Groupware for Planning

R. Alterman[1], S. Landsman[1], A. Feinman[1], J. Introne[1], S. Kirschenbaum[2].

[1]Computer Science Department
Brandeis University
Waltham, MA 02454 USA
(781) 736-2703
{alterman,seth,afeinman,jintrone}
@cs.brandeis.edu

[2] NUWC Division
Navy Research LAB
1176 Howell Street
Code 2211, Building 1171/1
Newport, RI 02841-5047
(401) 832-3835
kirschenbaumss@csd.npt.nuwc.navy.mil

Technical Report CS-98-200

ABSTRACT
This paper develops a design of groupware for distributed planning. Users of the system work together at the same time but in different locations. The first part of the design is system support for joint planning. A set of representations and operations that facilitate users' reasoning about *common ground* (Clark, 1996) and *shared plans* (Grosz & Sidner, 1990; Grosz & Kraus, 1996) are developed. The second part of the design adds adaptive components to the system that aid the development of conventions for coordinating behavior. These reduce user work in future planning sessions.

Keywords
Groupware, Distributed Planning, Adaptive System, Convention

INTRODUCTION

Groupware is defined as a computer-based system that supports two or more users and that provides an interface to a shared environment (Ellis, Gibbs, and Rein, 1991; also Schniederman, 1998). Groupware systems can be characterized by the locations (same place/different place) and times (same time/different time) in which users interact. Meeting room technology is an example of same time/ same place (face-to-face) interaction; electronic mail is an example of different time/different place (asynchronous distributed) interaction. Our interest is in same time/ different place (synchronous distributed) interaction. Previous efforts in this area of research have, for example, included group editors (Ellis, Gibbs, and Rein, 1991), shared workspaces for drawing (Greenberg, Roseman, Webster, and Bohnet, 1992), and air traffic control systems (Wiener and Nagel, 1988). The focus of the work described in this paper is on groupware for planning given the same time/different place condition.

Suppose that there are multiple users at remote locations who must plan out and execute, in real time, a coordinated set of operations. Each of the users has different capabilities and duties. Their only resource for coordinating behavior is the computer. At issue is how to build a system that supports their on-line cooperative activity. An assumption is that a community of users have an ongoing practice in using the system to support their continued effort at planning in their domain.

The first part of the design for interaction is concerned with system support for joint planning; here we will draw on prior work on groupware, the work of Clark (1996) on *joint activity* and *common ground*, and the work of Grosz & Sidner (1990) and Grosz & Kraus (1996) on *shared plans* and *collaboration.* Common ground is the knowledge, beliefs, and suppositions available to participants during the course of a joint activity. It provides a basis for participants to reason about the actions of other participants. Shared plans are a special case of common ground and are generated as a method of coordinating behavior. The design task is to develop a set of representations and operations that facilitate users' reasoning about common ground and shared plans.

The second part of the design adds adaptive components to the system. It will be necessary to identify a regularity of behavior in the joint activity of the group that can be converted into an improvement in performance. In the everyday world *conventions* are regularities of behavior; they are a community of actors' solution to a recurrent problem of coordination (Lewis, 1969). Conventions emerge from group practice; their development improves performance, while decreasing communication and planning costs (Garland & Alterman, 1998). The human-computer interaction issue is how to build a groupware system that facilitates the development of convention for coordinating on-line planning behavior within a community of users.

GROUPWARE FOR PLANNED JOINT ACTIVITIES

For a test domain we designed and built the VesselWorld system. In VesselWorld there are three users who must cooperate and coordinate their behavior. VesselWorld is a *groupware* system; users work at remote locations simultaneously. Each user is the captain of a ship. The task is to remove toxic waste from a harbor. Collectively the users must explore a large area to find and move all barrels of toxic waste to a large barge. Two of the users operate cranes that can be used to lift toxic waste from the floor of the sea. The third user is captain of a tugboat that can be used to drag small barges from one place to another. The cranes are able to individually lift and carry small or medium toxic waste barrels, jointly lift large barrels, and jointly lift (but not carry) extra large barrels. The tugboat cannot lift barrels, but can attach to and move small barges. Small barges may hold multiple barrels of various sizes.

Joint Activity

Users of the VesselWorld system are engaged in a *joint activity* (Clark, 1996). Joint activities have participants, who assume public roles. Joint activities advance one increment at a time, mostly through *joint actions*. Joint actions are created when people coordinate with each other.

