Using Cognitive Artifacts to Bridge the Tool-Agent Divide

Joshua Introne, Richard Alterman
Mitchom School of Computer Science
Brandeis University
Waltham, MA 02454
(781) 736-3000
[jintrone, alterman]@cs.brandeis.edu

ABSTRACT

A fundamental hurdle in building intelligent systems is in acquiring enough information about a user and her context, in a form that the system can use, without impairing the usability of the system. A variety of interaction paradigms have been explored in this regard – agents, spoken language understanding, the incorporation of biometrics and other interaction modalities – that aim to make interaction more natural for the user. In contrast to these efforts, we have studied how structure can be added to a system to improve the user's performance in the domain task. In this paper, we demonstrate how this structure can then be leveraged to provide useful intelligent support. Our domain of investigation is a groupware system, and we show how structured support for co-referencing activity can be used by the system to infer user intent. We also provide empirical results demonstrating the effectiveness of an adaptive component that is built on the intent inferencing procedure.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: *Interaction styles (e.g., commands, menus, forms, direct manipulation.)* H.1.2 [User/Machine Systems]: *Human Information Processing*

General Terms

Algorithms, Design, Human Factors, Languages

Keywords

Distributed Cognition, Cognitive Artifacts, Intelligent Interfaces, Groupware.

1. INTRODUCTION

Despite more powerful computers, numerous advances in AI, and decades of research, a general approach to developing adaptive software – software that intelligently responds to the needs of its users to improve their ability to perform a task – remains an elusive goal for researchers and engineers. Indeed, these difficulties have led some to question the merits of adaptation itself [31]. In our view, though, adaptive systems often fail because user's reject the interface techniques that have been developed to get the information the system needs to act intelligently. Like others [4], we believe that the problem is not with adaptation per se, but rather with the interface between the user and the adaptive system.

The design of interfaces to intelligent systems is hard because it is subject to competing requirements. On one hand, many AI techniques are most effective with structured information about the user and her environment. From this perspective, interface facilities should be provided so that the user may express her needs to the system. On the other hand, the average user does not wish to be burdened with the task of encoding information so that the system can understand it [25]. From this perspective, the interface should be designed to make the user's domain task as easy as possible.

A variety of approaches have been developed within the research community to satisfy these requirements. The ubiquitous metaphor of computer as a collaborative partner in a joint task (e.g. [28]) may be seen as a way of encouraging the user to provide more information to the machine. Efforts in natural language understanding (e.g. [13]), and diagram understanding (e.g. [16]) are designed to make communication with collaborating agents easier for people, but these approaches do not scale beyond fairly narrow task definitions.

Some intelligent interfaces make opportunistic use of multiple information channels in order to improve inferences about the user without requiring that the user do more work. Biometrics have been exploited to help interpret language [6], and gestural information can be incorporated in multimodal interfaces to help interpret deixis [10]. Many of these systems perform well, but these extra information channels often come at a cost in the form of cumbersome sensor arrays or careful and brittle environment configurations.

Another research area seeks to leverage computer readable information that is available in the domain environment itself. For example, the semantic web is a large scale effort to encode more system-accessible information, in the form of ontologies, into a virtual user's domain. A related approach relies on external information resources that can be associated with the user's location, as in mobile applications, which use of embedded information servers and/or GPS input to retrieve location specific information (e.g. [11]).

While all of the above research shows promise, it does not leverage, in any systematic way, the privileged role of the computer as cognitive artifact [26]. It has been pointed out by many researchers (e.g. [7][20][22][29]) that people can and do use external tools to manipulate the information processing requirements in cognitive tasks. Checklists can be used as an external memory aid in safety critical applications [21], transforming an error-prone memory maintenance task into a simpler monitoring task. External devices (e.g. the Mercator projection chart [20], and nomographs [7] in general) can be used by people to support rapid and complex calculations. In everyday activity, people create and use such devices without any coaxing (e.g. grocery lists, lists of phone numbers); such artifacts are used to simplify work and improve performance.

Thus, cognitive artifacts structure information in ways that make it easier for people to perform specific tasks, and people willingly (even happily) use these artifacts. The critical point we make here is that this same structure can be a source of information that can be used by an autonomous process to provide intelligent assistance.

