Supporting Intent Awareness in Groupware

Joshua Introne and Richard Alterman
Volen Center for Complex Systems
Brandeis University
[jintrone, alterman@cs.brandeis.edu]

Abstract

In this paper, we present experimental results that demonstrate the effectiveness of a mixed-initiative component, which we call a Coware component, that is designed to support awareness and planning in a collaborative domain. The Coware component embodies a design methodology for supporting collaboration in groupware that leverages work users do to stay coordinated to drive an intent inferencing procedure. Intent inference is in turn used to drive the mixed initiative Coware component, which provides users with high individual payoff via automatic plan generation, and supports mutual intent awareness.

We will provide a summary account of our intent inference framework, which uses: 1) information collected via Coordinating Representations, which are interface components designed to structure and facilitate the exchange of coordination specific information, and; 2) a probabilistic task-domain model represented as a Bayesian Belief Network. The design of the Coware component is discussed in depth, and an analysis of empirical results from usage studies is presented.

1 Introduction

When networked computers are interposed between people for tasks that require coordination, communication channels that typically support the unfolding joint activity become severely constrained, or vanish completely. Visual and contextual queues, and even direct voice chat may not be available. Many approaches have been taken to building groupware systems that attempt to provide means for users to overcome these deficits. Early techniques, which had to contend with more severe technological constraints (limited bandwidth and processing power), attempted to provide external structure (turn-taking, speech-act commitment networks [Flores, et. al.]) to replace the need for real-time information of others' activities. In general, these approaches were not successful, partly because collaborative activity is typically more fluid and dynamic than enforced external structure can allow for. More recently, several [Dourish and Belotti, 1992, Gutwin, et. al. 1996] have turned their attention to replacing the information channels that are missing in groupware systems; in general, these approaches employ interface components that support *awareness*. While these approaches have met with far more success, they have typically been limited to reporting information about location (in a shared workspace) and activities of other users at some level of abstraction. This is not, however, always the right kind of information, nor is it always feasible to provide enough detail to support the kind of reasoning necessary for people to coordinate effectively. More specifically, the ability to anticipate the actions of others, which is crucial for effective coordination, is not well supported.

One obstacle to providing collaborators with direct support for anticipatory reasoning is that it is difficult to elicit appropriate information about *intent* from users. [Dourish and Bellotti, 1992] explain this difficulty as a general problem with "active, informational" awareness mechanisms, which require users to report information for the benefit of the group. The three problems they identify are; 1) there is an apparent imbalance between cost and payoff for the individual; 2) information provided by the individual may not take into account the context of the recipients, and may not be timely or relevant; 3) recipients of this information incur cost in determining its relevance.

To address the problem of eliciting intent, we apply an approach called Non-Autonomous AI [Alterman, 2000], which allows us to perform imperfect intent inference by using coordination work that users must do anyways. Intent inference is used to drive a mixed-initiative groupware component, called Coware (Collaborative Awareness), that can offer semi-automatic planning as well as a means for users to maintain awareness of each others intentions.

Our approach can be summarized as follows:

- *Users must exchange information to stay coordinated in joint activities.*
- *Collaboration via networked computers makes the exchange and maintenance of some classes of this information difficult.*
- *Interface components called Coordinating Representations (CRs) [Alterman, et. al. 2001b] may be introduced to support the information requirements for difficult coordination tasks.*
- *Data collected from the users through these CRs is well suited to support a mixed initiative intent inferencing procedure that can in turn be used to drive automatic planning.*

Coware provides users with a list of possible intentions, and selecting from this list allows the system to automatically generate relevant plans for users. In making such selections, users implicitly confirm their intentions in a component that is visible to all users. Thus, we expect that Coware can ultimately be used to overcome the problems identified in [Dourish and Belotti, 1992]. We revisit this in the discussion section below.

Our approach is generally applicable to collaborative applications. First, we employ discourse analysis techniques [Afeinman and Alterman, 2003; Cohen, 1997] to identify coordination problems in an existing collaborative system. Coordinating Representations [3] that are suggested by an initial analysis of practice with the existing system are designed and integrated into the system. CRs typically lend structure to information that is communicated by users to support coordination. Our intent inferencing technique utilizes Bayesian Belief Networks [Pearl, 1988; Jenson, 1996] to convert this type of information, along with a task-domain model, into predictions about the intended goals of the users. Finally, we develop Coware components, a class of mixed-initiative Coordinating Representations, which use the output of the intent inferencing system to provide users with awareness of others' intents. A key feature in the design of these components is that they do not require perfect intent inferencing to provide users with a high degree of utility.

