Intelligent Automation in Collaborative Systems

Joshua INTRONE and Richard ALTERMAN
Volen Center for Complex Systems
Brandeis University
Waltham, MA 02454
[jintrone, alterman]@cs.brandeis.edu

Abstract. Intelligent automation has been a source of research and debate within the design community for several decades. When adding intelligent automation to single-user systems, two critical issues must be addressed. First, sufficient knowledge must be acquired about the user and her context to make high-level inferences at runtime. Second, the automation must be useful and delivered in a manner that does not impair the user's domain activity. These issues are equally relevant for collaborative systems. However, collaborative systems offer a potential solution to these problems by virtue of their privileged position as mediating artifacts within a collaborative process. Because coordination information must be exchanged through the system, there is an opportunity for the system to gain insights into user activities and context. Because mediating artifacts add structure to the information that passes through them to improve coordination, this information is made more accessible to standard AI algorithms. Thus, within a design solution for coordination problems in groupware, a solution to some of the issues with intelligent automation can also be found. Empirical evidence from a testbed domain is presented that validates this approach, along with a discussion of how the approach can be generalized to other collaborative systems.

Keywords: Coordinating Representation, Intelligent Interfaces, Awareness, Bayesian Networks, Plan Recognition

Introduction

Computer systems often function as artifacts that mediate people's activities or communication [1][2]. As mediating artifacts, the design of computer systems is intended to improve users' work by modifying the nature of the task. A computer might provide structure that serves as a resource for activity [3], or it may introduce a layer of abstraction that transforms work in a complex domain [4].

A difficulty in designing systems that do this effectively is that the facilities they provide must be designed prior to their actual use, and hence are limited to providing the kind of support that the designer can envision at design time. Early on, this resulted in systems that were only appropriate for narrow groups of end-users in constrained settings [5]. This problem has been a driving force for the design community, and many techniques have been developed to grapple with the problem. One large subfield of research in user interface design has sought to better understand the nature of activity and tool use in order to improve the work at design-time. Another large subfield has focused on ways to defer design decisions by anticipating many eventualities to be

detected at runtime. Both subfields have encountered their share of difficulties, and very often, the methods employed by the two approaches are seen as incompatible [6].

In this article, a technique for the harmonious integration of recently developed design techniques and intelligent automation in collaborative systems is presented. This approach rests on the observation that structure in a mediating artifact can both simplify work and at the same time render a portion of the users' runtime context interpretable for the computer. The solutions offered are drawn from the rich bodies of work on situated activity and distributed cognition.

In the early sections of this article, an ethnographic analysis technique that guides the development of mediating artifacts that improve collaboration is described. The primary focus of this article will be to show how the coordination work that people do via these mediating artifacts can be used by standard AI algorithms to introduce automation that improves the users' performance in a domain task. The article concludes with a discussion about how intelligent automation might generally be incorporated into collaborative systems to improve awareness, based on recent research and our experiences.

Difficulties with Design

The field of HCI has gone through many stages in its ongoing evolution [5]. At the outset, the design of interfaces was based upon measurements of cognitive variables in carefully controlled laboratory environments. These methods proved to be difficult to translate into real-world applications, as they were only applicable to prototypical users in rarified contexts. Subsequent developments in the field have led to insights that human activity is inherently situation dependant [7], and that the social, organizational, and political context often has as much to do with the acceptance of a piece of software as design itself [5].

The importance of runtime context and situation-dependence raises many design issues. How is it possible to design artifacts for a context that is not known at design time? If the only constant is the dynamicity of situation, how can static software representations be satisfactory? Identifying the kind of structure might be usefully incorporated in mediating artifacts that support work activity (e.g., [3][8][9][10]) has been a central research focus in CSCW since Suchman's observations about situated activity [7].

Difficulties with Automation

One way to deal with some of the above problems is to try to build interfaces that are sensitive to the user's runtime needs [11][12]. These systems try to infer the user's goals, context or characteristics in order to tune their behavior to use at runtime.

In some treatments, these systems are conceived of as a collaborative partner with the user [13]. In order to be a good collaborator, the system must have sufficient access to the context and user information that will allow it to make useful and timely inferences about the user's needs. Unfortunately, the computer is handicapped in this regard, as it can only "see" the user's activity and context through the narrow aperture of the user interface. In order to overcome this problem, a user knowledge acquisition strategy must be designed to provide the system with access to the information needed to make good inferences [14].

A fundamental problem is how to design a knowledge acquisition strategy that does not introduce too much work for the user or otherwise impair the usability of the system. Natural language interfaces might be a way to do this, but natural language understanding is not yet at a point where solutions are feasible for real-world systems. Another approach is to add a structured language to the interface so the user can express higher level intentions in a form the computer can understand, but this requires the user to manage two tasks instead of one. As the developers of one human-computer collaborative system noted, "it is often more efficient and natural to convey intentions by performing actions" ([15], page 23).

Another approach to knowledge acquisition is to add interface structure as part of the task that is mediated by the system. For example, the Epsilon collaborative learning environment [16] requires collaborators to use sentence openers (chosen from a list) for every line entered into chat. This allows the system to monitor the chat and help out with ineffective conversations. It important, however, to balance interface structure for knowledge acquisition with usability concerns. Outside of work discussed in the next section, the authors are not aware of any rigorous methodological approach for adding interface structure that both supports powerful machine inferences but does not impair natural use of the system.