Here, we demonstrate how we can use the structured information in cognitive artifacts to provide intelligent user support in a collaborative system. First, we provide some of the background that informs our approach. We then introduce our experimental platform and an intent inference procedure that infers users' domain goals. Baseline effectiveness metrics are established for intent inference when complete and accurate information about objects in the domain is available. After that, we will explain some of the problems with accessing complete and accurate information at runtime. We then describe a cognitive artifact, which is shared between the users, and improves their performance in the domain task. We show how this artifact provides the system with a large portion of the domain information the intent inference mechanism requires, and additional information about users' co-referencing behavior that further improves intent inference. We then present an adaptive component that uses the output of the intent inference procedure, and describe empirical evidence documenting that it was adopted and that it was effective. Finally, we will discuss some related approaches to building adaptive software, and conclude with a summary.

2. BACKGROUND

The tension between the requirements for the design of interfaces to intelligent systems reflects a debate within the interface community. This debate was crystallized in the system as agent vs. system as tool debate (Maes vs. Schneiderman) [23]. From the agent perspective, the system should autonomously act on behalf of and with knowledge about the user. The agent is a knowledgeable collaborator to whom the user may delegate work, and the agent might even interrupt the user with critical information. From the tool perspective, a system should provide only a set of representations that make the domain easier to interpret and manipulate. Control should never be ceded to the system, and user interactions are primarily focused on the domain, rather than the system itself.

The mixed initiative approach has been suggested as a response to this debate. Horvitz [19] suggested several issues that should be addressed in combining AI and direct manipulation. The thrust of these suggestions is that an agent in a mixed initiative system should take into account the uncertainty in the inference process, as well as the user's social and task contexts, in making a decision about whether or not to take some action. At a certain level, the expected utility of an interaction between the user and agent warrants a direct engagement.

A variety of agent-based, mixed-initiative systems (e.g. TRAINS, [14]; COLLAGEN [28]; Lumiere [18]) have been developed, however it has not been conclusively demonstrated that users prefer to interact directly with an agent, regardless of how careful we are about interrupting with less than perfect information. Interface agents are rarely seen in mainstream software; the much-maligned Microsoft Office Assistant is perhaps the only example casual computer users can point to. We suspect that the notion of a dialogic interaction itself may be a source of the problem.

We are interested in finding ways to introduce intelligent assistance to the user without invoking the agent metaphor. Cognitive artifacts offer a potential solution, because they introduce structure into the interface that makes a user's task easier, and can also be exploited by AI algorithms.

2.1 Distributed Cognition & Collaboration

The field of distributed cognition (e.g. [20][29]) documents how people use external artifacts to enhance task performance in particular domains. We seek to leverage the useful structure that such artifacts provide to create powerful adaptations (see Alterman[1] for a theoretical introduction to this approach).

The development of cognitive artifacts to support specific and complex coordination as part of an established work practice has been well documented in the literature ([34][17][20][24]). In addition to their role in transforming a user's cognitive task, these artifacts have several coordination aspects [31]. They are external representations that are distinct from the work domain itself. They lend persistence and structure to information, and make it easier to share this information among multiple collaborators. Finally, they are associated with a protocol that typifies their use within a particular work practice. We refer to these artifacts as "coordinating representations" ([3], c.f. Suchman & Trigg[34]).

In previous work [12], we have described a methodology by which coordinating representations (CRs) may be developed based on an ethnographic analysis of an existing practice. Through this methodology we identify "weak spots" in the representation system that reveal themselves as coordination breakdowns and emerging conventions. Here, we describe how a CR that has been developed via the application of our methodology can be leveraged to add intelligent support to an existing application.

3. EXPERIMENTAL PLATFORM

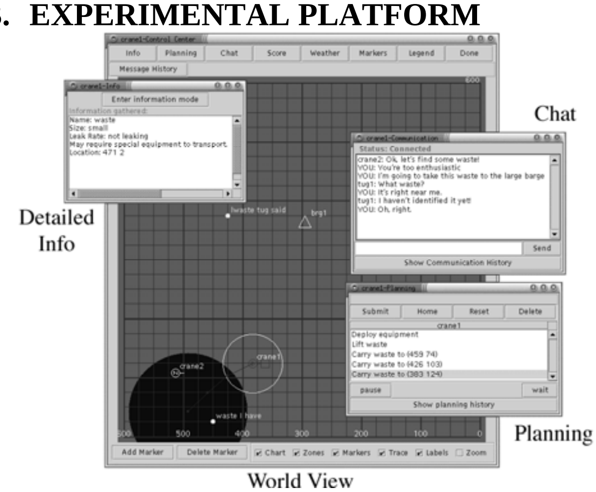

Figure 1 The VesselWorld System

Our experimental platform is a groupware system called VesselWorld, shown in Figure 1. To support analysis, VesselWorld logs complete interaction data that can be used to "play back" user activity. VesselWorld was demonstrated at CSCW 2000.

