# Intent Recognition in Collaborative Interfaces

**Josh Introne, Rick Alterman**

Department of Computer Science
Brandeis University
jintrone@cs.brandeis.edu
alterman@cs.brandeis.edu

**Abstract**

Collaboration consists of the joint activities of several participants. In order for joint activities to advance, participants must effectively coordinate their actions, and effective coordination appeals to common ground. [4,5] In this paper, we discuss an intent recognition technique that utilizes information that is part of common ground – namely, shared references to domain objects and their associated information. Our technique is designed to support an adaptive information management system for the Vesselworld collaborative system. The two key insights which motivate this work are: 1) *Coordinating representations* [2] may be introduced to groupware systems which structure shared information and facilitate coordination, and 2) AI techniques can be employed to take advantage of such structured information to support effective interface adaptations. Here, we introduce a technique for intent recognition that uses information from a coordinating representation designed to support common ground.

## Introduction

As computers are integrated into more complex environments, the need for systems that are explicitly designed to support multiple human operators becomes apparent. We have developed an incremental approach [1] to designing such systems. This 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 may be introduced to support the exchange of information for difficult coordination tasks.
- Data collected from the user of these components may be harnessed to support interface adaptations.

Our methodology is based on theoretical notions of coordination [4,5] and employs discourse analysis [9] to identify coordination problems in an existing collaborative system. Coordinating Representations [2] that are suggested through the execution of this methodology are designed and integrated with the existing system. Empirical results have shown CRs to be effective at improving coordination in our testbed domain, Vesselworld [1,3]. Use of CRs provides data that can be harnessed by the intent recognition technique we describe here.

In the following sections we will give some background on Vesselworld and the CR that provides the bulk of the data which drives intent recognition. We will then introduce our technique, and describe how information from the CR and task domain knowledge can be leveraged to produce predictions about user's intended goals. We will provide some preliminary results and observations, discuss future work, and summarize this work.

## Vesselworld and Coordinating Representations

The Vesselworld system is a collaborative system we have built for the purpose of developing a design methodology and adaptation technique for collaborative systems. In Vesselworld, three participants collaborate on remote workstations to remove barrels of toxic waste from a

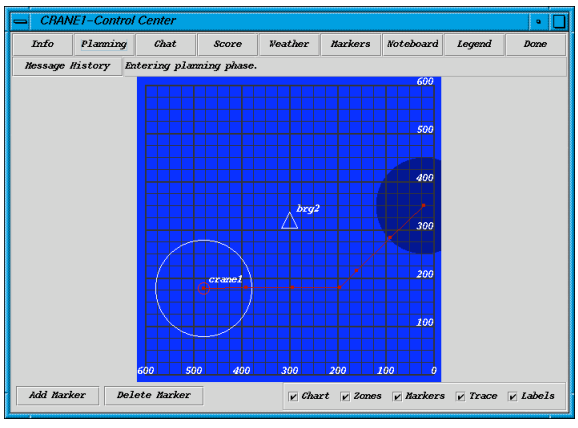


**Figure 1** The Vesselworld Interface

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. Each ship has geographically limited view of the harbor and some unique capabilities. To safely remove a barrel of waste from the harbor requires varying degrees of coordination between two or more actors. Play is turn based, in 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. Figure 1 is a snapshot of the main window (the Control Center) in Vesselworld for one of the ship captains.

Roughly eighty hours of data have been collected with the Vesselworld system. In analyzing data obtained from experiments with an initial version of the system, we identified several classes of coordination problems. On the basis of that analysis, we developed three Coordinating Representations that were designed explicitly to alleviate these errors.

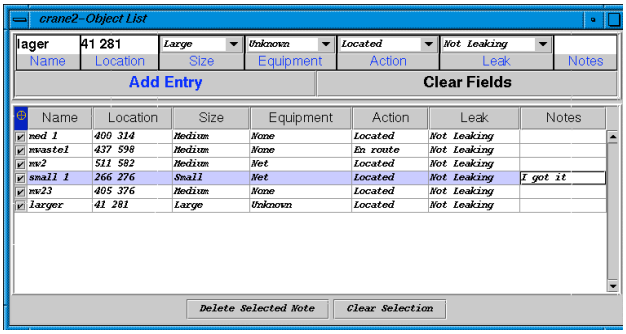


**Figure 2** The Object List

One of these CRs, which has been shown to be both effective and heavily used [1] is the Object List, which is shown in Figure 2. The Object List 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 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. 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. 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. Our intent recognition engine will be subsequently used to support adaptive information management in the interface.