In this paper we will focus on an empirical evaluation of an implemented system that employs the methodology described above. We will show that the evidence collected supports the non-autonomous claim; that we can leverage user work to make intent inference practical and useful. Specifically, our evidence shows that; 1) Users use Coordinating Representations; 2) Information from Coordinating Representations can be leveraged to provide good predictions about user intentions; 3) Users use these predictions during a problem-solving session to support their activity.

In following sections, we describe our domain, and detail the non-autonomous process as it has been employed here. We will then describe the design of the intent inferencing algorithm and the Coware component. In the final sections, we provide detail about our user study and present evidence that was collected. We conclude with a discussion of the results.

2 The Vesselworld Domain

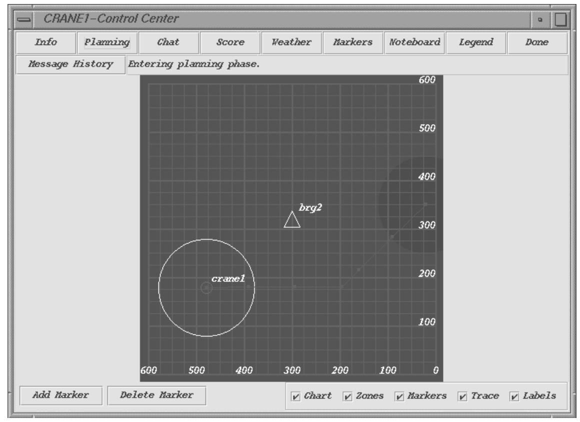

Figure 1 The Vesselworld Control Center

The Vesselworld system, which was demonstrated at CSCW 2000, is a collaborative system we have built for the purpose of developing a design methodology and adaptation techniques for collaborative systems. Figure 1 is a snapshot of the main window (the Control Center) in Vesselworld for one of the ship captains.

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. The users are scored by a function that takes into account the time it takes to remove all of the waste, the number of barrels cleared, the number of errors made, and the difficulty of the problem. Each ship has geographically limited view of the harbor, and thus ships in different locations will have different directly observable domain information. There are varying types and sizes of toxic waste barrels, which entail different coordination strategies that involve more than one of the actors. Each ship has some unique capabilities, which determine the type of toxic waste it is able to remove from the harbor.

The progression of a Vesselworld session is turn based, thus 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 tool or one of the special purpose CRs.

Roughly three hundred hours of data have been collected with the Vesselworld system, and we have developed a suite of domain independent tools for analyzing this data. In data obtained from experiments with an initial version of the system that offered only the chat tool for communication, we identified several classes of coordination problems [Alterman, et. al., 2001a; 2001b]. In the next section, we describe how non-autonomous AI was employed to alleviate some of these errors.

3 Non-Autonomous AI

Introducing AI technology into the interface is rarely a matter of simply plugging a module into an obvious socket. The appropriate contextual information that is necessary to support any technique must be identified, converted into a form usable to the technique under consideration, and in most cases, there's an additional unspecified something that gets thrown into the mix to make everything work. This argument is presented in detail in [Alterman, 2000].

Collaborative applications offer us an excellent opportunity to employ various AI techniques, because users must exchange certain types of domain specific information in order to stay coordinated. Our insight is that it is possible to both reduce the amount of work that users must do to communicate and maintain some types of coordination information, and at the same time convert this work into data that is easily used by general AI techniques [Alterman, 2001a]. To do this, we introduce a class of interface components that are called Coordinating Representations, which offer convenient and useful structure that users may take advantage of.

3.1 Coordinating Representations

Coordinating Representations (CRs) are artifacts that are designed to explicitly support the communication and management of specific types information that are necessary for actors to stay coordinated. The information in CRs may be entirely user-authored, or (as in the case of Coware) may be in part generated by the system.

CRs that are user-authored generally offer the user several structured or semi-structured fields, such as drop down lists and fields that are drag targets for other objects in the interface. If appropriately designed, this type of interface reduces the amount of work users need to do to communicate the type of information supported by the CR. The structured data that is generated during use is readily accessible to general AI algorithms.