In VesselWorld, three participants collaborate on remote workstations to remove barrels of toxic waste from a harbor. Each participant is the captain of a ship, and their joint goal is to remove all of the barrels from the harbor without spilling any toxic waste. Two of the users operate cranes that can be used to lift toxic waste barrels from a harbor and load them onto a large barge (which has a fixed position). The third user is the captain of a tugboat that can be used to drag small barges (which can be moved from place to place) from one place to another. The crane operators can load multiple wastes on the small barge, and can unload them later.

The progression of a VesselWorld session is turn-based, such that every user must submit a step to be executed by the server before the server can evaluate executions and update the world on each client screen. Users may plan any number of steps in advance, although any plan steps that involve objects are restricted to those objects that are currently visible, and only one step can be submitted to the server at a time. Communication may occur at any point, but all communication must occur through a text-based chat window that is part of the system. The users are scored by a function that takes into account the number of steps it takes to remove all of the waste, the number of barrels cleared, the number of errors made, and the difficulty of the problem.

Several complicating factors make coordination between participants a necessary part of solving a VesselWorld problem. Each ship has a geographically limited view of the harbor; thus ships in different locations will have different directly-observable domain information, and no player has prior knowledge about how many or where waste sites are. The toxic waste barrels are of different sizes, which entail different coordination strategies that may involve more than one of the actors. For example, a single crane may lift a small or medium barrel, but the 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 tugboat operator. Toxic waste barrels may require specialized equipment to be moved, and the cranes carry different types of equipment. Finally, the tugboat operator is the only actor who can determine the type of equipment a toxic waste barrel requires.

3.1 Intent Inference in VesselWorld

Planning in VesselWorld is a laborious and error prone operation [2]. User errors are often due to forgotten plan steps or joint plans that have become unsynchronized. Automatic plan generation could overcome some of these problems. However, over a hundred goals may be possible for each user at any given time, depending on the number and types of objects in the world. To avoid requiring that users specify their goals manually, intent inference was used to reduce the number of possible goals to a manageable list that users could select from.

To infer user intent we employed an intent-inference procedure based on Bayesian Networks (BN). Two BNs were developed that assess likelihoods for Crane and Tug operator intentions, respectively. At runtime, evidence about the state of the world is posted to these models for each agent-waste pair separately. The likelihood an agent has an intention with respect to a given waste is read off one of the nodes in the network, and the intention with the highest value for a given agent-waste pair is taken to be the most likely intention. New evidence is posted whenever a relevant change in the world is detected.

In this paper we restrict our analysis to the portion of the Crane network that predicts Crane lift intentions. This BN is shown in Figure 2; it models the likelihood that an actor has the intention to lift (or jointly lift with the other crane operator) a specific toxic waste based on information about the state of the world, including:

- The type of equipment required.
- The size of the waste (which determines whether a single crane can lift the waste, or if it need the support of another crane).
- Whether the cranes are close to or heading towards the waste.
- If the crane actor is currently holding a waste

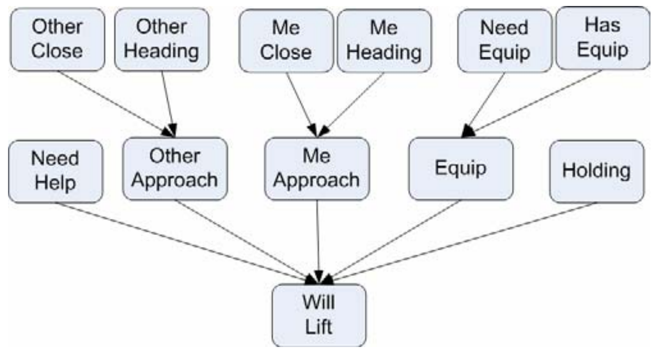

Figure 2 Schematic of BN used to infer Crane Lift Intentions

3.1.1 *Evaluating Intent Inference*

We are interested in how the performance of the intent inference procedure varies when different sources and types of input information are used. Here, we provide results on the performance of the procedure using "perfect" information about the simulated world – specifically, where each waste is, what size it is, and what type of equipment it requires are all known at the outset. The reason we evaluate the model with perfect information is to provide a basis for comparison. Perfect information is not usually available, and in following sections we will compare performance with perfect information to performance with imperfect information.

Table 1 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

To derive the conditional probability tables for the BN, we trained it using the parameterized EM learning algorithm [4] (η=1.8, and the network was trained until $\delta \leq .001$), and tested the network on the same data set, summarized in Table 1. To obtain the perfect state information required for this analysis, we were able to use the problem files that were used to initialize each user session.

