp. 181-212.
- Carroll, J. M., Rosson, M. B., Chin, G., & Koenemann, J. 1998. Requirements development in scenario-based design. *IEEE Transactions on Software Engineering*, 24(12: December), 1156-1170.
- Carroll, J.M., Singer, J.A., Bellamy, R.K.E., & Alpert, S.R. 1990. A View Matcher for Learning Smalltalk. In J.C. Chew & J. Whiteside (Eds.) *Proceedings of CHI90: Human Factors in Computing Systems* (Seattle, WA, April 1-5). New York: ACM, pp. 431-437.

- Carroll, J. M. & Thomas, J. C. 1982. Metaphors and the cognitive representation of computing systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 12(2), 107-116.
- Checkland, P. B. 1981. *Systems Thinking, Systems Practice*. Chichester: John Wiley.
- Chin, G., Rosson, M. B. and Carroll, J. M. 1997. Participatory Analysis: Shared development of requirements from scenarios. In *Proceedings of Human Factors in Computing Systems, CHI'97 Conference* (pp. 162-169). New York: ACM.
- Constantine LL, and Lockwood LAD, (1999) *Software for use: A practical guide to the models and methods of usage-centered design*. Addison-Wesley, Reading, Massachusetts.
- Cooper, A. 1999. *The Inmates are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, IN: SAMS Press.
- Cross, N. (2001) "Design cognition: Results from protocol and other empirical studies of design activity." In C. Eastman, M. McCracken & W. Newstetter (eds.), *Design knowing and Learning: Cognition in Design Education*. Amsterdam: Elsevier, pages 79-103.
- Cherny L, (1995) "Mud Community," *The Mud Register: Conversational Modes of Action in a Text-Based Virtual Reality* (Chapter, 3; pp. 42-126). Unpublished PhD dissertation, Stanford University, Palo Alto, CA.
- Dubberly H, and Mitsch D, (1992) "Knowledge navigator," *CHI'92 Special Video Program: Conference on Human Factors in Computing Systems* (Monterey, CA, May 3-7), BA Myers, ed., ACM SIGCHI, New York.
- Erickson, T. 1990. Working with interface metaphors. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 65-73). Reading, MA: Addison-Wesley.
- Erickson T, (1995) "Notes on design practice: Stories and prototypes as catalysts for communication," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 37-58.
- Freud S, (1900) *The Interpretation of Dreams, Standard Edition, Vol. IV*, Hogarth, London.
- Good, M., Spine, T.M., Whiteside, J., & George, P. 1986. User-derived impact analysis as a tool for usability engineering. In M. Mantei & P. Oberton (Eds.), *Proceedings of Human Factors in Computing Systems: CHI '86* (pp. 241-246). New York: ACM.
- Gray, W.D., John, B.E., & Atwood, M.E. 1992. The precis of Project Ernestine, or an overview of a validation of GOMS. In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), *Proceedings of Human Factors in Computing Systems: CHI '92* (pp. 307-312). New York: ACM.
- Haviland SE, and Clark HH, (1974) "What's new? Acquiring new information as a process in comprehension," *Journal of Verbal Learning and Verbal Behavior*, 13, 512-521.
- Holbrook H, (1990) "A scenario-based methodology for conducting requirements elicitation," *ACM SIGSOFT Software Engineering Notes*, 15-1, January, pp. 95-103.
- Hsia P, Samuel J, Gao J, Kung D, Toyoshima Y, and Chen C, (1994) "Formal approach to scenario analysis," *IEEE Software*, March, pp. 33-41.
- Jacobson I, (1995) "The use-case construct in object-oriented software engineering," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 309-336.
- Jacobson I, Booch G, and Rumbaugh J, (1998) *The Unified Software Development Process*, Addison-Wesley, Reading, Massachusetts.

- Jacobson, I., Christersson, M., Jonsson, P., & Övergaard, G., *Object-Oriented Software Engineering—A Use-Case Driven Approach*. Reading, MA, Addison-Wesley, 1992.
- Jarke M, Bui XT, and Carroll JM, (1998) "Scenario management: An interdisciplinary approach," *Requirements Engineering*, 3(3-4), 155-173.
- Kahn H, (1962) *Thinking about the Unthinkable*, Horizon Press, New York.
- Kahneman D, and Tversky A, (1972) "Subjective probability: A judgement of representativeness.," *Cognitive Psychology*, 3, 430-454.
- Kaindl H, (1997) "A practical approach to combining requirements definition and object-oriented analysis," *Annals of Software Engineering*, 3, 319-343.
