Intent inference techniques and applications at MERL

Neal Lesh, Charles Rich, Candace L. Sidner

TR2001-48 November 2001

Abstract

This paper overviews some research at MERL related to the task ofintentions of groups of people engaged in collaborative activities.

AAAI Fall Symposium, MA

Publication History:–

1. First printing, TR2001-48, February 2002

Intent inference techniques and applications at MERL

Neal Lesh, Charles Rich, and Candace L. Sidner
{lesh,rich,sidner}@merl.com

(MERL) — Mitsubishi Electric Research Laboratories, Cambridge, MA 02139

Introduction

This paper overviews some research at MERL related to the task of inferring the intentions of people interacting with computers and/or each other. First, we discuss techniques developed as part of the Collagen project for inferring the intentions of individuals engaged in one-on-one collaboration. We then describe two projects, Human-Guided Search and the Personal Digital Historian, which support face-to-face group collaboration. These two projects currently do not employ, but could benefit from, techniques for inferring the intentions of groups of people engaged in collaborative activities.

Plan Recognition in Collagen

Participants in a collaboration derive benefit by pooling their talents and resources to achieve common goals. However, collaboration also has its costs. When people collaborate, they must usually communicate and expend mental effort to ensure that their actions are coordinated. In particular, each participant must maintain some sort of mental model of the status of the collaborative tasks and the conversation about them.

We have shown how to use plan recognition algorithms to help maintain a model of what is mutually believed by both participants in a one-on-one collaboration (Lesh, Rich, & Sidner 1999, Lesh, Rich, & Sidner 2001). This work takes place in the context of a larger effort to apply principles of human collaboration to human-computer interaction using the interface agent metaphor. Specifically, we have built an application-independent collaboration manager, called Collagen (Rich & Sidner 1998a), based on the SharedPlan theory of task-oriented collaborative discourse (Grosz & Kraus 1996).

Figure 1 shows shows a screen image and sample interaction for four interface agents built by us and our collaborators in four different application domains using Collagen. We have also built agents for air travel planning (Rich & Sidner 1998) and email (Gruen *et al.* 1999). All of these agents are currently research prototypes.

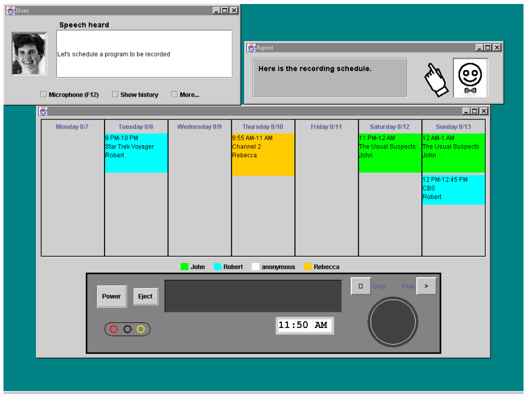
Plan recognition is the process of inferring intentions from actions. Plan recognition has often been proposed for improving user interfaces or to facilitate intelligent help features. Typically, the computer watches "over the shoulder" of the user and jumps in with advice or assistance when it thinks it has enough information. In contrast, our main motivation for adding plan recognition to Collagen was to reduce the amount of communication required to maintain a mutual understanding between the user and the agent of their shared plans in a collaborative setting. Without plan recognition, Collagen's discourse interpretation algorithm onerously required the user to announce each goal before performing a primitive action which contributed to it.

Although plan recognition is a well-known feature of human collaboration, it has proven difficult to incorporate into practical computer systems due to its inherent intractability in the general case. We exploit three properties of the collaborative setting in order to make our use of plan recognition tractable. The first property is the focus of attention, which limits the search required for possible plans.

The second property of collaboration we exploit is the interleaving of developing, communicating about and executing plans, which means that our plan recognizer typically operates only on partially elaborated hierarchical plans. Unlike the "classical" definition of plan recognition, which requires reasoning over complete and correct plans, our recognizer is only required to incrementally extend a given plan.

Third, in a collaboration, it is quite natural to ask for clarification, either because of inherent ambiguity, or simply because the computation required to understand an action is beyond a participant's abilities. We use clarification to ensure that the number of actions the recognizer must interpret will always be small.