One type of CR that was developed for Vesselworld was the Object List. In empirical studies the Object List was heavily used and led to reduced communication via the chat tool, and significantly improved performance in the domain task [Alterman, 2001a].

3.2 Object List

The Object List is shown in Figure 2. It is an external representation that structures and distributes memory for information about shared domain objects. The Object List replicates the same data in tabular format for each user, and modifications to data made by one user are displayed globally. Each row of user-entered data in the Object List contains several fields of information, including a user assigned name, the status, and the location of

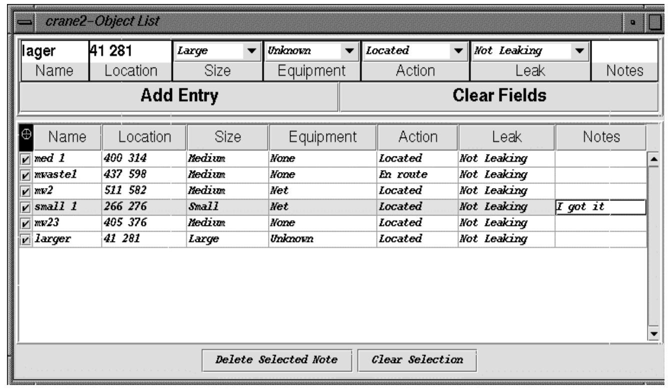

Figure 2 The Object List

the associated object. A free text field is also provided for each entry so that any other relevant information may be communicated. Icons representing objects with valid locations may be displayed in the Control Center interface, to assist users in mapping items in the list to objects in the domain.

Usage of the Object List generates tagged information about the perceived state of the world, and Vesselworld logs this along with other usage data. In all experiments, the Object List was heavily used, and the majority of users enjoyed using it [Alterman, 2001a]. This information provides us with both a dynamic representation of the user-perceived domain state as well as a set of references users employ during chat to discuss plans and goals concerning objects (this is not enforced, but users generally use the names entered into the Object List to refer to objects in chat). As this data is entirely user constructed, it is exceptionally well tuned to reasoning about intention in the space of user awareness. In the following section we describe how this data can be harnessed to do intent recognition that achieves better performance than would otherwise be possible.

3.3 Intent Inference

Our approach to intent inference uses the coordination work users must do to generate more accurate intent inference than would otherwise be possible. The Object List CR provides two readily available pieces of information that we can use in our intent recognition system. First, the Object List provides a list of active user references to objects in the domain, and these references are often used in chat to discuss plans for objects. Information about the domain as the users perceive it is also provided, including: where each object is; what type of equipment it requires, and; and how large each object is (and consequently its coordination requirements). Our goal is to provide accurate predictions about the toxic waste the users intend to address next in plan execution. In preliminary studies, references in chat to objects in the domain were tested as predictors of plans involving those objects, without including the domain information from the Object List. We found chat references to targets were good predictors of target plans in the five minutes preceding plan execution, especially where the target waste involved significant coordination. However, many references occurred in general throughout the logs, leading to very high false prediction rates. One possible approach to reducing the false prediction rate would have been to do a semantic analysis of the chat to determine which references were the objects of planning conversation. However, the additional information that was pre-tagged upon its entry into the object list was sufficient to drive task-domain model. In order to combine both reference and domain information into a predictive model, a Bayesian Belief Network (BN) was used.

3.4 The Vesselworld Belief Network

To perform intent inference, we use a BN that generates a likelihood estimate for every possible agent-waste-operator tuple, where wastes are taken directly from the Object List, and operators come from a set of high-level goals we have defined. We are only concerned with relative likelihood estimates, and not absolute probabilities of each goal, and we make the assumption that goals can be evaluated independently of each other.

Figure 3 portrays a slight simplification (two interior nodes and nodes for the tug operator have been omitted to simplify the diagram) of the Belief Network that is currently being used in Vesselworld. Data is accumulated from the Object List, chat transcript, and plans, and posted to the unshaded nodes in the network. The interior nodes (shaded) are incorporated to reduce the total number of conditional probabilities entries required by the model. Nodes are classified into three broad categories to serve as general guidelines for building BNs in other domains that employ Coordinating Representations.

