Beyond Dialogue: The Six W's of Multi-party Interaction

Penelope Sibun

The Institute for the Learning Sciences
Northwestern University
1890 Maple Avenue
Evanston, IL 60201
sibun@ils.nwu.edu

Abstract

While progress has been made in modeling mixed-initiative discourse, many interactive natural language systems still take much for granted. For example, the typical system doesn't worry about who or where its interlocutor is, or whether there is more than one of them. If a natural language system is to cope in any sort of unrestricted domain, such questions become paramount. In this paper, I describe ongoing research in which a "bot," an interactive character run by a computer program, is deployed in a MOO, a multi-user, text-based, interactive domain. I will show that before the intricacies of mixed initiative and discourse structure can be addressed, the what, who, where, when, why, and how of such discourse must be faced.

Introduction

It is often taken for granted that conversation is situated and dependent upon many factors above and beyond the text involved. As Biber (1980) notes, in the extreme, conversational participants must be collocated at the same time and place. Topics of conversation tend to be more concrete than topics of writing, and conversational participants tend to expend proportionately more effort on managing their interaction than on conveying information.

Since computers have much less access to the concrete aspects of a conversational situation than do humans, any computational system attempting to engage in conversation starts out with a large handicap. Many projects in computational linguistics are severely limited: for example, most natural language generation systems are very good at monologues, but cannot interact with any other agents. Today, there is increasing emphasis on building systems that can communicate directly with humans or with each other using natural language. Such a system might be a natural language front end to a database (e.g., Allen, et al. 1996), or two programs might converse with one another to test models of dialogue or discourse (e.g., Walker 1993; Novick and Ward 1993). Such systems accommodate an interlocutor by engaging in turn-taking and answering questions appropriately.

However, as Emanuel Schegloff points out in the proceedings of "Burning Issues in Discourse" (1993), multi-party interaction is importantly different when more than two parties are involved. Suddenly, there is no "the" interlocutor, but several potential interlocutors. And, if there is more than one person to talk to, how does one decide which one to talk to or whether to talk to anyone at all? Things become even more complicated when potential interlocutors may not be collocated in space or in time.

While people deal with all these issues every day, few, if any, natural language systems have these capabilities.[1] If we are to build systems that will behave robustly in the everyday human world, then we will need to give them the ability to cope with such issues.

My current research involves building a "bot" that functions in a MOO. A MOO is a type of MUD, or Multiple-User "Dungeon." Conceptually, a MOO is a place composed of rooms and that contains objects, in which characters act and speak. In a MOO, anything a character can see is described in text and anything that a character does produces a textual description of that action. Because it is text-based, a MOO is a very flexible environment, and participants can customize the appearance of places, the apparent effects of objects, and the expression of actions. In technical terms, a MOO is a large, extendible, object-oriented database to which many users can connect by telnet. Typically, a user creates a character and through that character moves about the MOO, creates and uses objects, builds places, and, perhaps most importantly, converses with other characters.

A bot is a character that, instead of being animated by a person, is run by a program connected remotely by telnet. A bot has available to it the same commands for moving, creating, and conversing that are available to any character; the trick is for the program to use these commands in a way a human would. If the bot were completely successful it would pass the Turing test; it is doing well if it manages more than a couple of conversational turns before being caught out as a fraud. Generally, the best a bot can hope for is to be tolerable enough that it is not shunned by all.

While MOO interaction is different in some ways from ordinary conversation (Curtis, 1992; Bruckman, 1994), for example, by being typed and by having inherent time delays, people still have strongly ingrained expectations of their interlocutors. I am building a system to meet these expectations.

At the Institute for the Learning Sciences, we administer a MOO called Muspell. (Muspell is based on LambdaMOO and the Pueblo Kernel Software licensed by Xerox Corporation to Northwestern University. It runs under UNIX on an IBM RS6000.) About 50 people have