Figure 2 illustrates roughly how plan recognition works in Collagen. Suppose the user performs action k. Given the root plan (e.g., A) for the current discourse purpose (e.g., B) and a set of recipes, the plan recognizer determines the set of minimal extensions to the plan which are consistent with the recipes and include the user performing k. If there is exactly one such extension, the extended plan becomes part of the new discourse state. If there is more than one possible extension, action k is held and reinterpreted along with the next event, which may disambiguate the interpretation (which l does not), and so on. The next event may in fact be a clarification.



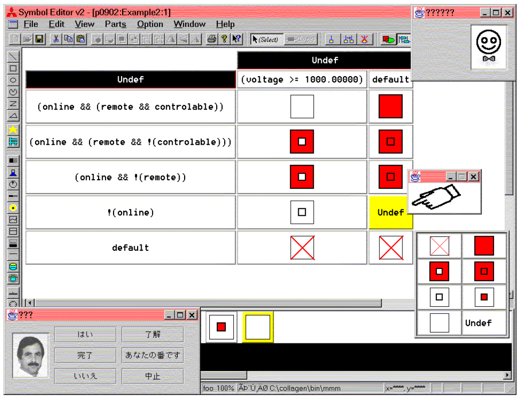
Let's schedule a program to be recorded

Here is the recording schedule.

Monday 8/7	Tuesday 8/8	Wednesday 8/9	Thursday 8/10	Friday 8/11	Saturday 8/12	Sunday 8/13
	9 PM-10 PM Star Trek Voyager Robert		9:55 AM-11 AM Channel 2 Rebecca		11 PM-12 AM The Usual Suspects John	12 AM-1 AM The Usual Suspects John 12 PM-12:45 PM CBS Robert

1 User says "Hello."
2 Agent says "Hello, what's your name?"
3 User says "Candy."
4 Agent says "Hello, Candy!"
5 User says "Let's set the clock."
6 Agent says "What is the time?"
7 User says "The time is 11:00 AM."
8 Agent sets the clock to 11:00 AM.
...
9 User says "How do I play back a video from my camcorder?"
10 Agent says "First, plug in the camcorder cables."
11 User says "How do I plug in the camcorder cables?"
12 Agent points to where you plug in the camcorder cables.
13 Agent says "The red cable plugs into the red terminal, the black cable plugs into the black terminal, and the yellow cable plugs into the yellow terminal. Make sure the other ends of the cables plug into the corresponding colors on the camcorder."
14 User says "What next?"
15 Agent says "Now press Play on your camcorder. You should see a picture on the TV screen."
16 User says "Ok."
17 Agent says "I don't see a picture. Make sure the camcorder is plugged in or has fresh batteries, and is turned on."
18 User says "Ok."
19 Agent says "Good. It's working."

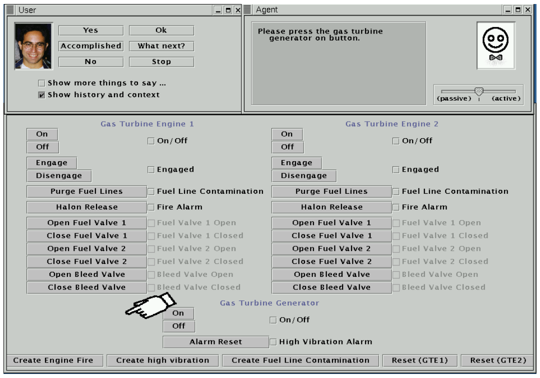
(a) VCR Agent.



Undef	(voltage >= 1000.00000)	default
(online && (remote && controlable))	□	■
(online && (remote && !(controlable)))	▣	▣
(online && !(remote))	▣	▣
!(online)	□	Undef
default	☒	☒

1 Agent says "Please list the other (non-boolean) attributes used in the columns of the logic table."
2 User says "The attributes are: voltage."
3 Agent creates a new user-defined attribute.
4 Agent enters "voltage" as the name of the new attribute.
5 Agent says "Please select the data type of the new attribute."
6 User selects float as the data type of the new attribute.
7 User clicks ok to accept the definition of voltage.
8 Agent says "Please edit the initial setting for the voltage attribute (if necessary) and then click ok."
9 User clicks ok to accept 0.0 as the initial setting for voltage.
10 Agent shows the expression editor for column 1.
11 Agent enters (voltage > 0.0) as the title for column 1.
12 Agent says "Please define an expression for column 1."
...[user goes to lunch]...
14 User says "Where are we?"
15 Agent says "You are defining an expression for column 1, as part of making a logic table."
16 User enters (voltage >= 1000.0) as the title for column 1.
17 Agent shows the logical operator window.
18 Agent selects the >= operator.
19 Agent drags and drops >= into expression for column 1.
20 Agent selects the voltage attribute.
21 Agent drags and drops voltage into expression for column 1.
22 Agent types "1000.0" into expression for column 1.