- Coordination Information: These nodes represent variables that are specific to the type of Coordinating Representation used.
- State Information: Information regarding the current state that determines the possible goals in the domain.
- Domain heuristics: These are heuristics that are not explicitly captured in a domain model, yet are powerful predictors of intent.

Within Vesselworld, these categories are instantiated as follows.

The ``Coordination Information'' nodes reflect a judgment as to how much a reference to a waste (taken directly from the Object List) that has appeared recently in chat influences the likelihood it will be lifted next. As co-referencing activities are pervasive in collaboration [Clark and Wilkes-Gibbs, 1990], we expect that other collaborative domains would benefit from CRs that provide similar referential information.

The "State nodes" reflect information about the state, such as whether the type of equipment is is appropriate, the size of the waste (which determines how many actors must be involved with the waste), and whether the agent is holding something. Some of this information (equipment and size) is derived from the directly from the Object List.

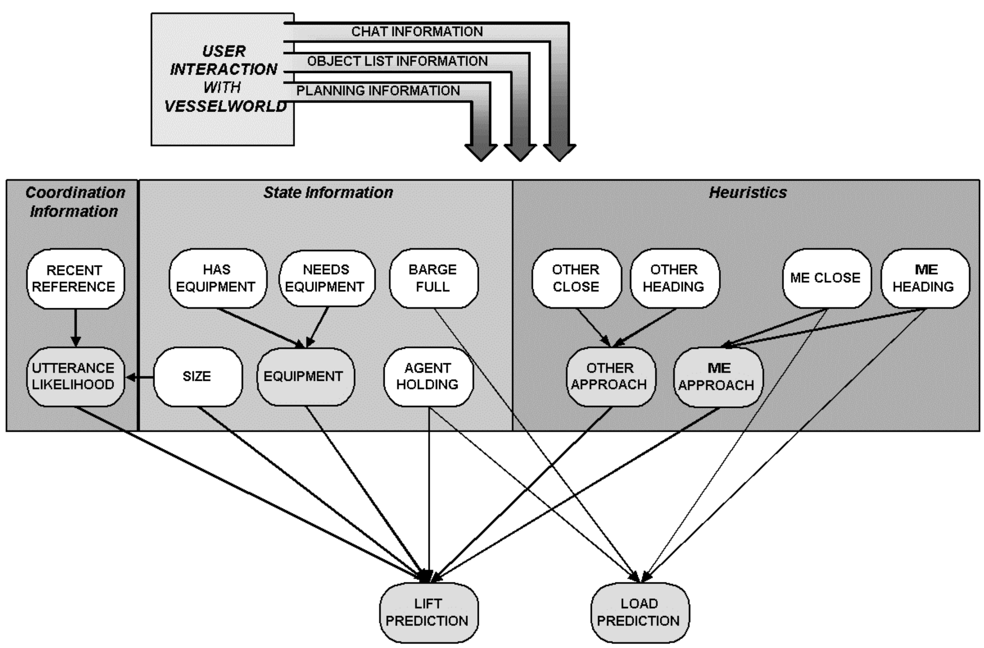

Figure 3 Simplified version of the Vesselworld Belief Network

The domain specific heuristics are heading and proximity, which may be derived from the positional information of objects in the Object List, and planning information from the users. The output of the network is a likelihood that a goal will be next. The "Agent State" node functions to switch between relevant portions of a network, so that either a "LOAD" or "LIFT" goal will be returned depending on whether the agent is currently holding a waste. The output is passed to the CoWare component described in the following section.

4 Coware

The Coware component was designed with two primary goals in mind:

1. It should not be intrusive or burdensome; users should use it because they want to.
2. It should be directly useful to the *individual*.
3. It should provide collaborators with awareness of other's future plans and goals.