To evaluate the performance of the network, we calculate the proportion of correctly guessed goals, or correct goal rate (CG_R); and the proportion of guesses that were false, or the false positive rate (FP_R). We count any uninterrupted sequence of correct guesses – recall, a guess is made whenever a relevant state variable changes – leading up to the step immediately preceding the execution of the predicted goal as a single correct goal. The total number of goals is the number of wastes lifted. Thus,

$$CG_R = \textit{correct goals} / \textit{total goals}$$

$$FP_R = \textit{incorrect guesses} / \textit{total guesses}$$

Our results (shown in Table 2) indicate that user intent can, in general, be reliably inferred in our domain using perfect information about the world. However, there are clear differences between groups. Closer inspection reveals a relationship between the number of wastes and the performance of the algorithm. There is a weak inverse correlation between the number of wastes in a problem and the correct goal rate (r=-.29), and a stronger positive correlation between the number of wastes and the false positive rate (r=.56). Hence, the variability between groups may be partially explained by the fact that it easier to guess a user's intentions when there are fewer possibilities to pick from, and groups 3 and 4 had more wastes per problem.

Table 2 Performance of Intent Inference with Perfect Information

Group	CG_R (StdDev)	FP_R (StdDev)
Group 1	.91 (.12)	.46 (.13)
Group 2	.91 (.10)	.47 (.13)
Group 3	.83 (.14)	.57 (.12)
Group 4	.77 (.15)	.56 (.12)
Average	.83 (.14)	.53 (.13)

The values in Table 2 establish baseline performance for the developed network when the system has access to perfect information about objects (toxic waste barrels) in the world. In most applications, though, perfect domain information is not available. In the next section, we will describe some of the problems in getting this information from VesselWorld at runtime (without using the problem files). We will then show how we can use information from a cognitive artifact to do nearly as well as with perfect information. We will also show how additional information made available by the artifact can be used to improve intent inference even if perfect information were available.

4. OBTAINING STATE INFORMATION

The problem files are the only source of complete, correct, and structured information about the toxic waste barrels in VesselWorld. Some information about toxic waste barrels can also found in conversations between users in chat, but it is unstructured, and hence difficult for the system to access.

An excerpt from chat during a typical planning session shown in Figure 3 demonstrates this. 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 operator replies (having apparently already investigated that toxic waste barrel) with corrected coordinates (105, 420) and specific information. In line 8, Crane2 thanks the Tug operator for the clarification, and the Tug closes the conversational turn in line 9.

Automatically extracting information about toxic waste barrels, which is required by our intent inference procedure, from chat logs would be very difficult; the above dialogue illustrates some of these problems. The dialogue occurs between 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 automatic reference resolution algorithms.

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 3 Excerpt from chat during VesselWorld session

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, we will discuss in the next section why it is reasonable for our domain, and describe empirical evidence supporting this claim.

4.1 Coordinating Representations

As we have discussed, people develop and use external structured representations to enhance their ability to perform complicated, error-prone, laborious, or critical activities. There is no well-defined methodology describing how to develop these representations in the general case. However, we have developed a methodology that allows us to do this for the specific case of collaborative applications [12]. Our method is based upon the ethnographic techniques of Suchman & Trigg [34] and Hutchins [20], and the discourse analysis methods of Sacks, Schegloff, and Jefferson [29].

The incorporation of coordinating representations into VesselWorld system was our solution to several problems uncovered in our analysis of online user behavior. One such problem was the users' difficulty in managing information about domain objects. Some of the groups handled these difficulties by developing mnemonic expressions for referring to domain objects; other examples of this kind of co-referencing behavior has been documented elsewhere (cf. [9]). However, users did not always agree on consistent mnemonics, and coordination errors in the maintenance of this information were frequent. Thus, one of the CRs that was introduced to the VesselWorld system was designed to support the organization and naming of objects in the world. We call this CR the Object List (Figure 4).

The Object List is a tabular WYSIWIS (What You See Is What I See) 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, including a user assigned name, the status, and the location of the associated object. The location field may be filled in by clicking on the field and then on the object in the interface (and hence has fixed structure). The size, equipment, action, and leak fields are filled in using drop-down menus. A free text field ("notes") is also provided for each entry so that any other relevant information 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.