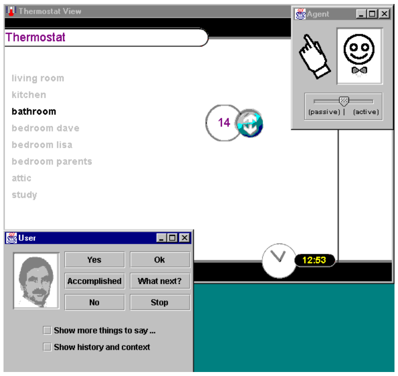
(b) Symbol Editor Agent.



Please press the gas turbine generator on button.

1 Agent says "Let's start the generator."
2 User says "Ok."
3 Agent says "You take it from here."
4 User presses the alarm reset button.
5 User says "What next?"
6 Agent says "Please press the gas turbine generator on button."
7 Agent points to the gas turbine generator on button.
8 User presses the gas turbine generator button.
9 Agent says "Good. We have successfully started the generator."

(c) Gas Turbine Agent.



1 Agent says "Let's set the temperature in a room."
2 User selects the bathroom.
3 User says "What next?"
4 Agent says "What should the temperature setting be?"
5 User says "18 degrees Centigrade."
6 Agent sets the temperature in the bedroom to 18 degrees.

(d) Thermostat Agent.

Figure 1. Example Agents Built with Collagen.

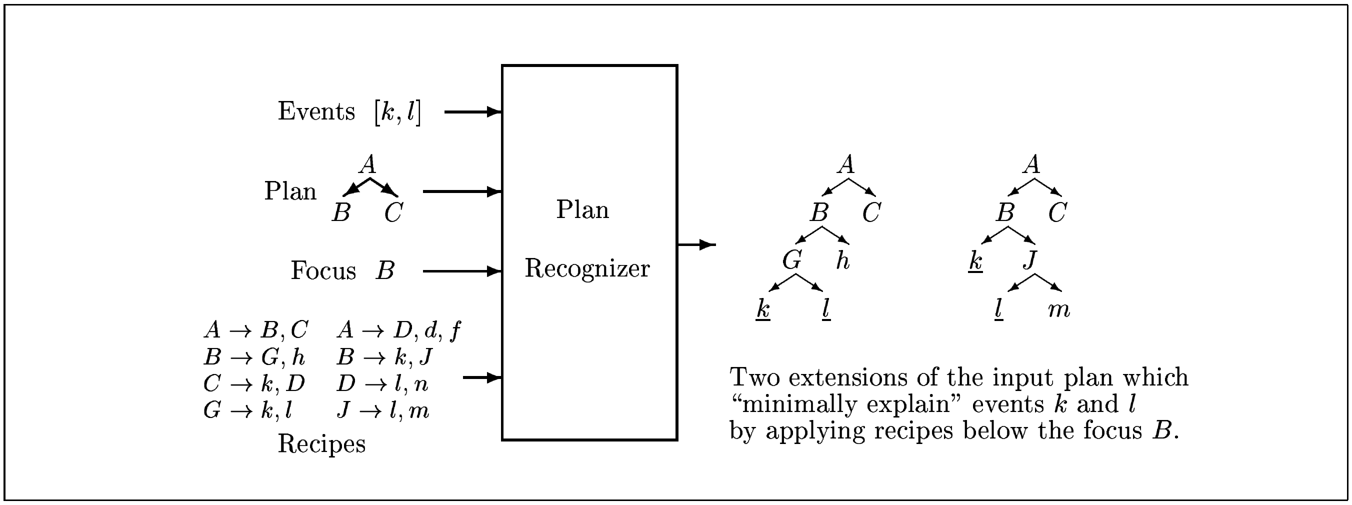

Events $[k, l]$
Plan
A
B C
Focus B
$A \to B, C$ $A \to D, d, f$
$B \to G, h$ $B \to k, J$
$C \to k, D$ $D \to l, n$
$G \to k, l$ $J \to l, m$
Recipes
Plan Recognizer
A
B C
G h
$\underline{k}$ $\underline{l}$
A
B C
$\underline{k}$ J
$\underline{l}$ m
Two extensions of the input plan which "minimally explain" events k and l by applying recipes below the focus B.