In addition to the knowledge acquisition problem in designing intelligent automation, the functionality provided by the automation should be useful, and it must not interrupt the user or hinder domain activity. Mixed-initiative approaches, such as that proposed by Horvitz [17], provide some guidelines in this regard. Horvitz advocates balancing the cost of the interruption against the expected utility of the automation itself. This entails recognizing where in a task a user is, and what the information requirements for that task are. However, mapping interface activity to a task structure is essentially a keyhole plan recognition problem, which has proven difficult to do in the general case.

Rather than grapple with the problems of identifying task boundaries in user interface activities, we seek to identify a more reliable and generally applicable approach for collaborative systems. To this end, the problem of awareness in collaborative activity is examined at the end of this article.

Mediating Artifacts that Support Coordination

The study of mediating artifacts in everyday and work activities has become widespread in HCI and CSCW. Suchman & Trigg [18] described the role of structured tools (e.g., the "complex sheet") in coordinating the distributed activities of the staff in an airport. Hutchins [2] explained how artifacts allow portions of a task to be "precomputed," effectively mediating communication between the designer and the user. Norman [1] has focused on the role an artifact plays in mediating the user's interaction with a domain, transforming the domain task into a form that makes it more cognitively accessible to the user. Schmidt & Simone [10] describe how artifacts serve to support the articulation of coordinated work activity, and introduce a notation (Ariadne) for the description of adaptable mechanisms that coordinate workflow. Activity theorists (e.g., [19]; also see [20]) highlight the role of mediating structures – which may be material artifacts, but might also be policies, conventions, etc. – in activity systems, which encompass a much larger range of factors than traditional HCI treatments.

Each of these approaches is representative of a rich and evolving line of research, but none provide concrete, design-level guidance that can indicate what kind of structure should be incorporated into a mediating artifact, or predict which artifacts will be used

and which introduce too much work. Ethnographic methodologies informed by activity theory and distributed cognition do provide guidance about how we might conceptualize activity and the role of mediating artifacts, and draw the analyst's attention to the need for mediation, but such theories stop short of explicit design recommendations for the artifacts themselves.

Recently, Feinman & Alterman [21] have provided just such a design level approach. It draws together insights from the ethnomethodological approach of Suchman & Trigg [18] and the analytical techniques introduced by Hutchins[2]. It provides a methodology for moving from the analysis of practice to concrete recommendations for structure that will be useful in a particular collaborative system. This structure is instantiated in shared mediating artifacts that can be incorporated in an existing platform. Following Suchman & Trigg, these artifacts are referred to as *coordinating representations* (CRs).

The methodology for designing and introducing coordinating representations is one solution to the design problems described above. The structure introduced to the interface by coordinating representations is also a solution to the knowledge acquisition problem. In the following, validating evidence is provided for these two claims. Finally, the shared information that accumulates in a coordinating representation may be well suited to providing awareness that is generally useful in collaborative systems.

1. Experimental Platform

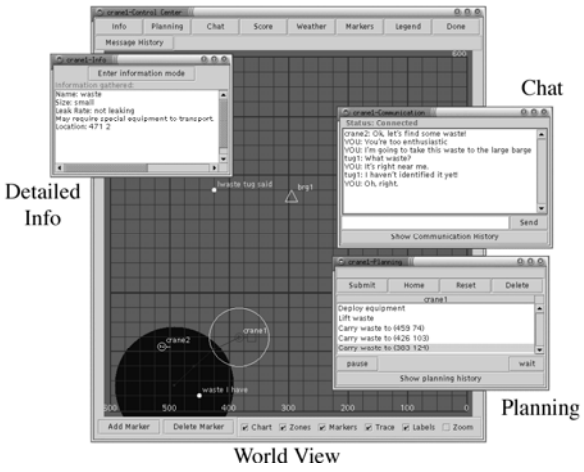

Figure 1: The VesselWorld System

To study the above problems, an experimental platform called VesselWorld (shown in Figure 1) was developed. VesselWorld has many features relevant to the study of groupware systems in general. The domain task entails varying degrees and types of coordination, collaborators have different roles and responsibilities, and awareness must be explicitly maintained. VesselWorld proved to be very challenging for its users; in studies, each user group was trained for two hours before data was collected, and performance usually didn't stabilize until after another five hours of use.

VesselWorld presents a relaxed WYSIWIS environment, in which three participants play the role of ship's captains, and their joint goal is to remove toxic waste barrels from a harbor without spillage. The main interface is a shared map. Each ship has a

geographically limited view of the harbor within this map, so each user has different directly observable domain information. The progression of a VesselWorld session is turn-based, such that every user must submit a step to the server before the server can evaluate the steps and update the world on each client screen. Users may plan any number of steps in advance, although each step can only involve objects that are currently visible, and only one step can be submitted to the server at a time. Users' actions are not visible to one another during the planning phase of each step, so awareness must be explicitly maintained. Communication may occur at any point, but all communication occurs through a text-based chat window that is part of the system.

Each ship has different capabilities. Two of them have cranes that can be used to lift toxic waste barrels from the harbor and load them onto a large barge (which has a fixed position). The third user is a tugboat that can be used to drag small barges from one place to another. For notational convenience, we adopt the convention of referring to the crane operators as "cranes" and the tugboat operator as the "tug." The cranes can load multiple wastes onto the small barge, and at least one of them must also be present to unload the barrels and place them on the large barge.