> In VesselWorld, users have public roles of either tugboat or crane operators. An example goal is the goal for two crane operators to lift a large barrel of toxic waste onto a barge. This task requires a hierarchy of joint activities, such as the tugboat captain tugging a small barge to a given location, and two crane operators coordinating their efforts to lift the toxic waste onto the small barge.

Joint actions have phases. Each phase has entry and exit points, each of which require coordination. The entry and exit boundaries of a joint activity (and its constituent joint action/activities) are jointly engineered by the participants. This is a problem of coordination.

> The small barge must be in position before the toxic waste is loaded; the two crane operators must begin lifting and end lifting at the same times.

Coordination of behavior requires common ground and communication. At any moment during a joint activity, what constitutes common ground has three parts (Clark, 1996: p. 43):

1. initial common ground
2. current state of joint activity
3. public events so far

The initial common ground is the set of background facts, assumptions, and beliefs presupposed at the outset of the joint activity; whatever conventions for coordinating behavior exist are part of the initial common ground. The current state of joint activity is where in the activity the participants presuppose themselves to be. The public events are those events presupposed by the participants as leading up to the current state of affairs. Common ground accumulates among participants during the course of activity.

> The initial common ground between the two crane operators includes knowledge that some barrels of toxic waste require two cranes to lift and that the crane operators must coordinate their lifting (e.g., begin at the same time). The current state of the world must be monitored until both crane operators are in position and prepared to lift the barrel of toxic waste, and until the barge is in position and anchored. The public event of the tug boat captain anchoring the barge signals the readiness of the barge for loading additional barrels onto the barge.

Following Grosz & Sidner (1990) and Grosz & Kraus (1996), we will assume that for activities that involve shared planning, common ground includes a fourth element:

4. a shared plan

The shared plan represents the participants' common understanding of what they plan to do in order to achieve their common goal. It develops during the course of the activity. It includes beliefs about actions that are to be taken and also the intentions of the participants.

> Before the large barrel of toxic waste can be lifted the intentions of the tug boat captain and the two crane operators must be clarified. The plan to lift the large barrel of toxic waste onto the barge will only succeed if the tugboat captain intends to anchor a small barge close to the site of the barrel, and if both crane operators intend to lift the barrel at the same time.

DESIGN FOR PLANNED JOINT ACTIVITY

In VesselWorld there are three mechanisms that help users to accumulate and monitor common ground.

1. the state of world map
2. a chat window
3. a panel of attachments

The **s**tate **o**f **w**orld map (SOW) is used to monitor the current state of the joint activity. Each time plans are submitted, the system calculates the result and the SOW is updated by the system. The chat window is used by participants to share intentions and communicate progress that is not represented in the SOW. The attachment window allows semi-structured notes of information to be published in a shared space. These notes can be altered and republished when the user wishes to update them, and then viewed by other users.

State of World Map

The SOW is a segment of common ground that graphically represents several kinds of information about the location and status of objects. It is a partial WYSIWIS (***W****hat* ***Y****ou* ***S****ee* ***I****s* ***W****hat* ***I*** ***S****ee*: Stefik, Bobrow, Foser, Lanning, and Tatar, 1987).

Figure 1 shows an example of an SOW for a crane operator. Only a portion of the virtual world is visible to each user at any time. The visible area is a circle of predefined radius around the actual location of the vessel. This area is represented in the interface by a darker circle surrounding the vessel. Each vessel can also see all large barges, and the tug operator can see all small barges. A large barge is represented by a triangle icon (e.g., brg2). Barrels of toxic waste are represented by a rectangular icon (e.g., xwaste1), the user's vessel is represented by a red circle (e.g., crane1), and other vessels are represented by white circles (e.g., crane2).