## Intent Inference

Our goal is to provide accurate predictions about the toxic waste the users intend to address next in plan execution. One approach is to extract information from ongoing chat transcripts by means of an automatic discourse analysis. One might use heuristics and semantic analyses to identify reference tokens and build a model of how references are typically used in discourse for planning. However, there are many difficulties with such an approach. Reference tokens in chat are subject to typos, and referents are often referred to using several different means (e.g. “MW3”, “that medium waste”, “the one right next to you” might all be used at different times to refer to the same object). Very often, these referents require a rich and domain specific understanding of contextual information in order to be interpreted. Furthermore, not all executed plans are discussed, and consequently other information must be used if all targets are to be predicted. The amount of effort involved in developing the domain knowledge necessary to support such an autonomous approach may be prohibitive, and the bulk of this work would not be readily transferable to other domains.

Our approach uses coordination work users are already doing to avoid complex semantic analysis. The Object List CR provides two readily available pieces of information that we can use in our intent recognition system. 1) Information about the domain as it is perceived by the users is provided. 2) A list of active user references to objects in the domain is provided, and these references are often used in chat to discuss plans for objects. We will show that the structured information generated through normal usage of the CRs supports more general techniques for intent recognition, and that these techniques are more rapidly implemented and likely outperform other methods which do not use this information.

For our analysis, plan execution that addresses a toxic waste site corresponds to the initial lift of that waste. Only the first lift for each waste is considered here without loss of generality. We will refer to these initial lift plans as *target plans* and the corresponding wastes as *targets* for the remainder of our discussion.

Preliminary studies with chat references alone found that the existence of references in chat to targets were good predictors of target plans in the five minutes preceding plan execution, especially where the target waste involved significant coordination. However, false prediction rates were very high, as we did no semantic analysis of chat logs

to determine the meaning associated with an utterance. To enhance the predictions made using reference information, information from the Object List was used to populate a probabilistic task domain model that is captured in a Bayesian Belief Network (BN).

## The Vesselworld Belief Network

Belief Networks (BNs) [6,8] are probabilistically sound causal reasoning systems, typically used to determine the likelihood of some set of outcomes or states given uncertain evidence about the world. BNs consist of a DAG, in which nodes of the network denote the variables and the links denote causal relationships between the variables. The topology encodes the *qualitative* knowledge about the domain. Conditional probability tables (CPTs) at each node quantify the quantitative details (strengths) of the causal relationships. BN engines with well-developed APIs are readily available. Several adaptive methods are available to assist in generating and tuning CPTs for BNs. We use an implementation of the Expectation-Maximization (EM) algorithm [7] which allows BNs to learn the conditional probability distributions in the CPTs from a set of observed cases. A similar method can also be used to adapt probability distributions on the fly.

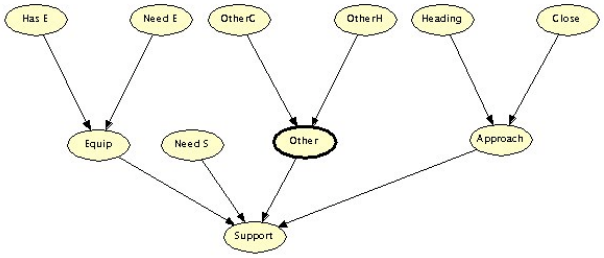


**Figure 3** Hand built network for Vesselworld

Our initial BN, shown in Figure 3, was hand built to reflect our knowledge about the domain. It contains domain information provided to the users (such as the type of equipment needed and coordination requirements for the wastes) and some heuristic information about heading and proximity. These heuristic measures indicate whether an agent's movement vector will intersect a given waste and how close it is relative to other wastes, as well as whether or not the waste is in the effective range for the agent. The output of the belief network is read from the "Support" node, which provides the likelihood that an object will be the next target for the users for each object known about.

## Learning CPTs

After some initial testing (results are included below), we decided to use the EM algorithm to tune the CPTs to our data. To generate a CPT for any node, the EM algorithm requires that discrete findings be posted to that node and all parent nodes. As the interior nodes in our graph do not correspond to observations in the world, but serve a conceptual role and to reduce the size of the CPTs, they were removed from the BN shown in Figure 3 to produce the BN shown in Figure 4. The resulting CPT for the "Support" node has $2^{10}$ conditional probability entries, and is too large to be easily filled by hand.