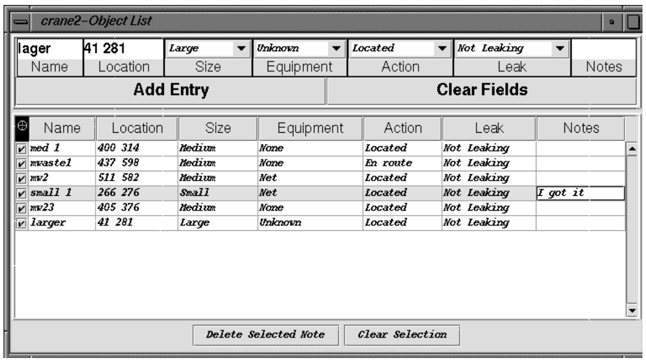

Figure 4 The Object List CR

In studies published elsewhere ([2][3]), we have found that the Object List, and other CRs, were used, and that they significantly improved user performance. Our studies revealed that the CRs reduced errors and time spent chatting, and on average halved the time it took to solve problems.

Another important feature that occurs in groups that used the Object List is that people almost always use the names that they have assigned to objects with the Object List to refer to objects in the world while chatting. This is perhaps an obvious occurrence in hindsight, but it provides the system with a significant piece of information for inferring intent.

5. INTENT INFERENCE FROM USER SUPPLIED INFORMATION

In using the Object List, users provide the system with some portion of the state information that our intent inference procedure requires. This information is of course not perfect – it is only revealed to users (and entered into the Object List) as they discover and examine wastes, and it is subject to errors, omissions, and duplication – but it can be posted directly to the BN above. The Object List also provides us with a set of references to objects in the world that can be used to mine chat for clues about user intentions. We were able to incorporate this information into our BN in a straightforward manner.

We found that the occurrence of references (labels assigned to wastes in the Object List) in chat were predictive of lift actions for roughly a fifteen-minute window of time preceding a lift. Table 3 depicts the likelihood that a reference for an object will appear in chat for the three consecutive five minute windows prior to a lift of that object at time *t*, based on a frequency analysis of the data. In the table, "Joint" and "Single" refer to whether or not a waste requires both or just one crane operator to lift. In the *~Lift* conditions, values reflect the likelihood some other waste is referred to prior to a lift of some waste.

Table 3 Probability of reference preceding a lift at time *t*

	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

Users in VesselWorld often refer to wastes in chat using the labels they've assigned in the Object List; as shown in the table, there is about a sixty percent chance that waste will be referred to in chat in the five minutes preceding the lift if that waste requires assistance, and about a forty percent chance if that waste can be lifted singly. Outside of a fifteen minute window, references were not a very good predictor of lift actions. On the basis of this analysis, we expanded our BN to include three five minute windows of chat history, with one node for each five minute window. These nodes are not assumed to be independent, allowing the EM algorithm to identify relationships between windows.

5.1 Evaluating Intent Inference with User Supplied Information

We compared the performance of our intent inference procedure across four conditions:

- Perfect Info – All information about toxic waste barrels (size, location, equip) 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; subject to user errors.
- Object List + Chat – The Object List condition, plus the occurrence of references in chat.
- Perfect Info + Chat – The Perfect Info condition, plus occurrence of references in chat.

These results are shown in Table 4. As before, we are interested here in how well the data and model predict the Cranes' lift intentions, rather than demonstrating the generality of the model itself. Therefore, we trained each of the four networks, as described previously, separately for each condition. The same starting parameters were used in each case (identical portions of each network had the same starting conditional probability tables), as the EM algorithm is sensitive to starting parameters.

Table 4 Intent inference results for different info sources

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

For each pair of conditions that are compared in the following, a two-tailed, paired t-test was used to determine the significance of the overall differences, and these values are reported.

As expected, the intent inference procedure performs significantly better (p<.001 for both CGR and FPR) with perfect information than it does with information from the Object List. However, it still performs quite well – seven out of ten lift intentions are predicted accurately (on average) vs. eight out of ten when working with perfect state information. The addition of reference information in chat improves upon this even further. The overall correct goal rate in the Object List + Chat condition is improved by .07 over the Object List condition (p<.001), although the change in false positive rate is not quite a significant effect (p = .058). The performance differences between the Object List + Chat and Perfect Info conditions are still significant (p<.05 for CGR, p=.058 for FPR).

The reference information made available by the Object List also improves intent inference when combined with perfect state information. The difference between the Perfect Info + Chat and the Perfect Info conditions are significant (p<.01 for CGR and p<.05 for FPR). Thus, regardless of our access to state information (for instance, if we had intelligent sensors placed in the world) the Object List introduces a source of information that further improves our ability to infer user goals.