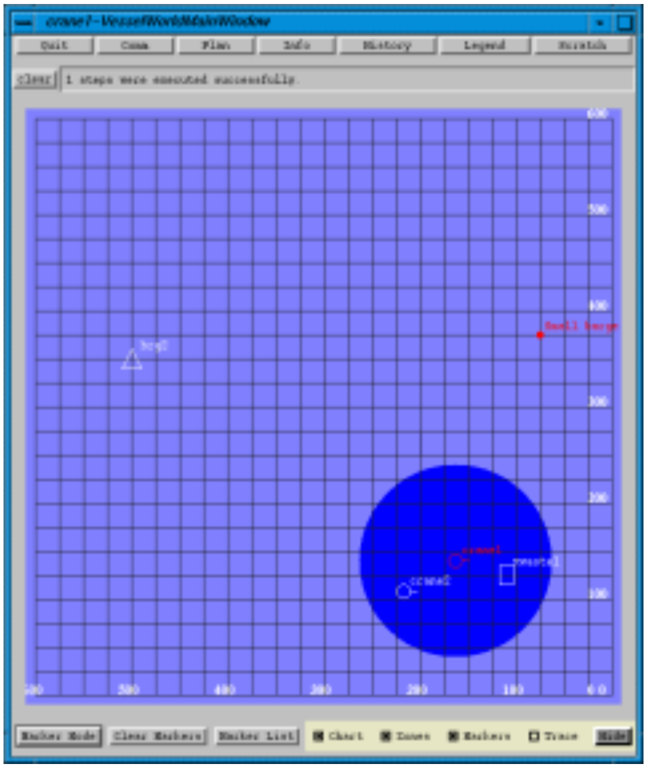

Figure 1: SOW map

The system includes a mechanism for a user to privately annotate the SOW through the use of markers. Markers are labeled points that the user may place on the SOW, which appear only on her SOW. The user can use this capability to label the location of objects and other points of interest. Figure 1 shows a marker denoting the location of one of the small barges. As the problem progresses, the SOW updates to show the current state of joint activities, and the results of public events so far.

Chat Window

VesselWorld provides a separate communication facility in a chat window. This is based on the interface used in **I**nternet **R**elay **C**hat (IRC) a public on-line forum (Shneiderman, 1998). Communication in the chat window occurs separate from other actions within the system. Each participant writes out whatever text they want to send and clicks the "Submit" button. Communication occurs simultaneously among all participants. The model has some of the same possible drawbacks as IRC (overlapping conversations and responses coming in out of sync), but also has the advantages (instant communication, and ease of use). Others have commented on the advantages of interactive communication (i.e., chat) over parcel-post (i.e., e-mail) for keeping common ground synchronized (Tatar, Foster, and Bobrow, 1991; and Trigg, Suchman, and Halasz, 1988).

All text that is sent or received by a user is saved in a communication history window. From this window each user can see what has been previously said and copy and paste text to other windows. This allows the user to review what agreements and decisions have been previously made and potentially reuse old conversations.

Attachment Window

In VesselWorld, part of the common ground is also explicitly represented by users in an attachment window (see Figure 2). The attachment window is a synthesis of a shared message board (Stefik et al, 1987, also Ellis et al, 1991) and structured message system.

All attachments posted to the window are WYSIWIS. The attachments are semi-structured messages which participants may use to monitor and communicate current information, initial common ground, and shared intentions/plans. Semi-structured communication has been shown to be more effective than either unstructured or fully structured communication in coordinating complex activities (Malone, Grant, Lai, Rao, and Rosenblitt, 1988). Attachments can be re-used within a problem-solving session and can be retained for use in future problem-solving sessions. Re-used attachments are the basis for the accumulation of common ground throughout the history of a community of actors using the system to coordinate activity.

Any messages which are posted to the attachment window may be modified by any user. Any message which has been modified by one user is flagged with an asterisk in the other users' displays until they view the updated version. This setup allows free collaboration on a plan to take place.

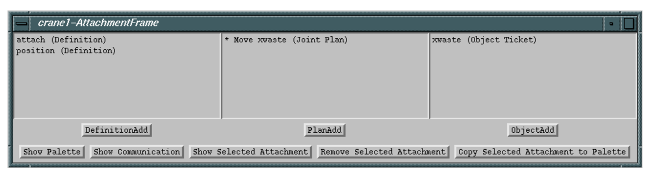

Figure 2: Attachment Window

The three panes in the attachment window provide storage for three different types of attachments: definitions, shared plans (dispatch or joint), and object tickets. Definitions are representations for a term generated by users. For example, users may define a method for representing the location of a given object (see Figure 3).

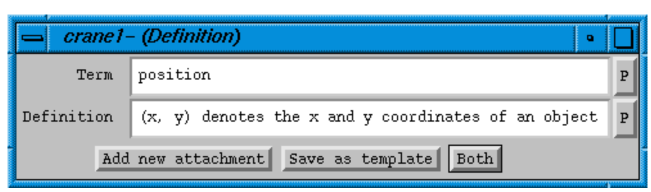

Figure 3: Definition

Shared plans are created by the users to coordinate sub-problems encountered during the session. Object tickets are created to monitor the state of known toxic waste barrels.