Data to be used by the EM algorithm for offline learning is provided as a set of cases that represent individual observations for each waste object, user, and state update. This information is provided by logged data from the Object List and plan executions for several training

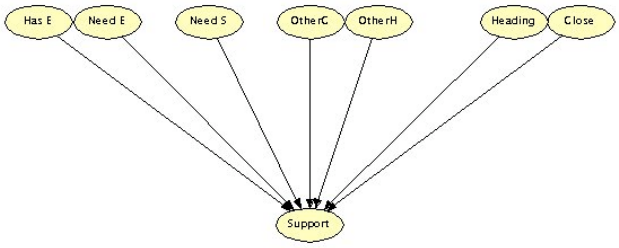


**Figure 4** BN without internal nodes

sessions. Note that the state of the "Support" node is not generally observable; it is a prediction about the likelihood a given waste will be addressed next, and is only known when a target plan is executed. To provide a value for the "Support" node in the case file, we enter a "true" if a target waste that is being considered is followed within five minutes by the associated target plan in the data log, and there is no other intervening target plan. For example, if waste MW3 is lifted at time $t_{MW3}$, each case occuring between $t_{MW3}$-5min. (or $t_X$, the time of the prior target plan execution if $t_X > t_{MW3}$-5min. for waste X) and $t_{MW3}$ receives a "true" observation for MW3, and a "false" observation for all other possible target wastes. The time window is chosen to maximize the predictive power of our case set by not entering predictions that cannot be made from a given state of the world.

## Prediction Using the BN

Predictions are produced each time new information is available to update the evidence that is posted to the belief network. For domain information alone (not incorporating references in chat) this occurs whenever the object list is modified and each time plans are submitted to the server.

When new information is available, evidence is posted to the network for each target and each participant in succession, and a probability is obtained which indicates the likelihood that target will be addressed next. The target with the highest likelihood greater the minimum likelihood specified is chosen as the prediction at that time. If more

than one target has the same highest likelihood, one is chosen deterministically at random.

## Early Results

The table below provides some early results comparing mean performance from the initial hand generated BN and the BN which derives its CPTs from the EM learning algorithm. The results were obtained from a set of data from one group of test subjects playing Vesselworld, spanning seven different sessions for a total of 7.7 hours of usage time and 138 different wastes. Three of the largest training sessions for this group were used to generate a case file for training the adapted BN that is 5,539 cases long.

| Min Confidence | Correct Rate | False Rate | Missed Rate |
|---|---|---|---|
| | *Case Adapted BN* | | |
| **0.65** | 0.66 | 0.42 | 0.34 |
| **0.70** | 0.66 | 0.37 | 0.34 |
| **0.75** | 0.55 | 0.37 | 0.45 |
| **0.8** | 0.24 | 0.39 | 0.76 |
| | *Hand Built BN* | | |
| **0.65** | 0.82 | 0.67 | 0.18 |
| **0.70** | 0.82 | 0.51 | 0.18 |
| **0.75** | 0.81 | 0.40 | 0.19 |
| **0.80** | 0.81 | 0.37 | 0.19 |

The rates listed in the table headings are computed as follows:

**Correct Rate**: Ratio of targets correctly predicted to the total number of targets.

**False Rate**: Ratio of false predictions to the total number of predictions.

**Missed Rate**: Ratio of targets not successfully predicted at all to the total number of targets.

For the these calculations, a waste is **correctly predicted** if there is a consecutive series of predictions (recall that a prediction is always the most likely target chosen when the BN is updated) for a target leading up the time that the associated target plan is executed. A target is **falsely predicted** if a series of one or more consecutive predictions do not immediately precede an execution involving the predicted target. A target is **not predicted** if no correct prediction for that target was made.

The confidence column in the table lists specified minimum confidence values that yielded the best results; for both networks, higher confidence values resulted in significantly less and frequently no correct predictions, and lower values saw no improvement in the Correct Rate.

### Discussion

It is apparent from the results that the hand-built belief net attains much better performance than the belief network that has been filled out via the EM algorithm. This may be due to the size of the CPT that must be derived by the training algorithm, which contains $2^{10}$ conditional probabilities; roughly a fifth of the total number of cases in the case file.

Our case generation method is also quite naïve, and more effective means might be found. In particular, observation cases in the case file that are closer to an observed plan execution may be more heavily weighted during training as they provide better indicators of predictive situations. It is also be possible a taller tree with a lower branching factor (and thus smaller CPTs) may be trained in pieces, or by using another learning algorithm.