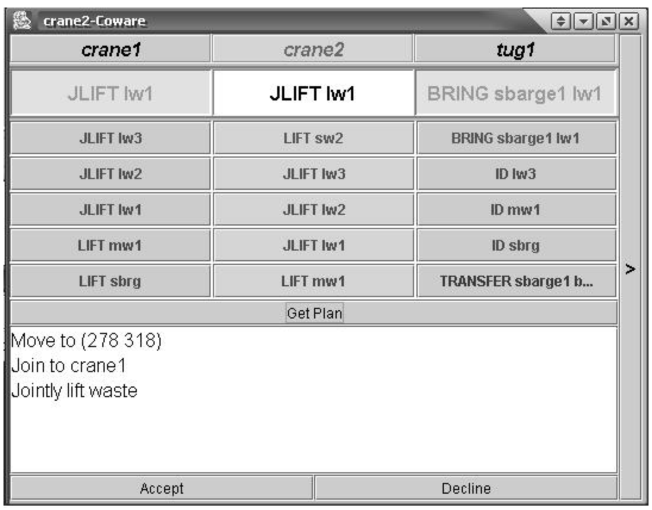

Figure 4 The Coware component

Thus, Coware is implemented as a passive, shared component. It is made useful by offering users automatically generated plans in exchange for a selection from a list of candidate intentions provided by the intent inferencing system. In selecting individual intentions, users communicate their current intentions to each other through the Coware interface.

Figure 4 is a screenshot of the Coware component. The system makes predictions about each of the participants' current plan. Each participant has the option to confirm one of five possible goals that the system has predicted they are working on. A user can only select a goal from their column. When a goal is confirmed, it is copied into the top row, which displays all users currently confirmed goals. This is the primary source of awareness information in the Coware component.

After a goal is selected, the user is provided an option to have the system automatically generate a plan for that goal. In cases where the goal involves multiple actors, the other actors are invited to join the plan. If all invited actors do not accept the invitation, a plan is not generated.

In summary, Coware uses the intent inferencing engine to significantly reduce the space of possible plans to a manageable set, and users will use the system because it simplifies plan generation. As a by-product of user interaction with Coware, intent information is shared with all users.

In the following section, we present the results of our user studies with the Coware component.

5 Empirical Studies

To evaluate they the effectiveness of Coware, we performed a 40 hour study with four groups of three people. The players were members of the Brandeis University community (including non-students). Each group 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. Subjects were asked to fill out exit surveys where they could give feedback about their experience with the system and the coordination issues arising in their group.

The participants were divided into two populations of two groups each. One was tested for 20 hours on the system without the Coware component. The other was tested on the system including the Coware component.

Our analysis seeks to answer the following questions:

1. Was Coware used?
2. Did it improve performance?
3. Did it reduce effort?
4. Did Coware improve intent awareness?

In the following, data is presented for the last 5 hours of play time for each group, by which time performance with the system had stabilized.

We show that Coware was used, that it improved performance by reducing errors, and reduced user effort during plan execution. In so doing, we demonstrate support for our claim that Coordinating Representations may be used to collect data that can be harnessed to drive an effective intent inference system.

5.1 Was Coware used?

All groups used Coware to generate plans within the system. Table 1 shows usage statistics for the Coware component, both as rate of activity and proportion of total planning activity. Our evidence show that system predictions were confirmed, and furthermore, plans were frequently requested, accepted and executed.

Our data shows that a confirmation was made by a user roughly every minute and a half during a game (games last anywhere from 45 min. to two hours). Confirmation

	Confirmation Frequency	Requests Per Conf	%Requests Accepted	%Accepted Completed	Coware Steps	Coware Crane Goals
Avg.	0:01:32	1.31	71.02%	82.88%	24.60%	43.34%
Std. Dev	0:00:46	0.30	18.88%	6.74%	7.59%	14.69%

Table 1 Usage statistics for Coware

frequency takes into account total clock time of each session to provide the average elapsed time between confirmations (when a user clicked on a prediction).

Once confirmations are made, a plan can be requested for that confirmation any number of times. The second column shows that there were at least as many requests as confirmations (though not necessarily a plan request for each confirmation), and that multiple requests were occasionally made for a confirmation.

After a plan is requested, it may only be used once (it is cleared from the users window after accepted). The third column shows that roughly 71% of the plans requested were accepted, and the final column shows that 83% of the plans accepted were actually executed to completion; that is, every step in the generated plan was submitted to the server.

In the second part of the table, we provide data that demonstrates a significant proportion of planning work was offloaded to Coware. The first column (Coware Steps) shows that nearly a quarter of all plan steps submitted to the server (where an average game contains about 380 total submitted individual steps) came from the Coware component.