The use of the attachment window and semi-structured messages is best illustrated by way of example. In this scenario, a crane operator encounters an extra large waste barrel (Figure 1). The crane operator first posts an object ticket (Figure 4) to the attachment window, so that the other users are aware of the location of the object.

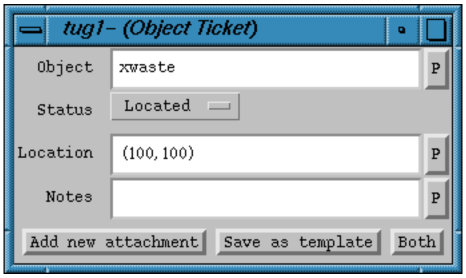

Figure 4: Object Ticket

The crane operator then proposes a *joint plan* (Figure 5) to have the tug operator move a small barge to the waste (Step 0), have all the vessels assemble at the location of the barrel (Step 1), have the cranes put the barrel on a small barge (Steps 2-3), and have the tug move the small barge to the large barge (Steps 4-5). At this point, the plan is posted, and the other users are free to alter the plan until a consensus is reached.

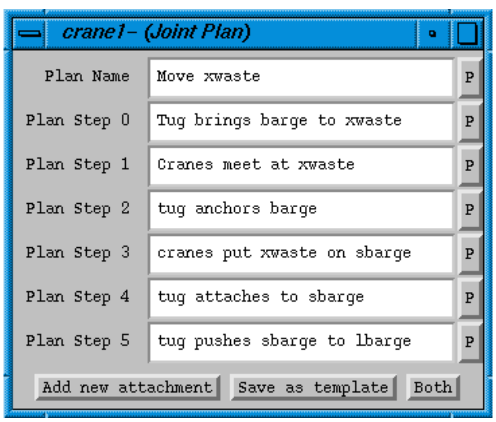

Figure 5: Joint Plan

The participants then carry out the steps of the plan, communicating where necessary to coordinate fine-grained actions not specified in the plan (such as the exact time the cranes must lift together). The joint plan is useful for step-by-step planning, but is cumbersome to use for more freeform planning, where the timing of actions is not as important as what they are doing. For this sort of planning, a different sort of plan is provided. This is the *dispatch plan*, shown in Figure 6.

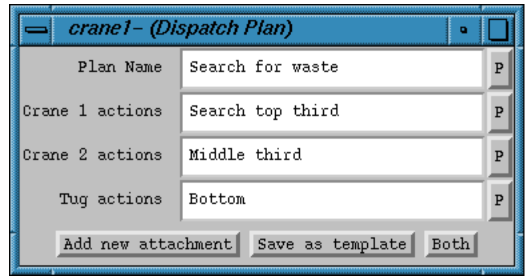

Figure 6: Dispatch Plan

The dispatch plan shows what actions each operator has agreed to perform to fulfill the plan. In this example, the operators are attempting to locate all the pieces of toxic waste. The dispatch plan provides a compact representation of this division of labor, both to communicate the plan to all the users, and to remind them of their responsibilities.

ADAPTIVE GROUPWARE AND CONVENTIONS

The set of experiences that are generated in the course of the users' continued work over many sessions are a source of information that can be used to adjust the system's behavior so as to better fit the user to the system and the system to the domain. Over multiple episodes of planning with the system, the practice that develops is a significant part of the performance in the person machine interaction.

The necessary requirement for building any adaptive system is the existence of a regularity in the user's behavior. How does the notion of a regularity in behavior translate into groupware for planning? What kinds of regularities of behavior between users develop as a result from the practice of planning together within a community of actors? Are they reliable? How can they be represented? What kinds of enhancements can be made to the design of the system to facilitate their emergence? What sorts of adaptations can be made that depend on these regularities of joint behavior?

Conventions

Joint activities are essentially a complex of coordination problems (Clark, 1996). Playing a duet, shaking hands, rowing a boat in tandem, eating dinner at a restaurant, conversing with another, and so on, are all joint activities that require that the participants solve a set of coordination problems. Coordination problems are resolved using coordination devices, which come in four types: salience, explicit agreement, precedence, and convention.