Toxic waste barrels are of different types and require different coordination strategies. A single crane may lift a small or medium barrel, but two cranes must join together to lift and carry a large barrel, and an extra large barrel may be jointly lifted but can only be carried on a small barge by the tug. Toxic waste barrels may require specialized equipment to be moved, and the cranes carry different types of equipment. The tug is the only actor who can determine the type of equipment a toxic waste barrel requires.

The users are scored by a function that takes into account the number of steps it takes to remove all of the waste barrels, the number of barrels cleared, the number of errors (dropped waste barrels) made, and the difficulty of the problem. In all user studies, the users were instructed to try to maximize their score.

To support analysis, VesselWorld logs complete interaction data that can be used to replay user activity. This is an important component of the methodology described in Alterman, et al. [22]. More details upon this portion of the methodology can be found in Landsman & Alterman [23].

2. Intelligent Automation in VesselWorld

Planning in VesselWorld is laborious and error prone. Errors often occur due to forgotten plan steps or joint plans that have become unsynchronized. Errors also occur because of forgotten or misunderstood commitments. In early versions of the system, a shared component was added to address these problems [22]. The CR allowed users to manually specify their goals and sequence their activities; however it was never used by users. In exit interviews, the users explained that the component introduced too much work, and was too hard to use. Hence, we sought to add intelligent automation to VesselWorld to provide this functionality. As envisioned, a semi-automated component would infer each user's goals, and make them visible to all users. Additionally, once these goals were identified, the system could automatically generate synchronized plans for users.

In order to infer user goals, the system needs to know about the state of the domain (where the toxic waste barrels are and associated information). Unfortunately, it is assumed that the simulated world is "outside" of the system itself, so the system has no

direct access to the domain at runtime. Furthermore, without this information, the system cannot automatically generate plans. Thus, as discussed above, a major hurdle in providing the envisioned automation was in acquiring the information necessary to infer user intent.

2.1. Obtaining State Information

At runtime, the system has access to user locations and recently executed plan steps, but the only source of information about toxic waste barrels is exchanged by the users via chat. This information is very hard for the system to interpret.

An excerpt from chat during a typical planning session shown in Figure 2 demonstrates this. In the dialogue, users frequently refer to wastes by their latitude and longitude coordinates on the shared map. In the first line of the example, Crane2 announces a waste at "120, 420." In lines 2-4, Crane1 asks for clarification about the specifics of the waste. In lines 5-6, the Tug replies (having apparently already investigated that toxic waste barrel) with the corrected coordinates "105, 420" and specific information about the barrel. In line 8, Crane2 thanks the Tug for the clarification, and the Tug closes the conversational turn in line 9.

1. Crane2: I found a waste at 120 420
2. Crane1: ok
3. Crane1: what type of waste?
4. Crane1: large,small?
5. Tug1: 105 420 needs a dredge, i think that is where you are
6. Tug1: small
7. Crane1: ok
8. Crane2: Thanks for checking
9. Tug1: no problem

Figure 2: Excerpt from chat during VesselWorld session

Automatically extracting information about toxic waste barrels from chat logs would be very difficult; the sample dialogue illustrates some of these problems. There are three active participants, and conversational turns that might be used to narrow the reference resolution scope are hard to identify. Also problematic is that referring expressions can change from utterance to utterance even within the same conversational turn. For example, line 1 refers to the waste as "120 420" and line 5 refers to the same waste as "105 420." People can sometimes handle such ambiguities, but this is problematic for natural language processing algorithms.

Rather than developing specialized algorithms to deal with the nuances of three-way, live chat in the VesselWorld domain, it would vastly simplify our task if users were to enter all the information the system needs in a structured form. Although this might seem to unnecessarily burden the user, the next section explains why it is reasonable for this domain, and describes empirical evidence supporting this claim.

2.2. Coordinating Representations

Coordinating representations (CRs) can be introduced to collaborative systems to enhance people's ability to coordinate their activities in a joint task. Feinman &

Alterman [21] describe an approach to developing CRs by examining collected usage data from an existing collaborative system for evidence of recurrent coordination problems, explicit talk about coordination, and emergent structure. They also detail a technique for analyzing the co-referencing activity of users who are engaged in a collaborative task. The results of this methodology are a set of recommendations for structure to be incorporated into the platform. This methodology was employed to develop coordinating representations for VesselWorld.

One of the difficulties observed in the analysis of VesselWorld usage data was with users' ability to manage information about domain objects. Some of the groups handled these difficulties by developing mnemonic expressions. However, users did not always agree on consistent expressions, and coordination errors in the maintenance of this information were frequent. Thus, a CR called the Object List (Figure 3) was designed to support the organization and naming of objects in the world, and was added to the VesselWorld system.

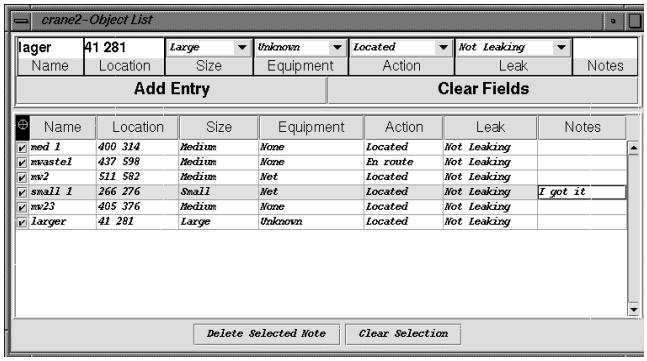

Figure 3: The Object List CR.