The second column shows that 43% of the *goals* accomplished by the cranes (where a crane's goal is one of "Lift", "Load", "Joint Load", or "Joint Lift") that could have been produced by the Coware component, were. This number could not be calculated for the tug operator, as tug's plans do not have readily distinguished goals (e.g. a plan for the tug may consist of any number of moves between two arbitrary locations at any time).

5.2 Did Coware improve performance?

To determine what impact Coware had on performance, we compared our two populations for changes in interface activity, communication rate, and error rate. Our most significant result is that the error rate decreased for the population with Coware, though we can draw no conclusions from the other two measures. These results are shown in Figure 5 (normalized to the Base populations average for the purposes of graphing), and error bars reflect the standard error. We describe each of the values in the graph from left to right.

The number of mouse clicks per minute provides a rough idea of how much interface effort was required to use the system. Although we expected that the Coware component would reduce the amount of interface work, this was not the case. We suspect that the interface costs incurred when using the Coware component offset the reduction in clicks to generate plans manually.

Communication per minute increased slightly in the groups that used Coware. Because our population size is so small, further analysis of the dialog will be required to determine why this increase occurred, and what it indicates. It could mean that the Coware groups were simply more talkative than the others; it could also prove that the use of Coware resulted in more deliberation during planning.

Coware did help reduce the number of errors/min. We broke the number of errors down by type to identify the types of errors Coware helped avoid. We found that the biggest reduction was in the joint error rate. This result is shown in Figure 6.

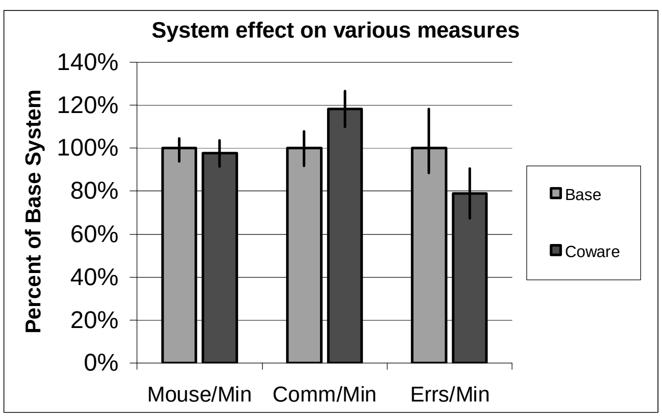

Figure 5 Coware effects on various metrics

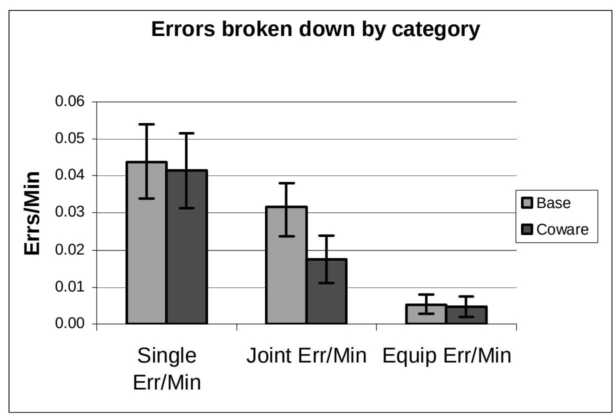

Figure 6 Coware errors broken down by category

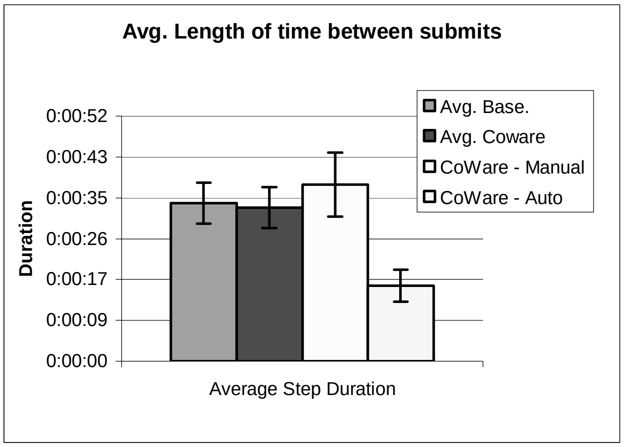

Figure 7 Plan step submission frequency

The three types shown in the chart are: joint errors, which are errors that occurred during some plan step involving two actors; equipment errors, which occur when an actor did not deploy the equipment needed to lift a waste; and single errors, which are all other errors. The number of joint errors was reduced in half for the Coware group, though the difference was not quite statistically significant.