> In VesselWorld, two crane operators are near a large barrel of waste. They decide without communication to lift it together; the *salience* of the action means that only minimal coordination is needed. One of the crane operators then asks the tug operator to move a small barge to the barrel; the tug operator agrees to this *explicit* request. The last time the users loaded a barrel onto a small barge, they then proceeded to the large barge to transfer the barrel to the large barge; so, following *precedence*, the crane operators follow the tugboat to the large barge and unload the barrel. When the tugboat operator finds another large barrel, all he does is create an object ticket denoting its location; by

convention, the crane operators know to show up at that location and load the barrel as they had before.

Lewis (1969) defines convention as a solution to a recurrent coordination problem. Conventions are the regularities of behavior that develop among a community of actors with a tradition of common goals and shared activities.

In the domain of VesselWorld, over time, conventions develop for coordinating behavior between participants in a joint activity. Locations of objects are communicated between users using a certain notation scheme. Standardized procedures for loading barrels onto a small barge emerge that reduce the communication necessary to coordinate the lifting and loading of the barrel with the anchoring of the barge by the tug boat captain.

Lewis defines several important features of conventions, three of which are: 1) conventions are common knowledge within a community of actors; 2) it is common knowledge within the community that everyone expects almost everyone else to conform to the convention for coordinating behavior; and 3) conventions are a set of preferences regarding all possible combinations of actions.

Two crane operators loading a large barrel onto the small barge may become a regularity of behavior. In order for this procedure to be a convention, the procedure must be commonly known (feature 1), and it also must be common knowledge that everyone conforms to this procedure (feature 2). This procedure involves a complex of actions (feature 3), including: anchoring the small barge, lifting together, and loading together.

At Brandeis we have been developing a cognitive model of how convention emerges from group practice (Alterman & Garland, 1998; Garland & Alterman, 1998). The main focus of this work is on how new conventions develop in novel communities of activity. The model details the emergence of convention in circumstances where there is no ruling body of knowledge (as developed by prior generations of actors within the community) to draw on and to guide cooperative and coordinated behavior. The example domain is a group of actors who are part of a moving company. Their job is to move boxes and furniture from a house into a truck. With practice, individuals within the community begin to converge on a set of conventionalized behaviors that best match the regularly occurring problems of coordination in the domain of activity. An analysis of a large set of experiments shows that the development of convention has several important properties; an example of these results is shown in Figure 7.

- A group of individuals with varying experiences can converge on a patterns of cooperation for problems of coordination that regularly occur.
- The development of convention improves performance (see Figure 7).
- The development of convention reduces communication and planning costs.
- New conventions for coordinating joint activities develop as a result of practice.
- The development of convention benefits novices.

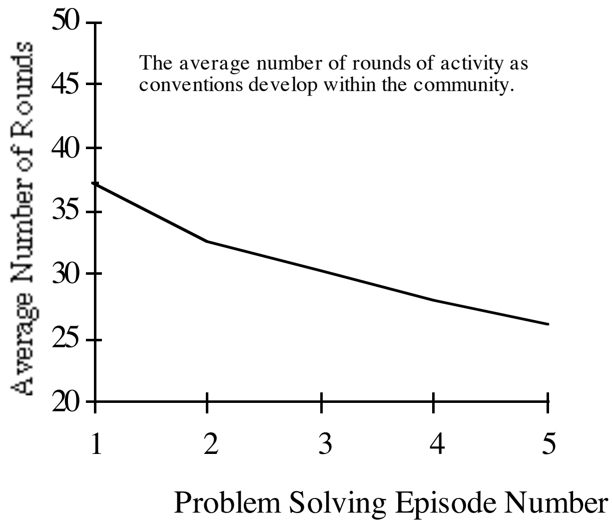

Figure 7: Overall Community Performance Improves

Each of these findings has significance in the development of systems for groupware planning. Conventions are fixes to problems in coordination of joint activity. Adding technology to a system that facilitates the emergence of convention will result in improved system work and a reduction in user effort. New conventions emerge from practice, which can be converted into system adaptation. Explicit representations of conventions potentially aid novice users of the system.

DESIGN FOR THE EMERGENCE OF CONVENTION

Adapting a system can require a mix of a user and system effort. The two extremes are the *adaptable system*, which is based solely on human effort, and the *adaptive system*, which automatically adapts (Oppermann & Simm, 1994). End-user programming is an example of a method for building adaptable systems. The advantage of this approach is that the user has a great deal of control over adaptations in the system; the disadvantage is that it requires considerable user work and skill (Nardi, 1993). **P**rogramming **b**y **d**emonstration (Cypher, 1994), PBD, is a method of end-user programming that attempts to mitigate the problems with this approach by reducing the expectations for end-user work and skill. The EAGER system (Cypher, 1991) is an example of a PBD system for creating macro-operators that is mostly an adaptive system. The system observes the user's behavior and builds macro-operators, behind the scenes, which the user can, at her discretion, use or not use. The advantage of this approach is that expectations about user work and skill are lowered; the disadvantage of this approach is that the system builder is burdened with the task of building into the system a theory that is robust enough to account for much of the user's behavior. This sort of high-level interpretation is notoriously difficult to do.