The Object List is a tabular WYSIWIS component that helps users to manage and coordinate reference and state information. Users enter and maintain all of the data in the Object List. Each row of data contains several fields of information about a specific object. The "Name" field is a free-text field, assigned by the user. The "Location" field may be filled in by clicking on the field and then on an object that is shown in the primary map (and hence has fixed structure). The "Size", "Equipment", "Action", and "Leak" fields are filled in using drop-down menus. The "Notes" field is also a free-text field, and is provided so that any other relevant information about the toxic waste barrel may be communicated. Entries in the Object List can be displayed on the primary map interface as icons that are annotated with the name that is in the "Name" field at the coordinates in the "Location" field.

In studies it was found that the Object List, and one other CR, were used and significantly improved user performance [22]. Groups that used the CRs had fewer errors, spent less time chatting, and on average took half the amount of time to solve problems. These findings demonstrate that it is possible, using the methodology described, to develop CRs that do not compromise the usability of the system. Rather, they become part of the domain activity of the users, while introducing structure that helps them coordinate. In using a CR, collaborators also create a structured stream of data about their shared context that the system can use to infer user needs.

2.3. Information Provided by the Object List

Use of the Object List generates two types of information that might be used for intent inference. One type is structured information about shared domain objects (toxic waste barrels). This information is not perfect – it is only entered into the Object List as users discover and examine wastes, and it is subject to errors, omissions, and duplication – but it is well-structured and can be readily used by the system.

Another, unanticipated type of information provided is the set of names assigned by users to toxic waste barrels in the Object List. VesselWorld collaborators used these names regularly in chat to refer to objects they were planning to deal with (lift, move, or otherwise). Thus, these names can be used to mine chat for clues about user intentions. A frequency analysis of references preceding actual lifts was performed to establish the utility of this information.

Table 1: Probability a reference precedes a lift at time *t* (in minutes)

	t-5 to *t*		*t-10* to *t-5*		*t-15 to t-10*	
	Joint	Single	Joint	Single	Joint	Single
Lift	.62	.42	.27	.15	.25	.08
~Lift	.15	.11	.10	.07	.08	.04

It was found that the occurrence of references to toxic waste barrels in chat were predictive of lift actions for roughly a fifteen-minute window of time preceding a lift. Table 1 depicts the likelihood that a reference for an object will appear in chat for the three consecutive five minute windows preceding the lift of an object at time *t*. In the table, "Joint" and "Single" refer to whether or not a waste barrel requires both or just one crane to lift. In the *~Lift* conditions, values reflect the likelihood some barrel is referred to prior to the lift of some other barrel.

There is about a sixty percent chance that a toxic waste barrel will be referred to in chat in the five minutes preceding the lift if that barrel requires assistance, and about a forty percent chance if that barrel can be lifted singly. Prior to fifteen minutes before the lift, references were not a very good predictor of lift actions.

2.4. An Intent Inference Procedure

An intent inference procedure was developed to predict user goals, using information about toxic waste barrels and references to them in chat. Two Bayesian Networks (BNs) were developed to assess likelihoods for crane and tug goals. The analysis presented here is restricted to the portion of the crane network that predicts the cranes' lift intentions. This BN is shown in Figure 4; it models the likelihood that an actor has the intention to lift (or jointly lift with the other crane) a specific toxic waste barrel based on information about the state of the world, including:

- The type of equipment required for the waste barrel.
- The size of the waste barrel (which determines whether a single crane can lift the barrel, or if it needs help from the other crane).
- Whether the cranes are close to or heading towards the barrel.
- If the crane is currently holding a barrel.

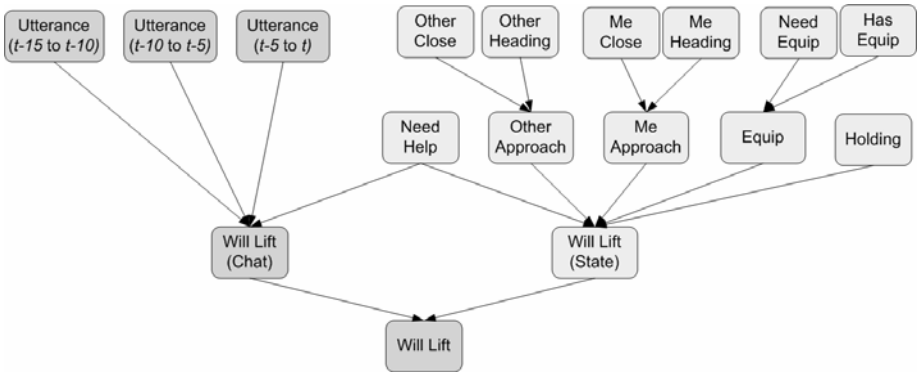

Figure 4: Schematic of BN used to infer crane lift intentions

The network also uses reference information from chat. According the results of the frequency analysis above, three five minute windows of chat history, with one node for each five minute window, are included in the network.

A primary aim of this article is to quantify the utility of the information provided by collaborators for predicting user intentions during normal use of the Object List. To do this, the performance of the above intent inference process was compared across four information conditions; with complete domain information (with or without chat), and with information from the Object List alone (with or without chat).