- Kaindl H, (2000) "A design process based on a model combining scenarios with goals and functions," *IEEE Transactions on Systems, Man, and Cybernetics*, 30(5), 537-551.
- Karat J, (1995) "Scenario use in the design of a speech recognition system," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 109-133.
- Karat J, and Bennett JB, (1991) Using scenarios in design meetings – A case study example," *Taking design seriously: Practical techniques for human-computer interaction design*, J Karat, ed., Academic Press, Boston, pp. 63-94.
- Kieras, D. 1997. A guide to GOMS model usability evaluation using NGOMSL. In M.G. Helander, T.K. Landauer, & P.V. Pradhu (Eds.), *Handbook of Human-Computer Interaction, Second Edition* (pp. 733-766). Amsterdam: North-Holland.
- Kuutti K, (1995) "Work processes: Scenarios as a preliminary vocabulary," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 19-36.
- Kuutti, K. & Arvonen, T. 1992. Identifying potential CSCW applications by means of Activity Theory concepts: A case example. In J. Turner & R. Kraut (Eds.), *Proceedings of Computer-Supported Cooperative Work: CSCW '92* (pp. 233-240). New York: ACM.
- Kyng M, (1995) "Creating contexts for design," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 85-107.
- Lévi-Strauss C, (1967) *Structural anthropology*, Anchor Books, Garden City, New York.
- Mack, R.L., Lewis, C.H., & Carroll, J.M. 1983. Learning to use office systems: Problems and prospects. *ACM Transactions on Office Information Systems*, 1, 254-271.
- MacLean A, Young RM, and Moran TP, (1989) "Design rationale: The argument behind the artifact," *Proceedings of Human Factors in Computing Systems: CHI '89*, pp. 247-252, ACM, New York.
- Madsen, K.H. 1994. A guide to metaphorical design. *Communications of the ACM*, 37(12), 57-62.
- McKerlie D, and MacLean A, (1994) "Reasoning with design rationale: Practical experience with Design Space Analysis," *Design Studies*, 15, 214-226.
- Medin DL, and Schaffer MM, (1978) "A context theory of classification learning," *Psychological Review*, 85, 207-238.
- Moran, T. & Carroll, J.M. (Editors). 1996. *Design Rationale: Concepts, Techniques, and Use*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Muller, M.K. 1992. Retrospective on a year of participatory design using the PICTIVE technique. In A. Janda, (Ed.), *Proceedings of Human Factors of Computing Systems: CHI '92* (pp. 455-462).

- Muller MJ, Tudor LG, Wildman DM, White EA, Root RA, Dayton T, Carr R, Diekmann B, and Dykstra-Erickson, E, (1995) "Bifocal tools for scenarios and representations in participatory activities with users," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 135-163.
- Mylopoulos J, Chung L, and Nixon B, (1992) "Representing and using nonfunctional requirements: A process-oriented approach," *IEEE Transactions on Software Engineering*, 18(6), 483-497.
- Nardi, B.A. (Editor). 1996. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge, MA: MIT Press.
- Nielsen J, (1995) Scenarios in discount usability engineering," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 59-83.
- Nielsen, J. & Mack, R.L. 1994. *Usability Inspection Methods*. New York: John Wiley & Sons.
- Norman, D. A. 1986. Cognitive engineering. In D.A. Norman & S.D. Draper (Eds.), *User Centered System Design* (pp. 31-61). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Norman DA, (1988) *The Psychology of Everyday Things*. New York: Basic Books.
- Orlikowski WJ, (1992) "Learning from Notes: Organizational issues in groupware implementation," in J. Turner & R. Kraut (Eds.), *CSCW'92: Proceedings of the Conference on Computer Supported Cooperative Work* (pp. 362-369). New York: ACM.
- Polson PG., Lewis C, Rieman J, and Wharton C, (1992) "Cognitive walkthroughs: A method for theory-based evaluation of user interfaces," *International Journal of Man-Machine Studies*, 36, 741-773.
- Potts C, (1995) "Using schematic scenarios to understand user needs," *Proceedings of ACM Symposium on Designing Interactive Systems: DIS'95*: (Ann Arbor, Michigan), ACM Press, New York, pp. 247-256.
- Propp V, (1958) *Morphology of the folktale*, Mouton, The Hague (originally published in 1928).
- Rosch E, Mervis CB, Gray W, Johnson D, and Boyes-Braem P, (1976) "Basic objects in natural categories," *Cognitive Psychology*, 7, 573-605.
- Rosson, M. B. 1999. Integrating development of task and object models. *Communications of the ACM*, 42(1: January), 49-56.