In the model of adaptive groupware that we describe, the work done by the user fits the following criteria:

1. it is work the user wants to do
2. it is work that directly benefits the user on the current problem
3. it is work that is readily convertible into system adaptation

The key idea is to tie the adaptation of the system into the machinery used by participants in the joint activity to manage the representation of common ground. Our insight is that the explicit representations built by users to coordinate their behavior for a given problem are objects that the user will want to re-use in future problem solving sessions and are consequently readily convertible into adaptations to the system.

Re-use of Shared Plans and Adaptation

What is the work the users do that they want to do and that benefits the users on the current problem? Common ground! In order to coordinate their behavior during the course of a joint activity, participants must develop, manage, and monitor common ground. Since common ground is a product of work that users want to do and improves their performance on the task at hand, it is a prime candidate for being the work that is converted into adaptations to the system.

One example is the re-use of definitions agreed upon in prior planning sessions. For example, once a term like *position* has been defined technically in terms of x and y coordinates, its definition can be saved so that the participants do not need to re-define it in future planning sessions. Once saved and re-used it begins to become a part of the initial common ground that informs the participants' joint activity. As new users join the community of actors, access to these definitions will aid the development of novices.

The system also allows messages to be saved and re-used. Any message that a participant receives may be saved by the individual user so that it can be re-used in the future, within and across sessions.

A more interesting case of developing common ground are the shared plans (joint plans and dispatch plans) that are created by the users in order to coordinate their efforts. For regularly occurring problems of coordination in the domain of activity, the potential exists for these shared plans to be re-used and developed into conventions. Re-using shared plans has several advantages:

1. Over time re-used shared plans can be debugged.
2. Re-using a shared plan reduces the amount of work for the user in creating a new shared plan.
3. Re-used shared plans become a part of the initial common ground assumed in future problem-solving sessions.
4. Re-used shared plans represent regularities of user behavior that can be used by the system as a basis for off-loading some of the user work in managing information.

Continuing the problem shown in Figure 1, let us suppose that the waste barrel found has been safely transported to the large barge. The tug operator then comes across another barrel while pulling a small barge around. This situation is shown in Figure 8. Recognizing that the situation is very similar to the one for moving the previous barrel of waste, the tug operator decides to reuse the old plan, with minor modifications.

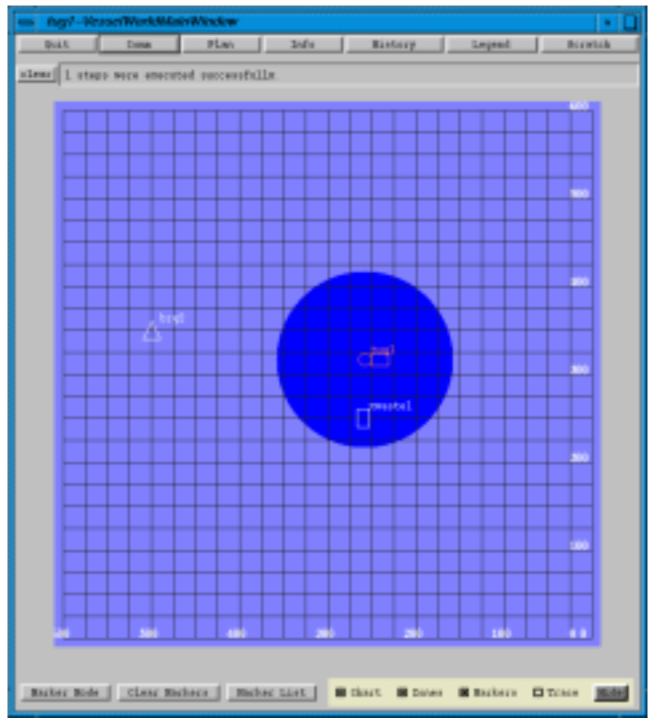

Figure 8: SOW for the tug

First, the tug operator creates an object ticket for the new barrel of waste. Pulling up the old plan (Figure 5), the user makes some minor modifications, and publishes the new version of the joint plan, noting that step 0 can be skipped, as the small barge is already present (Figure 9). She then submits this re-used plan.