We will now show that the level of inference provided by the procedure described above, using information from the Object List and chat, is good enough to support useful intelligent assistance at runtime.

6. AN ADAPTIVE COMPONENT

In the preceding sections, we have demonstrated several things. First, we have verified that our intent inference technique is valid; it performs well with access to perfect information. We have also shown how intent inference with the domain information performs nearly as well. Finally, in using this CR, users share a set of referring expressions with the system, and we have shown that this information can be used to further improve intent inference.

To provide users with planning assistance that makes use of the intent inference procedure, we developed a WYSIWIS component (Figure 5) to present the five most plausible goals output by the intent inference procedure to each user at any point in time.

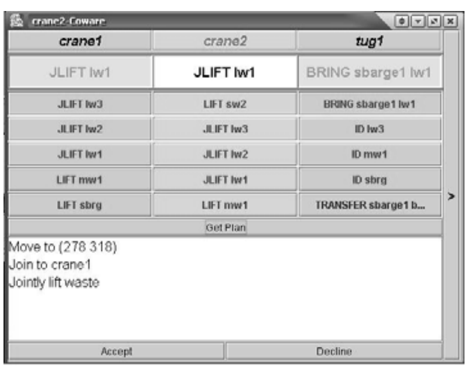

Figure 5 The adaptive component

The function of this 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 offers each user up to five possible goals.
2. When a user selects a goal, it is displayed so that all users can see it. The user that selected the goal has the option to request an automatically generated plan for the goal.
3. The system generates a plan that the user can inspect. If the goal involves multiple actors, the other involved actors are invited to join the plan. If all invited actors accept the invitation, a plan is generated; if invited users do not accept the invitation, the requesting user is so informed.
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.

6.1 User Studies

To evaluate the effectiveness of the above component, we performed a 40-hour study with four teams of three people. 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 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 adaptive 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 time performance of the users had stabilized.

The component was used

All groups used the component to generate plans within the system. On average, 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.

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 operator 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").

The component reduced errors

The groups that had the component had 45% (p=.069) fewer joint errors (failures during joint actions) per minute than the groups that did not. This difference is not significant at the .05 level, because of the small sample size and overall low proportion of joint errors. A reduction in joint errors corroborates prior analysis of use of the VesselWorld system [1], 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.

The component reduced cognitive effort

To measure the change cognitive effort between the two populations, we examined the amount of interface work together with the amount of time it took users to execute plans. We found that the amount of clock time taken by users between submitting steps of automatically generated plans was 57% less ($p<.01$) than in groups without the adaptive component, and we also found no significant differences 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 effort of the collaborators.

In summary, our studies indicate that the adaptive component was heavily used, improved user performance, and made plan execution easier. These results demonstrate that collaborating users can generate enough structured information when using a shared cognitive artifact to drive useful intelligent support. We conclude that the approach to adding intelligent support demonstrated here was successful for this domain. Our approach can be generalized to other domains; in the following, we will describe some related work that illustrates this.

7. DISCUSSION

In this and other articles ([2][3][12]), we have outlined a methodology for adding adaptive support to a system by introducing useful structure at the interface. While the description of this methodology is novel, other researchers have leveraged the techniques we have described in building adaptive systems. These other systems point the way to the generality of our approach. For example, Pu & Lalanne [27] describe several information visualization techniques that can be used as a front end to various CSP algorithms. In these systems, the interface representations help people to think about a constraint solving task and to guide the underlying algorithms, while these algorithms explore the problem space and present results via the same representation. As described by the authors, the interface representation is informed by an understanding of the abstract representation used by the underlying CSP algorithms. Thus, the system's internal representation is brought to the interface so that it is accessible to the user. In contrast, we have brought information from the users' practice into an interface representation that the system can use. However, both approaches leverage structure in the interface to satisfy the competing requirements on the interface representation.

St. Amant [33] presents a framework for studying the tradeoffs between *interpretability* and *efficiency* in intelligent systems. These qualities are very similar to the dual interface requirements we introduced at the beginning of this article. Efficiency refers to the amount of interface work a user must do to specify and evaluate system states. Interpretability "is the ability of the system to infer the relevance of information and actions to tasks, within and between states," or the simplicity with which the system can interpret the user's actions. St. Amant sketches an approach to analyzing interpretability of a system, and in so doing highlights the notion that some kinds of structure (as opposed to agent based dialogic interaction) may be used to simplify the identification of the user's needs. While we feel that this structure should be informed by an analysis of a particular work practice, there is much utility in understanding how different kinds of structure can be leveraged by autonomous algorithms.