2.4.1. Evaluating Intent Inference

Table 2: Population summary for evaluations

Group	Sessions	Avg. # of wastes per problem	Total Hours
Group 1	10	11.7	9.9
Group 2	6	11	8.4
Group 3	9	14.3	9.1
Group 4	16	14.5	8.7
All	41	13.5	34.3

The intent inference procedure was evaluated using a dataset spanning roughly 34 hours, which contained usage data from groups that used a version of VesselWorld with the Object List. This data is summarized in Table 2. The usage data contains information about all domain actions, chat, and use of the Object List. It does not however, contain complete and accurate information about the initial state of the domain (where each waste barrel is, what kind of equipment it requires, etc.). For the sake of the following evaluation, complete domain information was derived from the domain definition files that were used to initialize each session of use.

The four information conditions compared were:

- Complete Info – All information about toxic waste barrels (size, location, equipment) is known at the outset, and is correct.
- Object List – Information about toxic waste barrels is taken from the Object List as it becomes available, which is subject to user errors.
- Complete Info + Chat – The Complete Info condition, plus the occurrence of references in chat. This condition uses the names associated with objects in the Object List, but uses complete domain information in the inference process.

- Object List + Chat – The Object List condition, plus chat reference occurrences.

In the non-"Chat" conditions, the belief network shown in Figure 4 was used without the nodes specific to chat (a darker shade in the figure).

For each information condition the network was trained (using the *EM(η)* algorithm [24]) on the complete dataset in Table 2, and then tested against the same data. Training and testing on the same data set is not typically appropriate for validating a particular machine-learning technique. However, the aim here is to establish the relative utility of various information sources rather than to validate the generality of the technique.

Two performance metrics were calculated in each condition; the proportion of correctly guessed goals, or correct goal rate (CGR); and the proportion of guesses that were false, or the false positive rate (FPR). A guess is made whenever a relevant state variable changes. Any uninterrupted sequence of correct guesses leading up to the execution of the predicted goal is counted as a single correct goal. The total number of goals is the number of wastes lifted. Thus,

$$CG_R = \textit{correct goals} / \textit{total goals}$$
$$FP_R = \textit{incorrect guesses} / \textit{total guesses}$$

The results of the evaluation are presented in Table 3. A single factor ANOVA demonstrated that differences between groups were highly significant for CGR ($F(131,3)=10.84, p<.0001$), and significant for FPR ($F(131,3)=3.98, p<.01$).

Table 3: Intent inference results for different info sources

Condition	CGR (StdDev)	FPR (StdDev)
Complete Info	.83 (.14)	.53 (.13)
Object List	.70 (.17)	.60 (.16)
Complete Info + Chat	.87 (.12)	.51 (.11)
Object List + Chat	.77 (.15)	.58 (.15)

The "Complete Info" case, in the top row of the table, provides a baseline against which results for the other conditions may be compared. It is a rough indicator of the best the intent inference procedure can do, given complete and accurate information about the state of the world. Across the four user groups in the dataset, the CGR for the "Complete Info" case ranged from .77 to .91, and there was a weak correlation between problem size (number of toxic wastes) and performance ($r=.23$), reflecting the fact that it is more difficult to make good guesses when there are more options to choose from. In general, these metrics indicate that the inference procedure is effective.

As expected, the intent inference procedure does not perform as well with information from the Object List alone. However, results from the "Object List" condition were still good, and demonstrate that use of the Object List was reliable enough to be useful for intent inference.

The "Complete Info + Chat" condition demonstrates that references add significant information that cannot be derived from knowledge about the state of the domain. Thus, regardless of access to state information (for instance, if there were intelligent sensors placed in the world) the Object List adds information that still improves intent inference.

The combination of reference information from chat and domain information from the Object List (the "Object List + Chat" condition) improves the performance of the procedure to a point where it is nearly as good as with complete information alone. This result provides validation of the claim that, for VesselWorld, the addition of the Object List provides a rich source of structured information that can be used to infer users'

intentions. The introduction of a semi-automated component that uses this information, described in the next section, provides validation that this level of intent inference is good enough to improve the users' domain performance.

3. A Semi-Automated Component

The intent inference procedure above was developed to drive a component that would improve users' awareness of each other's goals, and fix some of the difficulties they had in creating and coordinating plans. The component that was developed for this purpose is shown in (Figure 5). The top portion of the component (everything above the "Get Plan" button) contains the same information for each user. It displays the five most plausible goals calculated by the intent inference procedure for each user at any point in time. When a user selects from among these goals, the goal is copied into the top row of the component, making it apparent to others, and the user is given the option to retrieve an automatically generated plan.

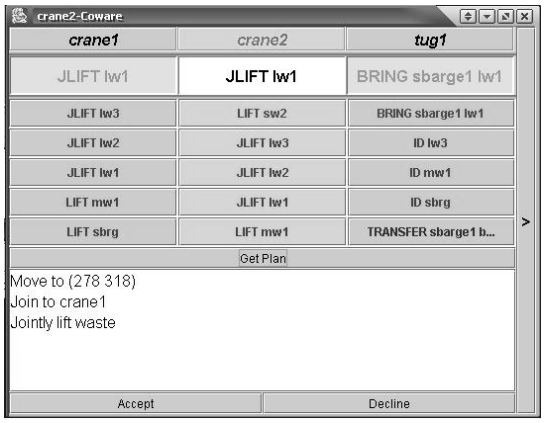

Figure 5: The adaptive component

The specific function of the component is as follows:

1. After each update to state information, (e.g., plan execution, information added to the Object List, a reference to an object mentioned in chat, etc.) the system updates the lists of plausible goals for each user.
2. When a user selects a goal, it is copied to the top row so that all users can tell what was selected. The user that selected the goal is then given the option to request an automatically generated plan by clicking the "Get Plan" button.
3. The system generates a plan that the user can inspect. If the goal involves other users, they are invited to join the plan. If all invited users accept the invitation, a plan is generated; if they do not, the requesting user is so informed and a plan is not generated.
4. The user may then accept the plan, in which case it is copied into the user's planning window for execution. If the plan is generated from correct state information (i.e. the Object List reflects correct state information), and no user modifies the state in such a way that conflicts with the generated plan, the plan will succeed.