[1] There are multiple agents in Novick and Ward's model: while two converse, others may overhear. However, it appears that these roles are predesignated for the agents.

```
The Kitchen
The floor is covered with Black and White linoleum, in a plaid pattern that
almost, but not quite, reminds you of seventies modernism. There is  a counter
with an old Mr. Coffee on it, a refrigerator, and of course, the kitchen sink.
Above the sink is a window, revealing a pastoral  scene of boxy ponies and
rolling hills.
You see a Refrigerator, a Kitchen Sink,  and a Counter here.
Obvious exits: Foyer...<west>
You have arrived.
box arrives from the Foyer.
You say, "hi box"
box says, "Hi pen"
You say, "have you seen my cat?"
box says, "No."
>drop cat
The Cat jumps from your arms.
box picks up the Cat.
>look at box
cubic, transparent, with small clouds, and raindrops
Sleepy Cat curls up in box's arms.
box says, "Wow!  Nice cat."
The Cat jumps from box's arms.
>look at box
cubic, transparent, with small clouds, and raindrops
box is awake and looks alert.
box says, "bye"
box goes west
```

Figure 1: Sample interaction on a MOO.

characters on Muspell with more being added weekly. While most of the players are undergraduates at Northwestern University, there are players of all ages and from as far away as Australia. Typically, between two and seven players are active at any one time. Figure 1 shows a sample interaction from the point of view of one of my characters.

While we usually have two or three bots running at any one time, the bot with the most experience and the most innovation is named Robin, and I will refer to Robin throughout the rest of this paper. (The bot implementation is written in Allegro Common Lisp and runs on Pentium Pros under Windows 95.) Robin is robust, running for days at a time; most commonly, its execution is interrupted only by difficulties of its or Muspell's machine or the telnet connection between them. Robin is part of a CAETI-funded project for teaching science to middle school students; the bot is one of several interfaces to a system that produces self-explanatory simulations of such physics phenomena as evaporation.

Bots have been running on MOOs and MUDs for many years. The most well-known of these bots is Julia, written by Michael Mauldin (Foner, 1993). Julia was built over the course of several years and became a large program tailored to MUD-based activities. She had many clever modules, such as an algorithm for guessing a character's sex based on the probable gender of its name. She also engaged in activities such as giving directions in the MUD.

To the best of my knowledge, Julia was not built with a principled discourse or conversational model in mind. Robin is an experiment in building such a principled and robust discourse model that will function in the relatively unrestricted world of Muspell. While ultimately intended to tutor such difficult subjects as evaporation, Robin is currently working on engaging in and disengaging from conversations gracefully and in continuing them appropriately. The bot's natural language understanding is accomplished by a simple pattern-matching mechanism. Most of its natural language generation is via canned text; for each turn in a conversation, it chooses randomly among a set of appropriate pre-generated choices. In future versions of the bot, the natural language generation, as well as the discourse management, will be handled by Salix (Sibun, 1991, 1992). Robin can do various things when told to do so. For example, any character can tell Robin to do such things as go either in a specified direction or to a particular place. Robin can also be told to "shut up," which is useful when it becomes a nuisance. Certain characters are authorized to command Robin; these characters may tell Robin to go to sleep, to wake up, to turn itself off, or to ignore or obey another character.

Most of what Robin does is engage in conversation. In order to do this, it needs to figure out what, who, where, when, why, and how: what sort of conversation it's having; whom it's talking to; where that character is; when to talk to her; why it's talking to her; and how it's going to have it. I believe that once the bot has strategies to successfully

```
[Galadriel has been active, but does not respond to Robin.]
Robin arrives from Caer Sidi.
Robin says, "Hi!!!!!"
Robin says, "Hmm.  Maybe I'm talking to a brick wall."
The Piper's music fills the air, the time is: 6:30 P.M. (CDT)
Robin sighs.
Robin whistles tunelessly and jiggles from foot to foot.
```

Figure 2: Galadriel has been active, but does not respond to Robin.

negotiate these questions, it will be able to interact with other characters in ways that they can anticipate and respond to. These "Six W's" are essential questions for any interactive language system to address. In the next section, I will describe how Robin answers or fails to answer these questions, drawing on actual transcripts of the bot's interactions.

The Six W's

What

Robin represents *what* sort of interaction it is having by a small number of conversational modes (currently, there are three). These modes are: respond to an interlocutor; initiate a conversation with an interlocutor; and prompt an interlocutor who is apparently distracted.

In *respond* mode, Robin says something every time its interlocutor does. These responses tend not to be very interesting. Often they are simply prompts. One response is, "I'm not much of a conversationalist, but I like to go places." This response is intended to suggest to an interlocutor that Robin may be told to go somewhere. (Unfortunately, few interlocutors rise to the bait. This may be because Robin's limited language skills cause them to doubt its other capabilities.) Robin may also respond with the important message, "You can always tell me to shut up, <interlocutor's name>."