This result is expected, and confirms other data collected with the Vesselworld system. In particular, joint errors account for a significant portion of errors made in Vesselworld [Alterman, et. al. 2001a], because they require close coordination, and thus the heightened attentions of the participants. Coware produces coordinated plans in advance, and users may simply submit each step and be assured that actions will be coordinated. This is further confirmed by the data discussed in the next section.

5.3 Did Coware reduce cognitive effort?

Our data (Figure 7) demonstrates that the plan step submissions occurred more rapidly for plans that were generated by Coware. This is indirect evidence that the cognitive effort during planning was reduced during execution of plans derived from Coware.

To collect this data, we examined the average time between plan steps for each group, as well as during manual and Coware phases of activity within the Coware group. Figure 7 charts the average inter-step time and standard errors for four selections of steps. Though there is insignificant overall difference between the two populations and manual phases of activity, step duration during execution of Coware plans was reduced by about half to about 16 seconds. This result corroborates the error data above, which points to the utility of automatically generated coordinated sequence of actions.

6 Discussion

The results presented in the previous section validate the non-autonomous approach to building adaptive groupware. Recall the questions asked at the beginning of section 5,

1. Was Coware used?
2. Did it improve performance?
3. Did it reduce effort?
4. Did Coware improve intent awareness?

Our data provides positive answers for the first three questions. Coware was used heavily, it improved performance by reducing joint errors, and it reduces cognitive load during plan submission.

However, we have not addressed the fourth question. From our exit interviews, we have surmised that although Coware does provide intent information, users did not use it to stay aware of each other's intentions.

In response to one of the questions asked in the exit interview, "Do you have any ideas about improvements to the system that would help you stay coordinated?" one user responded,

"The only thing that I can thing of is maybe combining some of the different things together? …Sometimes the most difficult part of the problem was knowing where people were or where they were going. I guess you could use the Coware stuff to see that, but for some reason we didn't. Maybe if the planning window had something where it could show the final destination ? Although this is kind of ridiculous because that's exactly what Coware did."

Another user responded to the same question,

"…Also, the interface is wayyy too busy, and it would help if all the crap I needed to look at were fixed in place, in a way where key info would be visible"

Our preliminary conclusion from this data is that the awareness information provided by the Coware component was not used because it was not *situated* in the interface. Rather, information was provided in a separate component, and required interpretation to be resolved to objects in the worldview.

This result is echoed in the work of [Gutwin, et. al. 1996; Gutwin and Greenberg 2001] who emphasize the importance of situated information in the user interface. We are currently exploring the possibility of trying to incorporate information from the Coware component directly in the shared worldview.

Acknowledgments

This research was supported by the Office of Navel Research under grants No. N00014-96-1-0440 and N66001-00-1-8965.

References

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

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

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

[Clark, H. and Wilkes-Gibbs, D., 1990] Referring as a collaborative process. *Cognition*, 22:1-39.

[Cohen, R., 1987]. Analyzing the Structure of Argumentative Discourse. *Computational Linguistics*, 13:11-24.

[Dourish, P. and Belloti, V., 1992] Awareness and coordination in shared workspaces. In *Proceedings ACM Conference on Computer Supported Cooperative Work*.

[Feinman, A., and Alterman, R., 2003] Discourse Analysis Techniques for Modeling Group Interaction. Ninth International Conference on User Modeling (in press)

[Flores, F., et. al., 1988] "ComputerSystems and the Design of Organization Interaction," ACM Transactionson Office Information Systems, 6:2(153-172)

[Gutwin, C., et. al. 1996]. A Usability Study of Awareness Widgets in a Shared Workspace Groupware System. Proceedings of ACM CSCW'96 Conference on Supported Cooperative Work, Boston, Mass., November 16-20, 258-267, ACM Press.

[Gutwin, C., and Greenberg, S., 2001] A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work (CSCW), The Journal of Collaborative Computing*, Kluwer Academic Press

[Jenson, F. 1996]. An Introduction to Bayesian Networks. Springer.

[Pearl, J, 1988]. *Probabilistic Reasoning in Intelligent Systems.* San Francisco, Calif.: Morgan Kaufmann.