The component does not interrupt users. Rather, it is a resource that collaborators can use to monitor one another's goals, and to generate accurate plans. We expected to find that the component would be heavily used, that use would result in fewer planning errors, and users would use it, instead of chat, to maintain awareness of each other's activities.

3.1. Evaluation

To evaluate the component, a single 40-hour study with four teams of three people was performed. The players were a mix of students and local-area professionals, with varying degrees of computer proficiency. Each team was trained together for two hours in the use of the system, and then solved randomly chosen VesselWorld problems for approximately ten hours. To alleviate fatigue concerns, the experiment was split into four three-hour sessions.

The participants were divided into two populations of two teams each, one that had the automated component, and one which did not. For the teams with the component, the inference procedure used information from the Object List and chat to infer user goals. The following results report on the last 5 hours of play time for each group, by which point user performance had stabilized.

Use of the automated component was analyzed, and several metrics were compared across the two groups, including: time taken per waste barrel, number of mouse clicks; amount of communication; and the number of joint and single errors. Additionally, exit interviews were provided to determine if the component was well received, and if it was in fact used to stay aware of other users' activity. All interview questions were answered on a seven point scale.

The component was used. All groups used the component to generate plans within the system. Users confirmed a goal every 1.5 minutes (SD=46 seconds), requested a plan for each confirmed goal, accepted 71% of plans requested (SD=19%), and completed the execution of 83% (SD=6.75%) of these plans. Overall, this indicates that roughly 59% of confirmed goals resulted in a plan that was executed to completion.

In answer to the question, "What did you think of the component?" the average survey response was 5.4 (SD=.8) (1= "Hated it", 7= "Loved it"). To the question "How did the component influence the difficulty of the problems," the average response was 5.6 (SD=.8) (1="Harder", 7= "Easier").

For each problem solving session, one quarter of all plan steps submitted to the server were generated by the component (SD=8%). Finally, the component generated plans for 43% (SD=15%) of the domain goals it could have predicted for the cranes. It was not possible to obtain a similar statistic for the tug because it is difficult to recognize goals in the collected log files (goals for the tug are not bracketed by easy to detect plan steps like "LIFT" and "LOAD").

Joint errors were reduced. Although there was no significant change in the number of individual errors by groups that had the adaptive component, these groups did have 45% fewer joint errors (failures during joint actions) per minute (p=.069). This difference is not significant at the .05 level, because of the small sample size and overall low proportion of joint errors. However this finding corroborates prior analysis of use of the VesselWorld system [22], which indicated that joint errors were usually the result of plan submissions becoming unsynchronized. Because the component generates

coordinated plans in advance, users may simply submit each step and be assured that actions will be coordinated.

Cognitive load was reduced. The average time per waste decreased slightly for users with the plan-generation component, but this difference was not at all significant. With closer investigation though, it was found that the amount of clock time taken by users between steps of automatically generated plans was 57% less than in groups without the component ($p<.01$). Time taken between the submission of automatically generated plan steps was also less than time taken between manually generated plan steps within groups that had the component (52% reduction, $p<.01$). Furthermore, there was no significant difference in the number of mouse clicks per waste. Because the reduction in clock time for groups with the component cannot be explained by a reduction in the amount of interface work, we conclude that the component reduced the cognitive load of the collaborators.

The component was NOT used to maintain awareness. The overall amount of chat was not reduced in groups that had the component. By itself, this finding does not necessarily indicate that the component did not improve awareness; however none of the survey respondents indicated that the component was used to monitor other users' goals.

In general, these results demonstrate that intent inference using information provided by the users as part of their coordination work was good enough to support useful automation. Furthermore, the automation resulted in improved domain performance and reduced cognitive load, and in this respect it was successful. However the component did not appear to improve users' awareness of one another's activities. The following discussion examines this result more carefully.

4. Discussion

This paper has presented validation for an approach to adding useful intelligent automation to collaborative systems. To a large degree, this approach can be readily applied to other collaborative systems. The analytical process that led to the development of the Object List in VesselWorld is a repeatable design technique that has been documented in detail and shown to work with other systems [22][21]. CRs introduce work that people are willing to do and improves their performance in a domain task. They also add structure to coordinating information. Others have discussed the potential of structured collaborative information in supporting user-sensitive runtime support (e.g., [9], also see [25] for a theoretical treatment). The work that was reported upon here provides a concrete example of how this information can be used to produce powerful runtime inferences, which in turn support the integration of useful automation.

The automation added to VesselWorld addressed users' planning needs. However, these needs are fairly specific to the VesselWorld domain; moreover, users did not use the automated component to maintain awareness, which is a more persistent problem in groupware environments. We examine this result more closely against the backdrop of existing awareness research.