Figure 2: Example Plan Recognizer Inputs and Outputs in a Collaborative Setting.

Currently, Collagen is designed for collaboration between one user and one computer agent. We have, of course, thought about extending Collagen to handle multi-person collaborations. Our focus would be to develop techniques for maintaining a model of what the computer agent believes is mutually believed by all the participants, which might include intentions held by only some (but known by all) of the participants. However, although the SharedPlan theory is formalized for the multi-person case (Grosz & Kraus 1996), no algorithms have been proposed for interpreting and generating discourse under this theory for multiple people. Furthermore, very little empirical data has been collected and analyzed about how to model the conversations that occur when groups of people collaborate.

Human-Guided Search

Most previous research on scheduling, routing, and layout problems has focused on developing fully automatic algorithms. There are, however, at least two reasons for developing cooperative, interactive systems for optimization problems like these. First, human users may have knowledge of various amorphous real-word constraints and objectives that are not represented in the objective function given to computer algorithms. In vehicle-routing problems, for example, human experts may know the flexibility of certain customers, or the variability of certain routes. The second reason to involve people in the optimization process is to leverage their abilities in areas in which humans (currently) outperform computers, such as visual perception, learning from experience, and strategic assessment.

We have been exploring a new cooperative paradigm for optimization, Human-Guided Simple Search (HuGSS) (Anderson *et al.* 2000). We have developed prototype systems for interactively solving vehicle routing problems, graph partitioning problems, and jobshop scheduling problems. In our current framework, the computer performs a simple, hill-climbing search. One or more people interactively "steer" the search process by repeatedly performing the following three actions:

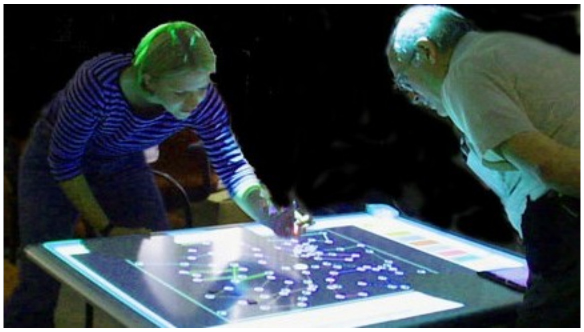


Figure 3: The Optimization Table.

1. Edit the current solution.
2. Invoke a focused local search, starting from the current solution. The user controls which moves are considered, how they are evaluated, and what type of search is used.
3. Revert to an earlier solution, or to an initial seed solution generated randomly prior to the session.

For our initial implementation we have used a tabletop display, which we call the *Optimization Table* (see Figure 3). We project an image down onto a whiteboard. This allows users to annotate candidate solutions by drawing or placing tokens on the board, a very useful feature. In addition, several users can comfortably use the system together. For this kind of problem, creating an effective visualization is an intrinsic challenge in bringing the human into the loop. Figure 4 shows our attempt to convey the spatial, temporal, and capacity-related information needed for the vehicle routing problem.

The central depot is the black circle at the center of the display. The other circles represent customers. The pie slices in the customer circles indicate the time windows during which they are willing to accept

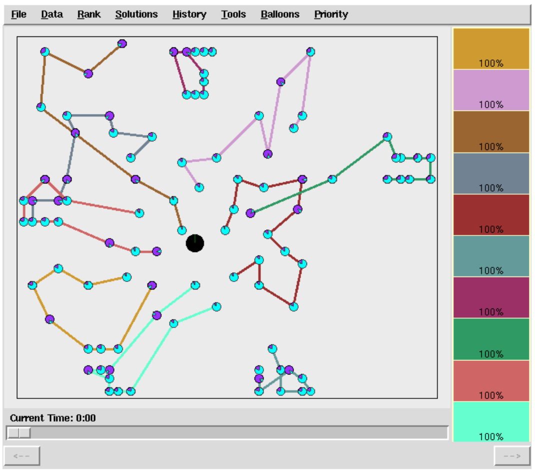


Figure 4: Top view of optimization table.

delivery. The truck routes are shown by polylines, each in a different color. At the user's option, the first and last segments of each route can be hidden, as they are in the figure, to avoid visual clutter around the depot. The search-control operations described in the previous subsection are supported by mouse operations and pull-down menus. Detailed information about individual customers and trucks can also be accessed through standard interface widgets.