Robin may *initiate* a conversation if it has no interlocutor or if its current interlocutor seems to be unresponsive. Robin finds someone to talk to by scanning down the list of active characters until it reaches the name of the most recently active character on its authorized list. Authorized characters are those whose players have agreed to be approached by the bot. When an authorized character is found, Robin issues a MOO command to join that character wherever it may be. Sometimes the join command may fail, because the character is in a room that is inaccessible to the bot. In such a case, the bot notices the failure, and aborts the attempt. If the bot can join a character, it does so and greets the char by saying, "Hi!!!!!". The character so greeted becomes the bot's new interlocutor.

A character may also become the bot's new interlocutor by speaking to it. Generally, a character is considered to have spoken to the bot when it uses the bot's name. We are working on an extension by which the bot will consider itself spoken to if it and another character are alone in a room and the other character asks a question. Once the bot has acquired a new interlocutor, it moves to respond mode and responds each time its interlocutor speaks.

The bot will enter *prompt* mode after some period during which the interlocutor is not speaking. An interlocutor may stop speaking for a variety of reasons: the character may be engaged in other Muspell activity, such as room construction, or, more likely, the player is not currently attending to the MOO connection. Periodically, the bot issues a prompt to its interlocutor suggesting further conversation (see figure 2). Prompts include, "Are you still there, <interlocutor's name>?" This expression is frequently seen in MOOs, since players are aware that other players often desert their screens. Robin looks for somebody new to talk to every few minutes. If several tries yield no one, and the current interlocutor has remained unresponsive, the bot will issue a prompt to its current interlocutor. These prompts come about five times an hour. This length of time is purely a product of experimentation. I expect revisions may be necessary if Muspell becomes more active; at the moment, no one wants to talk to bots for more than a few minutes, so the possibility that it might trundle off and find someone new to talk to has not become a problem. However, I expect the situation will someday arrive and then the bot will need to have a mechanism for deciding between sticking with the old conversation and finding a new one.

Who

Although they may find it taxing, people are adept at figuring out *whom* they are talking to and coping with several interlocutors at once. People use many verbal and non-verbal signals to catch and retain the attention of others; similarly, people are very good at picking up such cues. A person can make clear that she is addressing one person, or she may include the entire group. In addition, someone can easily pick a single conversation out of a hubbub (Cherry, 1953), or switch back and forth between two or three separate conversations with different interlocutors. At any time, a new interlocutor may interrupt and catch the bot's attention. Such an interruption is not problematic if the old interlocutor is no longer interested in conversing with Robin. But if both interlocutors continue to participate, the bot, which only has room in its model for one interlocutor, will become confused and continue talking to the wrong person or at the wrong time.

```
The Boxpen
The woods gradually give way to rolling hills, covered with Kentucky
bluegrass. Small, boxy ponies wander aimlessly in a large wooden corral.
Obvious exits: Woods of Annwn...<north> and Entryway...<south>
You have arrived.
box arrives from the Entryway.
Robin arrives from the Woods of Annwn.
Robin says, "Hi!!!!!"
box says, "Hi robin"
Robin says, "I enjoy talking with you, Box."
box says, "That's nice"
Robin says, "What?"
box says, "I like talking with you Robin"
Robin says, "What?"
You say, "how you doing, box?"
Robin says, "What?"
box says, "Things are OK."
Robin says, "I'm not much of a conversationalist, but I like to go places"
You say, "we're not talking to you, robin"
Robin says, "I'm not much of a conversationalist, but I like to go places."
You say, "fine; go north"
Robin goes north.
```

Figure 3: Robin tries to converse with two interlocutors who are themselves trying to conduct a private conversation.

On a MOO, there are fewer signals, since everything available to a player is in text. While text can still convey enough information for a player to tell, for example, whether other characters are present in a room, this would not be a trivial task for a bot, because in the limit it would require full parsing capabilities in order to identify all characters' names.