Awareness is a multi-faceted issue that is central to groupware development ([26][27]). One difficulty with supporting awareness in collaborative settings stems from asymmetries in information production and consumption [28][29]. Individuals who are responsible for generating awareness information do not always reap its benefits, and consumers of awareness information cannot guide its production. An approach to this

problem is to passively collect information about user activity, and then to make it available as background information that collaborators may use as necessary.

This was the approach taken with VesselWorld. The Object List structured a portion of the coordinating information that was generated, enabling automated inferences about activity. These inferences were published as background for the shared activity. However, our empirical studies indicated that while this background information did not interfere with activity, it was not useful. Clearly, some design guidance for providing the right *kind* of awareness information is necessary.

CRs may collect information that is well suited for supporting *activity awareness*, which is awareness of how work is embedded within the context of the overall activity [30]. Activity awareness is distinct from "social awareness" (awareness of who is around) and "action awareness" (awareness of what is happening). The term "activity" is used to point to an activity theoretical framework, and as such this concept of awareness is more richly textured than can be effectively summarized here.

The role of activity awareness in collaborative settings becomes clearer if the role of context in activity is considered. In ordinary work environments, tasks cannot be neatly organized into preplanned episodes of behavior. New tasks appear dynamically, and existing tasks may bifurcate or be de-prioritized. Each time a new task is engaged, the relevant external information must be brought into focus and cognitive resources realigned accordingly in order to proceed. Upon returning to an earlier task, that task's state must be recovered so that activity may proceed. Context shifts increase cognitive load, and each such shift is a potential loss of prior context [31][32].

Collaborative work is characterized by rapid shifts between individual and carefully coordinated activity [28][27]. These shifts are partially informed by pre-defined workflows, but they are just as often unpredictable and opportunistic. Support for activity awareness in collaborative environments is one way to ameliorate some of the problems inherent in the continual context switching that characterizes collaborative activity. For example, as collaborators in a virtual environment move from a shared super-task to individual sub-tasks, helping them to maintain awareness of the super-task should help eliminate errors like forgotten commitments. The use of shared timelines in a collaborative learning environment as described by Carroll, et al. [30] is such an approach.

The forgotten or misunderstood commitments observed in VesselWorld may be attributed to a lack of activity awareness. As more toxic waste barrels are found in a VesselWorld session, more complicated plans are formed which involve multiple segments - for instance, "Get the wastes in <region>" or, "Get the two Extra Large wastes," which involves several sub-plans and all three users. In executing these more complex plans, individual users must move through several layers of context, which are not explicitly available in any external representation. A loss of high-level context will result in forgotten or mis-remembered commitments, and lack of an external representation of high-level context makes it difficult to catch misunderstandings early on.

The automated component may not have been used to maintain awareness in VesselWorld because it provided information about the immediate individual goals (a form of action awareness) rather than the encompassing shared goal. For any given user, a reminder about their own current (low-level) goal is not very helpful, especially since the plan-generation component automatically generates plans for low-level goals. A reminder about other users' low-level goals may not be useful without seeing how they

fit into the encompassing shared task context. A more useful automation would provide this kind of information.

In general, providing a background collaborative context based on passively gathered information can overcome asymmetries in the production and consumption of awareness information. One type of awareness that may be supported is about shared, high-level context and its relationship to individual activity. This type of awareness is especially important as collaborators move through various phases of coupling, because there is substantial opportunity to lose track of encompassing, shared context.

Coordinating representations are useful for generating the information required to provide this kind of awareness because they capture and structure information that constitutes the users' shared context. The computer, as a mediating artifact with significant abilities to summarize, sort, and synthesize structured data is in a good position to automatically combine and provide this information in the background. We conclude that while the specific automation provided in VesselWorld may not be easily generalized to other domains, CRs may be generally useful for supporting automated activity awareness.

5. Summary

This article has presented an approach that combines ethnographic design with intelligent automation in order to improve collaborative activity, and this approach has been validated with an example. Specific attention has been given to the utility of information that is generated by users in the course of their collaborative work, and it has been shown that, in VesselWorld, this information is nearly as good as complete knowledge of the state of the domain. It was shown how this information can be used to support useful run-time automation, in the form of planning support.

The presented approach is built on the observation that mediating artifacts can structure communication to improve coordination. In adding structure to coordinating information, it is made accessible to autonomous algorithms. The approach may be generalized to other collaborative domains. With regards to the design of coordinating representations and the knowledge acquisition problem, there is a strong case for the generality of the approach to be found in existing and prior research. With regards to automation, existing research that guides the development of generally useful awareness support has been highlighted. The approach we've presented allows the system to passively monitor coordinating information that may be very useful for generating activity awareness. In future work, this hypothesis will be investigated more directly.

References

[1] Norman, D. A.:1991, "Cognitive artifacts." In J. M. Carroll, editor, *Designing interaction: psychology at the human-computer interface*, Cambridge University Press. 17-38.

[2] Hutchins, E.:1995, "Cognition in the Wild." Cambridge, MA. MIT Press.

[3] Schmidt, K.:1997, "Of Maps and Scripts," *Proceedings of Group'97.* Phoenix, AZ. ACM Press, NY: 138-147.

[4] Hutchins, E., Hollan, J., Norman, D.:1985, "Direct Manipulation Interfaces." *Human-Computer Interaction.* 1, 311-338.