- Rosson, M. B., & Carroll, J. M. 1993. Extending the task-artifact framework. In R. Hartson, & D. Hix, (Eds.), *Advances in Human-Computer Interaction, Vol. 4*. New York: Ablex (pp. 31-57).
- Rosson MB, and Carroll JM, (1995) "Narrowing the gap between specification and implementation in object-oriented development," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 247-278.
- Rosson, M.B. & Carroll, J.M. 1996.. The reuse of uses in Smalltalk programming. *ACM Transactions on Computer-Human Interaction*, 3(3), 219-253.
- Rosson, M.B. and Carroll, J.M. 2000. Nonfunctional requirements in scenario-based development. *Proceedings of OZCHI 2000* (pp. 232-239), North Ryde, NSW, Australia, CSIRO Mathematical and Information Sciences.
- Rosson, M.B. & Carroll, J.M. 2001a. Scenarios, objects, and points-of-view in user interface design. M. van Harmelen (Ed.), *Object Modeling and User Interface Design*, London: Addison-Wesley Longman.
- Rosson, MB and Carroll JM (2001b) *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. San Francisco: Morgan Kaufmann.

- Rosson, M. B., & Gold, E. 1989. Problem-solution mapping in object-oriented design. In N.Meyrowitz (Ed.), *Proceedings of OOPSLA'89: Conference on Object-Oriented Programming Systems, Languages, and Applications*. (pp. 7-10). New York: ACM.
- Rosson MB, Maass S, and & Kellogg WA, (1988) "The designer as user: Building requirements for design tools from design practice," *Communications of the ACM*, 31, 1288-1298.
- Rubin K & Goldberg A, (1992) "Object behavior analysis," *Communications of the ACM*, 35, 9, 48-62.
- Schön DA, (1967) *Technology and change: The new Heraclitus*, Pergamon Press, New York.
- Schön DA, (1983) *The reflective practitioner: How professionals think in action*, Basic Books, New York
- Schön DA, (1987) *Educating the reflective practitioner*, Jossey-Bass, San Francisco.
- Scriven M, (1967) "The methodology of evaluation," *Perspectives of curriculum evaluation*, R Tyler, R Gagne, and M Scriven, eds., Rand McNally, Chicago, pp. 39-83.
- Sommerville, I. 1992. *Software Engineering* (Fourth Edition). Reading, MA: Addison-Wesley.
- Sutcliffe, AG and Minocha S, (1998) "Scenario-based analysis of non-functional requirements," *Proceedings of RESFQ 2000*, Pisa, Italy, June 8th-9th, 1998.
- Tversky A, and Kahneman D, (1974) "Judgements under uncertainty: Heuristics and biases," *Science*, 185, 1124-1131.
- Verplank, W.L. 1988. Graphic challenges in designing object-oriented user interfaces. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 365-376). Amsterdam: North-Holland.
- Virzi, R.A. Sokolov, J.L., & Karis, D. 1996. Usability problem identification using both low- and high-fidelity prototypes. In *Proceedings of Human Factors in Computing Systems: CHI '96* (pp.236-243). New York: ACM.
- Wasserman, A.I. & Shewmake, D.T. 1982. Rapid prototyping of interactive information Systems, *ACM Software Engineering Notes*, 7(5), 171-180.
- Weidenhaupt K, Pohl K, Jarke M, and Haumer P, (1998) "Scenarios in system development: Current practice," *IEEE Software*, 15/2, 34-45.
- Wertheimer M, (1938) "Laws of organization in perceptual forms," *A Sourcebook of Gestalt Psychology*, WD Ellis, ed., Paul, Trench, Trubner, London.
- Wirfs-Brock R, (1995) "Designing objects and their interactions: A brief look at responsibility-driven design," *Scenario-based design: Envisioning work and technology in system development*, JM Carroll, ed., John Wiley & Sons, New York, pp. 337-360.
- Wirfs-Brock R and Wilkerson B, (1989) "Object-oriented design: A responsibility-driven approach," N. Meyrowitz (Ed.), *Object-Oriented Programming: Systems, Languages and Applications*, *Proceedings of OOPSLA'89*. New York, ACM, 1989, pp71-76.
- Wirfs-Brock R, Wilkerson B and Wiener, L, (1990) *Designing Object-Oriented Software*. Englewood Cliffs, N.J, Prentice Hall.
- Young RM, and Barnard PB, (1987) "The use of scenarios in human-computer interaction research; Turbocharging the tortoise of cumulative science," *Proceedings of CHI+GI'87: Conference on Human Factors in Computing Systems and Graphics Interface* (Toronto, Canada, April 5-9), ACM, New York, pp. 291-296.