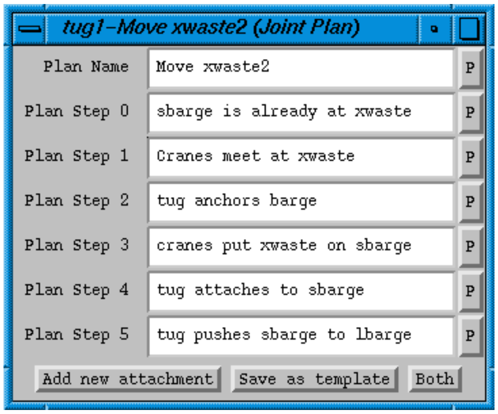

Figure 9: Re-used joint plan

EVALUATION

In an earlier version of the system, SYSTEM0, users could communicate only by parcel-post. The design that has been presented in this paper is SYSTEM1. A pilot study for the Vessel World task was conducted using each of these interfaces.

Method

Participants

The participants were experienced computer users including programmers, engineers, and Navy specialists. They worked in three-person groups, formed by scheduling availability.

Apparatus and environment

The experiment took place in a quiet computer laboratory. Four linked workstations were used, with one serving as the control server and the others as distributed clients, one for each user. Five-foot tall padded panels separated each of the workstations so that users could not see one another or each other's screens. Once a session began users could only communicate via the system-provided facilities.

Problem difficulty was determined by the number and size of barrels to be cleared. Thirty levels of difficulty ranged from minimal (two small and one large barrel) to high (six small, four medium, five large, and one extra large barrel). For the first session, problem difficulty was minimal. After the first session, problem difficulty was selected randomly.

Procedure

On the first day, each group was trained to operate the system. Instructions were read and actions demonstrated and practiced by each user. Following the demonstration, the first session began. Each group solved one problem per day for each of five days.

Results

As we report only the preliminary results of a pilot study, the results are largely qualitative and observational. The most important of these is a within subjects comparison for Pilot Group1, a group that worked with both systems. There was a two month gap between trials with SYSTEM0 and SYSTEM1. A second pilot group, Pilot Group2 has also used SYSTEM1.

When working with SYSTEM0, the users frequently duplicated their efforts. On one occasion, all three users independently searched the same space, found the identical barrel of toxic waste, and then proceeded to label the barrel differently. This clearly is a problem of miscommunication and indicates a lack of common ground. In contrast, with SYSTEM1, the same users used object tickets to create a common frame of reference in identifying objects, and consequently avoided problems of this sort. The result was an improvement in performance.

Another example of miscommunication also arose from a failure to establish common ground. Two crane operators were trying to lift a large barrel together. The two cranes could see one another, but their view areas did not overlap completely (see Figure 10). Each could see one large barrel; however, they were looking at different barrels. Each time they tried to lift together their joint action failed, and they were unable to understand why.

Figure 10: Overlapping viewpoints

After Trial 2, a convention developed among the members of Pilot Group1. They began each session by reporting their own location to one another. This convention arose without an explicit agreement. Toward the end of Trial 3, Pilot Group1 "found" a bug associated with the attachment window. To maintain common ground and coordination, they devised a work-around that used the scratch window and chat mechanism to share plans.

Where problems solved using SYSTEM0 took 22.7 minutes per barrel (weighted by barrel size), this was reduced to 10.2 minutes per weighted barrel the first time SYSTEM1 was used. Figure 11 shows the learning curve for this group for five trials. We expect that users who are initially trained on SYSTEM1 will show similar learning rates as Pilot Group2 solved their first problem in 9.6 minutes per weighted barrel.