In collaborative environments, the application of coordinating representations to provide intelligent support has a rich history. The COORDINATOR [15] applied structure to inter-office messages and related these messages to transactions in a network of speech-acts. The system used this information to provide users with reminders and identify those who defaulted on commitments. Similarly, the Information Lens [24] added structure to messages to support better filtering and rule-based processing for end-users. Our work differs from these earlier systems in our strong ethnographic basis for the introduction of cognitive artifacts.

The computer supported problem-solving environment (CPSE) described by Chin et al. [8] was developed in a manner similar to ours in this regard. Multiple coordinating representations were developed by and for a community that used a large scale distributed computing environment. These representations were based not on theory, but actual user experiences with the system. For example, many scientists planned their own experiments on paper as workflow diagrams. A workflow based experiment interface was thus incorporated into the system. This structure made it easier for scientists to coordinate the use of many available distributed resources, and helped them to plan and monitor the progress of large scale experiments. Although the prototype system did not offer any automation, the authors recognized the possibility of leveraging structure in the workflow representation for that purpose. The prototype CPSE exemplifies the methodology we have described in our work.

8. CONCLUSION

As we have discussed, the design of interfaces to intelligent systems is hard because it is subject to competing requirements. These requirements are created by the system's need for information about the user and context at runtime, and the user's need for a fluid interaction with the domain. Here, we have described how both of these requirements can be addressed by introduction cognitive artifacts. On one hand, such artifacts provide the users with structure that improves their cognitive task; on the other, this structure can be used by the system to interpret user actions and provide intelligent assistance. We have demonstrated a methodology for the development of such artifacts, called coordinating representations, in collaborative interfaces.

In this paper, we have shown how one type of CR, which supports labeling and organizing information about objects in a shared domain, can be used to support intent inference. Not only does this CR provide the system with access to domain information that might not otherwise be available, but it also provides the system with access to referring expressions that people use in natural dialogue. This later datum is used naïvely here to improve intent inference; other technologies may be able to derive even further mileage by interpreting the human dialogue that occurs around these referring expressions. For the system described here, our intent inference procedure was powerful enough to drive an adaptive component that significantly improved the users' task.

There is a need to develop both an approach and a catalogue of representations that can be applied to single-user as well as collaborative user interfaces. In this article, we have referred to several sources from which to draw insight, and there are many other sources that are not discussed here. We hope that others will join us in drawing upon this fount to develop new approaches for integrating AI and human activity.

9. REFERENCES

[1] Alterman, R. Rethinking Autonomy. In *Minds and Machines*, 10(1), 2000, 15-30.

[2] Alterman, R., Feinman, A., Landsman, S., Introne, J. *Coordination of Talk: Coordination of Action*. Technical Report CS-01-217, Department of Computer Science, Brandeis University, MA, 2001.

[3] Alterman, R., Feinman, A., Landsman, S., and Introne, J. Coordinating Representations in Computer-Mediated Joint Activities. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, 2001, 43-48.

[4] Bauer, E., Koller, D., & Singer, Y. (1997). Update rules for parameter estimation in Bayesian networks. In *Proceedings Uncertainty in Artificial Intelligence (UAI)*. 1997, 3–13.

[5] Birnbaum, L., Horvitz, E., Kurlander, D., Lieberman, H., Marks, J. Roth, S. Compelling Intelligent User Interfaces: How Much AI? In *Proceedings of IUI '97* (Orlando, FL, January 1996). ACM Press, NY, 173-175.

[6] Bosma, W., and André, E. 2004 Exploiting Emotions to Disambiguate Dialogue Acts. *Proceedings of IUI '04*. (Funchal, Madiera, Portugal, 2004). ACM Press, NY, 85-92.

[7] Card, S.K., Mackinlay, J.D., and Shneiderman, B.(eds.) *Readings in Information Visualization-Using Vision to Think*. Morgan Kauffman Publishers, San Francisco CA, 1999.

[8] Chin Jr, G., Leung, L. R., Schuchardt, K., and Gracio, D. New paradigms in problem solving environments for scientific computing. In *Proceedings of IUI'02,* (San Francisco, CA, 2002). ACM Press, New York, pp. 39-46.

[9] Clark, H. H., and Wilkes-Gibbs, D.. Referring as a collaborative process. In *Cognition* , 22, 1986. 1-39.

[10] Kaiser, E. 2005. Multimodal New Vocabulary Recognition through Speech and Handwriting in a Whiteboard Scheduling Application. In *Proceedings of IUI'05* (San Diego, CA, 2005). ACM Press, New York, 2005, pp. 51-58.