We have run over 100 hours of experiments with our interactive vehicle routing system (for the single-user case). Our results show that human guidance greatly improves the performance of our hill-climbing engine to solve vehicle routing problems. Additionally, despite its simplicity, our prototype system is competitive with the majority of previously reported systems on benchmark academic problems, and has the advantage of keeping a human tightly in the loop to handle the complexities of real-world applications.

There are many possible advantages to the computer being able to infer the users' intentions. For example, the history of solutions maintained by the computer could be annotated with the users' intentions. Furthermore, the system could be made more "mixed-initiative", i.e., the computer could make suggestions or more productively use its computation between user-invoked searches.

Personal Digital Historian

We often find ourselves using photos to "re-connect" with people, whether it be with our family who we have not seen all day, a friend or colleague whom we have not seen in a year, or our parents who live across the country. The goal of Personal Digital Historian (PDH) project is to enable informal storytelling using personal digital data such as photos, audio and video in a face-to-face social setting (Shen *et al.* 2001). The project as a whole includes research in the areas of the design of the shared-display devices, on-line authoring, story-listening, user-guided image layout, content-based information retrieval and data mining. The PDH project is in a very early stage; the system we describe has been designed but we are only beginning to implement it.

Most existing software in the area of digital storytelling provides users with either powerful query methods or authoring tools. In the former case, the users can repeatedly query their collections of digital content to retrieve information to show some one. In the latter case, a user experienced in the use of the authoring tool can carefully craft a story out of their digital content to show or send to someone at a later time. Furthermore, current hardware is also lacking. Desktop computers are not suitably designed for group, face-to-face conversation in a social setting, while handheld story-telling devices have limited screen sizes and can be used only by a small number of people at once.

We designed our system to work on a touch-sensitive, circular table-top display, as shown in Figure 5. The layout of the entire table-top display, as shown in Figure 6 consists of a large story-space area encompassing most of the table-top until the perimeter, one or more narrow arched control panels, and an outer rim which can be used to rotate the contents of the table.

The table top can instantiate multiple control panels, e.g., one for each person at the table. When people use the system they will cause different sets of their pictures to be displayed on the table. The pictures will be displayed in a radial-pattern in the story-space, so that pictures initially will be oriented towards the outer rim of the table. Users have the option of moving or re-orienting pictures using the touch-screen, in a rough simulation of how real pictures would slide along a table. By dragging one's hand along the outer rim of the table, the users can spin the entire contents of the table, much like a lazy-susan.

The primary method of navigation is organized about four questions essential to storytelling: who?, when?, where?, and what? Control panels located on the perimeter of the table contain buttons labeled "people", "calendar", "location", and "events", corresponding to these four questions. When a user presses the "location" button, for example, the display on the table changes to show a map of the world. Every picture in the database that is annotated with a location will appear as a tiny thumbnail at its location. The user can pan and zoom in on the map to a region of interest, which increase the size of the thumbnails. Similarly, by pressing one of the other three buttons, the user can cause the pictures to be organized by the time they were taken along a linear straight timeline, the people they contain, or the event-keywords which the pictures were annotated with.