[5] Carroll, J.:1997, "Human-computer interaction: psychology as a science of design." *Int. J. Human-Computer Studies* 46, 501-522.

[6] Maes, P., and Shneiderman, B.:1997, "Direct Manipulation vs. Interface Agents: A Debate," *Interactions*, 4(6), ACM Press.
[7] Suchman, L.:1987, "Plans and Situated Actions. The Problem of Human Machine Communication." Cambridge University Press.
[8] Kaplan, S., Tolone, W., Bogia, D., Bignoli, C.:1992, "Flexible, Active Support for Collaborative Work with Conversation Builder," in *Proceedings of CSCW '92*. Toronto, Canada, 378-385.
[9] Malone, T., Lai, K., Grant, K.:1997, "Two Design Principles for Collaboration Technology: Examples of Semiformal Systems and Radical Tailorability." In J. Brandshaw (Ed.), *Software Agents*, AAAI Press/The MIT Press, 109-143.
[10] Schmidt, K., and Simone, C.:1996, "Coordination mechanisms: Towards a conceptual foundation of CSCW Systems Design," *Computer Supported Cooperative Work. The Journal of Collaborative Computing*, 5(2-3), 155-200.
[11] Hefley, W.E. and Murray D.:1993, "Intelligent User Interfaces." In *Proceedings of IUI'93*, Orlando, Florida. ACM Press, NY, 3-10.
[12] Benyon, D.:1993, "Adaptive Systems: A Solution to Usability Problems." User Modeling and User Adapted Interaction, 3, 65-87.
[13] Rich., C. and Sidner, C. L.:1998, "COLLAGEN: A Collaboration Manager for Software Interface Agents." *User Modeling and User-Adapted Interaction*. 8(3-4):315-250.
[14] Jameson, A.:2002. "Adaptive interfaces and agents." In J. A. Jacko & A. Sears (Eds.), *Human computer interaction handbook*. Mahwah, NJ: Erlbaum.
[15] Lesh, N, Rich, C., and Sidner, C.L.:1999, "Using plan recognition in human-computer collaboration," In *Proc. 7th Int. Conf. on User Modeling*, pp. 23—32.
[16] Soller, A.:2004, "Computational modeling and analysis of knowledge sharing in collaborative distance learning," *User Modeling and User-Adapted Interaction*. 14: 351-381
[17] Horvitz, E.:1999, "Principles of Mixed-Initiative User Interfaces." In *Proceedings of CHI' 99*. Pittsburgh, PA. ACM Press, NY, 159-166.
[18] Suchman, L. and Trigg, R.:1991, "Understanding Practice: Video as a Medium for Reflection and Design." In Joan Greenbaum & Morten Kyng (eds.), *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.
[19] Engeström, Y.:1987. "Learning by Expanding: An Activity-Theoretical Approach to Developmental Research." (Helsinki: Orieta-Konsultit).
[20] Nardi, B.:1996, "Context and Consciousness: Activity Theory and Human-Computer Interaction." Cambirdge, MA. The MIT Press.
[21] Feinman, A., and Alterman, R.: 2003, "Discourse Analysis Techniques for Modeling Group Interaction." In Brusilovsky, P., Corbett, A., and de Rosis, F, editors, *Proceedings of the Ninth International Conference on User Modeling*, 228-237.
[22] Alterman, R., Feinman, A., and Introne, J.: 2001, "Coordinating Representations in Computer-Mediated Joint Activities." *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, Boston, MA, 43-48.
[23] Landsman, S., and Alterman, R.:2005, "Using Transcription and Replay in Analysis of Groupware Applications". *Brandeis University Technical Report CS-05-259*.
[24] Bauer, E., Koller, D., & Singer, Y.: 1997, "Update rules for parameter estimation in Bayesian networks." In *Proceedings 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 3–13.
[25] Alterman, R.: 2000, "Rethinking Autonomy." *Minds and Machines*, 10:1 15-30.
[26] Tatar, D., Foster, G., Bobrow, D.: 1991, "Design for Conversation: Lessons from Cognoter." *Int. J. of Man-Machine Studies*. 34(2), 185-209.
[27] Gutwin, C., Greenberg, S.: 2002, "A Descriptive Framework of Workspace Awareness for Real-Time Groupware." *Journal of Computer Supported Cooperative Work.* 11:411-446.
[28] Dourish, P. and Bellotti, V.:1992, "Awareness and Coordination in Shared Workspaces." *Proceedings of CSCW*, Toronto, 107–114.
[29] Grudin, J.:1988, "Why CSCW Applications Fail: Problems in the Design and Evaluation of Organisational Interfaces." In *Proceedings of CSCW '88*, Portland, Oregon. 83-95. 30.Carroll, J., Neale, D., Isenhour, P., Rosson, M.B., McCrickard, D.S.:2003, "Notification and Awareness: Synchronizing Task-Oriented Collaborative Activity." *Int J. Human-Computer Studies,* 58, 605-632.
[30] Carroll, J., Neale, D., Isenhour, P., Rosson, M.B., McCrickard, D.S.:2003, "Notification and Awareness: Synchronizing Task-Oriented Collaborative Activity." *Int J. Human-Computer Studies,* 58, 605-632.
[31] Kirsh, D.:2001, "The Context of Work," *Human-Computer Interaction.* 16, 305-322.
[32] Kirsh, D.:2000, "A few thoughts on cognitive overload," *Intellectica, CNRS*, 30, 19–51.