[11] Krüger, A; Butz, A; Müller; Stahl, C.; Wasinger, R.;Steinberg, K; Dirchl, A. The Connected User Interface: Realizing a Personal Situated Navigation Service. In *Proceedings of IUI'04.* (Funchal, Madiera, Portugal, 2004). ACM Press, NY, 2004, 161-168

[12] Feinman, A., and Alterman, R. 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*, (Johnstown, PA, 2003), 228-237

[13] Ferguson, G. and Allen, J. TRIPS: An Intelligent Integrated Problem-Solving Assistant, in *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (AAAI-98), (Madison, WI. 1998) pp. 567-573.

[14] Ferguson, G., Allen, J., Miller, B. Trains-95: Towards a mixed initiative planning assistant. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems,* (1996) 70-77.

[15] Flores, F., Graves, M., Hartfield, B., Winograd, T. "Computer Systems and the Design of Organization Interaction," *ACM Transactions on Office Information Systems*, 6:2 (1988), 153-172.

[16] Forbus, K., Ferguson, R., and Usher, 2001. Towards a computational model of sketching. In *Proceedings of IUI'01*, (Santa Fe., NM., January 14-17, 2001). ACM Press, New York, 2001, 77-83.

[17] Goodwin, C. and Goodwin, M. Formulating Planes: Seeing as a situated activity. In *Cognition and Communication at Work,* edited by David Middleton and Yrjö Engestrom. 1996. Cambridge: Cambridge University Press, pp. 61-95.

[18] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., Rommelse, D. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users, In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (Madison WI, July 1998). Morgan Kaufmann Publishers, 256-265

[19] Horvitz, E. Principles of Mixed-Initiative User Interfaces. In *Proceedings of CHI' 99*: (Pittsburgh, PA, 1999) ACM Press, NY, 159-166.

[20] Hutchins, E. *Cognition in the wild.* MIT Press, Cambridge, MA, 1995.

[21] Hutchins E. 1995. How a cockpit remembers its speeds. In *Cognitive Sciience*, 1995;19:265–88.

[22] Lave, Jean, Michael Murtaugh, and Olivia de la Rocha. "The Dialectic of Arithmetic in Grocery Shopping." In *Everyday Cognition: Its Development in Social Context*, edited by Barbara Rogoff and Jean Lave, pp. 67-94. Cambridge, Mass.: Harvard University Press, 1984

[23] Maes, P., and Shneiderman, B., Direct Manipulation vs. Interface Agents: A Debate. *Interactions*, Vol. IV Number 6, ACM Press, 1997.

[24] Malone, T.W., Grant, K.R., Lai, K-Y., Rao, R., and Rosenblitt, D. Semi-Structured Messages are Surprisingly Useful for Computer-Supported Coordination. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'86)* (Austin, Texas, Dec.). 1986, pp. 102-114

[25] Nardi, B. *A Small Matter of Programming*. The MIT Press. Cambridge, MA. 1993.

[26] Norman, Donald A. Cognitive Artifacts. In J. M. Carroll (ed.): *Designing Interaction.Psychology at the Human-Computer Interface*. Cambridge University Press, Cambridge, 1991, pp.17-38.

[27] Pu, P. and Lalanne, D. Design Visual Thinking Tools for Mixed Initiative Systems, in *Proceedings of IUI'02*, (San Francisco, CA, 2002) ACM Press, New York, 2002. 119-126.

[28] Rich, C. and Sidner, C. COLLAGEN: a collaboration manager for software interface agents. In *User Modeling and User-Adapted Interaction*, 8(3/4). Kluwer Academic Publishers, Hingham, MA. 1998. 315-350.

[29] Rogers, Y. and Ellis, J. Distributed cognition: an alternative framework for analyzing and explaining collaborative working. In *Journal of Information Technology* 9, 1994, 119-128.

[30] Sacks, H., Schegloff,, E., Jefferson, G.: A simplest systematics for the organization of turn-taking for conversation. In *Language*, 50 (4) (1974), 696-735.

[31] Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 1998.

[32] Schmidt, K. and Simone, C. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. In *Journal of CSCW*, 4(2-3), 1995, pp. 155-200.

[33] St. Amant, R., Dinardo, M., and Buckner, N. Balancing Efficiency and Interpretability in an Interactive Statistical Assistant. In *Proceedings of IUI'03.* (Miami, FL, 2003). ACM Press, New York, 2003. 181-188.

[34] Suchman, L. and Trigg, R. 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, February. 1992. 65-90.