The user can form implicit Boolean queries, or filters, by selecting items or regions in the different navigational views. For example, if someone selects two friends in the people view, then (until this selection is retracted) only pictures containing one or both of these friends will be highlighted in subsequent navigation. If the user next selected "location", for example, then they would see where they have traveled with either of these friends by observing where the highlighted

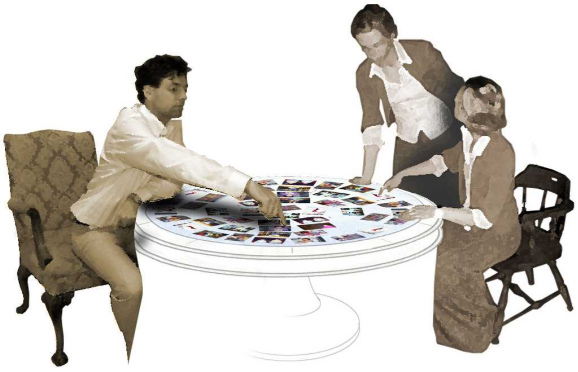


Figure 5: Artistic rendering of PDH in use.

pictures appear on the map. If the user selected "calendar" instead, they would see when they have taken pictures with these friends. Another non-menu driven query metaphor used in PDH is invoked when the user presses and holds down a particular picture. The system then offers the user the ability to display pictures taken at a similar time, or a similar place, or with the same people, or at the same event as the selected picture

Additionally, our system can project a slowly moving stream of pictures, which are somehow related to the users' recent requests even if not specially matching the current query, along the perimeter of the table. The simplest form of this can simply be to randomly show photos that were taken at similar locations or times as the other photos that are being displayed. As another example, when looking at a picture of one's grandparents, the PDH might bring out images and passages that related to historical events around the time the pictures were taken.

As with Human-Guided Search, the PDH system could benefit from the ability to infer the intentions of the users. The ability to track the focus of attention seems very useful to support informal storytelling and to help disambiguate queries from the user. Furthermore, a better understanding of the context, or goals of the storyteller, could improve the ability of the system to suggest associated pictures, layout pictures requested by the user, and even summarize sets of pictures.

References

Anderson, D.; Anderson, E.; Lesh, N.; Marks, J.; Mirtich, B.; Ratajczak, D.; and Ryall, K. 2000. Human-guided simple search. In *Proc. 17th Nat. Conf. AI*, 209–216.

Grosz, B. J., and Kraus, S. 1996. Collaborative plans for complex group action. *Artificial Intelligence* 86(2):269–357.

Gruen, D.; Sidner, C.; Boettner, C.; and Rich, C. 1999. A collaborative assistant for email. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems.*

Lesh, N.; Rich, C.; and Sidner, C. L. 1999. Using plan recognition in human-computer collaboration. In *Proc. of 7th Int. Conf. User Modeling*, 23–32.

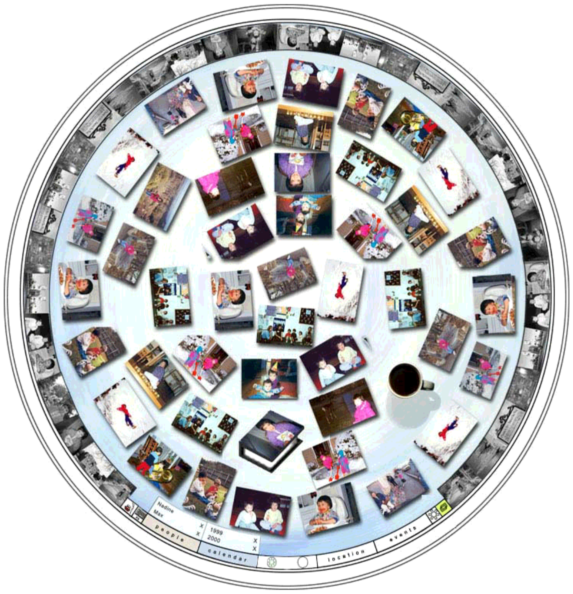


Figure 6: Top view of PDH.

Lesh, N.; Rich, C.; and Sidner, C. L. 2001. Collaborating with focused and unfocused users under imperfect communication. In *Proc. of 8th Int. Conf. User Modeling.*

Rich, C., and Sidner, C. 1998a. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction* 8(3/4):315–350.

Rich, C., and Sidner, C. 1998b. Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction* 8(3/4):315–350. Reprinted in S. Haller, S. McRoy and A. Kobsa, editors, *Computational Models of Mixed-Initiative Interaction*, Kluwer Academic, Norwell, MA, 1999, pp. 149–184.

Shen, C.; Lesh, N.; Moghaddam, B.; Peardsley, P.; and Bardsley, R. S. 2001. Personal digital historian: User interface design. In *CHI 2001, Design Expo (Extended Abstract).* Extended Abstract.