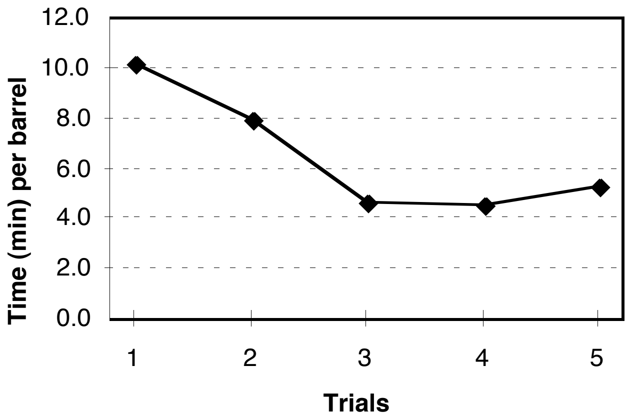

Figure 11: Learning curve for Pilot Group1

SUMMARY REMARKS

This paper has presented a design of a groupware system for planning. The design is theoretically motivated by current results and models in the Cognitive Science literature on group activity and planning. Users of the system work at the same time, but at remote locations. They are participants in a joint activity. The system facilitates the creation, maintenance, and monitoring of common ground; common ground is the basis of participants reasoning about each other's behavior. A special case of common ground are the shared plans created by users to better coordinate their efforts. Since common ground is a product of work that user's want to do and improves their performance on the task at hand, it is a prime candidate for being the user work that is converted into adaptations to the system. For regularly occurring problems of coordination in the domain of activity, the potential exists for shared plans to be re-used and developed into conventions. Re-used shared plans become a part of the initial common ground in future planning sessions, and user work decreases as the initial common ground grows.

ACKNOWLEDGMENTS

This work was supported in part by ONR (N00014-96-1-0440). Additional funding was provided by NSF (ISI-9634102).

REFERENCES

1. Alterman, R. and Garland, A. Conventions in Joint Activity. Technical Report CS-98-199, Computer Science Department, Brandeis University, 1998.
2. Clark, H.H. *Using Language*. Cambridge University Press, Cambridge, 1996.
3. Cypher, A. (Editor) Watch What I Do. MIT Press, 1994.
4. Cypher, A. Eager: Programming Repetitive Tasks by Example. Proceedings of CHI 1991, ACM Press, New York, pp. 31-39.
5. Ellis, C.A., Gibbs, S.J., and Rein, G.L. Groupware: Some Issues and Experiences. *Communications of the ACM 34*, 1, 1991, 38-58.
6. Ellis, C.A., Gibbs, S.J., and Rein, G.L. Design and Use of a Group Editor, in Cockton, G. (ed.), *Engineering for Human-Computer Interaction*. Elsevier Science Publishers, North-Holland, 1990.
7. Garland, A., and Alterman, R. Learning Cooperative Procedures, in Bergman, R. and Kott, A. (eds.), *1998 AIPS workshop on Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments*, AAAI Technical Report WS-98-02, AAAI Press, 1998.
8. Greenberg, S., Roseman, M., Webster, D., and Bohnet, R. Issues and Experiences Designing and Implementing Two Group Drawing Tools, in *Proceedings of the 25th Annual Hawaii International Conference on the System Sciences,* Hawaii, 4, 1992, 139-150.
9. Grosz, B., and Kraus, S. Collaborative Plans for Complex Action. *Artificial Intelligence 1996*, 269-357.
10. Grosz, B., and Sidner, C.L. Plans for Discourse, in Cohen, P.R., Morgan, J., and Pollack, M.E. (eds.), *Intentions in Communication*. MIT Press, Cambridge, MA, 1990.
11. Lewis, D.K. *Convention: A Philosophical Study*. Harvard University Press, Cambridge, MA, 1969.
12. Malone, T.W., Grant, K.R., Lai, K., Rao, R., and Rosenblitt, D. Semistructured Messages are Surprisingly Useful for Computer-Supported Coordination, in Grief, I. (ed.), *Computer-Supported Cooperative Work*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.
13. Nardi, B.A.. *A Small Matter of Programming*. MIT Press, Cambridge, MA, 1993.
14. Oppermann, R., and Simm, H. Adaptability: User-Initiated Individualization, in Oppermann, R. (ed.), *Adaptive User Support*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.
15. Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley, Reading, MA, 1998.
16. Stefik, M., Bobrow, D.G., Foser, G., Lanning, S., and Tatar, D. WYSIWIS Revised: Early Experiences with Multiuser Interfaces . *ACM 1987.*
17. Tatar, D.G., Foster, G., and Bobrow, D.G. Design for conversation: lessons from Cognoter. *International Journal of Man-Machine Studies 34*, 2, 1991, 185-209.
18. Trigg, R.H., Suchman, L.A., and Halasz, F.G. Supporting Collaboration in Notecards, in Grief, I. (ed.), *Computer-Supported Cooperative Work*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.
19. Wiener, E.L. and Nagel, D.C., eds., *Human Factors in Aviation*, Academic Press, New York, 1988.