We are encouraged by the early results for the hand-built network but feel that the false positives need to be further reduced to support useful adaptations. Upon close examination of the results it appears that many false predictions occur sporadically, whereas correct predictions generally occur in longer sequences. Thus, we might incorporate the persistence of a likelihood estimate over time into the final prediction volunteered by the system as a whole.

Our next immediate step is to include information about references from chat into the prediction engine. We discuss this and several envisioned interface adaptations in next section.

## Future Work

Our next step is to incorporate reference information from chat transcripts. In initial studies, we found that names for wastes that are entered into the object list are used in chat, and frequently where those wastes involve multiple actors. Thus we propose to include this information in the belief network. One possible design for this belief net is shown in Figure 5.

The “Ref” node will receive “true” for a given target if a reference to that target is observed in chat within some time window preceding the lift. The addition of the direct link between the “Need S” node (which indicates the type of coordination a waste needs) and the “Prediction” node is introduced to condition the significance of chat references on the coordination requirements for the waste under consideration.

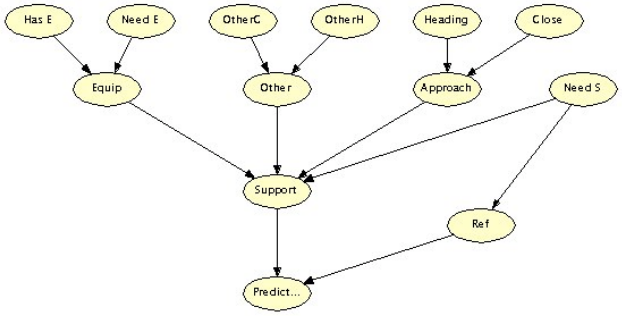


**Figure 5** BN incorporating reference information

Once we have generated a successful intent recognition method, we will introduce adaptations to the interface. Depending on the confidence in our predictions, several adaptations are possible. If our predictions are good enough (low false positive rate and high correct rate) we might simply highlight and/or make information relevant to the currently predicted target more obvious in the interface. We might also provide information about the currently predicted targets of other users, in an effort support common ground and enhance situation awareness.

If we cannot reliably predict the exact domain target for each user, but can generate a list of possibilities that nearly always includes a correct prediction we can introduce a selection mechanism that allows users to choose the current intended waste target from a small list. In this case, there must be high payoff to the users to balance the intrusion created by this mechanism, such as offloading some of the planning work for the user.

## Summary

We have presented a method for performing intent recognition in collaborative systems. Our approach depends upon the users' need to exchange information to stay coordinated, and the data collected by coordinating representations that are designed to support coordination work that is hard in specific collaborative domains.

Our intent recognition approach employs Belief Networks to represent domain knowledge and heuristics, and uses data available from Object List Coordinating Representation as a source of evidence. We are currently incorporating object referencing that occurs in chat into our framework.

Our preliminary results are encouraging, and several avenues of approach are open to improving performance. Once we have attained good performance levels, we will introduce interface adaptations that use the intent recognition system, and perform experiments to judge the effectiveness of those adaptations.

## References

1. Alterman, R., Feinman, A., Landsman, S., and Introne, J. (2001) Coordination of Talk: Coordination of Action. Technical Report CS-01-217, Department of Computer Science, Brandeis University, 2001
2. Alterman, R., Feinman, A., Introne, J., and Landsman, S. (2001) Coordinating Representations in Computer-Mediated Joint Activities. Proceedings of 23rd Annual Conference of the Cognitive Science Society, 2001.
3. Alterman, R., Landsman, S., Feinman, A., and Introne, J. Groupware for Planning. Technical Report CS-98-200, Computer Science Department, Brandeis University, 1998.
4. Clark, H. and Wilkes-Gibbs, D. (1990). Referring as a collaborative process. *Cognition*, 22:1-39.
5. Clark, H. (1996) *Using Language.* Cambridge University Press.
6. Jenson, F. (1996). An Introduction to Bayesian Networks. Springer
7. Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis* (19): 191-201.
8. Pearl, J (1988). *Probabilistic Reasoning in Intelligent Systems.* San Francisco, Calif.: Morgan Kaufmann.
9. Sacks, H., Schegloff, E., and Jefferson, G.. (1974) A simplest systematics for the organization of turn taking in conversation. *Language*, 50:696-735.