Robin always responds when a character authorized for it speaks to it. The bot also responds to anyone who addresses it by name. Two characters may find themselves in a situation where they are vying to be the bot's interlocutor. For example, if two authorized characters speak to the bot in turn, the bot will respond to each of them and with each response assign the character to which it responded as its interlocutor. This may result in desired behavior. However, if the two characters decide they wish to speak to each other, the bot is unlikely to note the switch. Figure 3 is an example of two authorized characters: one talks to Robin and then the two try to talk to each other. The problem lies not in whether or not Robin switches its attention back and forth: in this case, *neither* character is addressing the bot, but there are no readily-available cues to that effect. It is possible that if the bot were capable of fully parsing the text output by the other characters, it would understand that it was no longer meant to be included in the conversation. However, in general, knowing when it has been excluded from a conversation is going to be difficult, if not impossible, for a bot.

Where

As I have described above, the bot knows that in order to initiate a conversation with a character, it must be *where* that character is. As initially designed the bot cannot keep track of whether it remained in the same place as its interlocutor. This oversight had the consequence that the bot would continue to talk to an interlocutor who had walked away, disconnected, or otherwise disappeared. There was also an instance in which Robin was ejected from the home of its interlocutor (that is, it was forcibly removed to its own home) and did not notice. These problems can be corrected by having the bot periodically check the "who" listing and ascertain that it and its interlocutor remain in the same room, and rejoin the interlocutor if the situation changes.

When

I have already alluded to the issues of *when* to enter a particular conversational mode. There are undoubtedly times when other actions will become appropriate. For example, if there has been no activity on the MOO for a while, Robin should perhaps give up both prompting its interlocutor and searching for new ones for some period of time, and take a nap.

There are other timing issues that are particular to interaction on a MOO. These mostly have to do with network delays. Human players learn to adjust to a particular latency period, and learn to expect a certain asynchrony between their actions and those of others. There are delays built into the bot's code to allow for most time lags. The time lag actually works in the bot's favor, since any character's responses can seem disjointed depending on their network connections.

How

There are several modalities available to characters in the MOO when they decide *how* to communicate. A character uses the "say" command to express text that will be seen by all other characters in the same room. A "page" command, on the other hand, will send a message to a specific character regardless of where they are. Most MUD conversation is conducted by using either "say" or "page." People encountering each other in a room use "say"; close friends will often conversation by paging each other regardless of location, as if they have virtual walkie-talkies. There are two other saying commands, a directed say, and a whisper, both of which are directed to an individual character in the same room as the speaker, but these are rarely used. There is also an "emote" command which allows a character to express something that is not intended to be said. For instance, a character can emote a sigh or emote that he scratches his head.

Robin uses the "page," "say" and, to a lesser extent, "emote" commands. The bot prefers using "say" and will only approach new interlocutors using that command. (It is generally considered rude to page someone you don't know in a MOO.) The bot will respond to both pages and says. It responds in kind; this is important because a page is usually considered a private communication and it would be inappropriate to respond to a "page" with a "say" even if the interlocutor were in the same room, because a "say" might be overheard by other characters. And of course, it may also be inappropriate because someone who is paging the bot my not be in the same location.

Why

We have worked our way through five of the six W's. The one that remains is *why*. If we were to ask why a bot such as Robin is engaging in conversation at all, we might answer that it is doing so because it has a goal, envisioned in our overall project, of tutoring someone in evaporation. But that question may be less interesting than questions at a finer level. Why would a bot choose to approach another character? Why would a bot respond when spoken to? Why would a bot decide to leave an interlocutor and seek out a new one? These may seem like silly questions to ask about a program, but if we turn these questions around and create bots who differ in their behaviors in seeking out interlocutors or responding to potential interlocutors we would create bots with different personalities. We could then say that a bot that never sought out other characters and only sometimes answered when addressed was reserved. We might also say that a bot that was constantly finding a new interlocutor was a social butterfly. Then the question of why would be answered by appealing to these personalities.

Conclusion

In this paper I have shown that an interactive agent in an unrestricted environment will need to manage aspects of its interaction that are beyond the scope of most computational interactive natural language systems. Such a system needs to cope with such issues as who its interlocutor is, where its interlocutor is, and what the state of the interaction is. I refer to these considerations as the six W's that must be addressed by any interactive agent.

Robin and the other bots on the Muspell are very much works in progress. Every time Robin engages in conversation with another character, we gain a wealth of information. Conversational behavior is a phenomenon that is difficult to describe and model abstractly, but it is one that is easy for humans to evaluate instantly and mercilessly. I expect to continue to refine Robin's behavior and by doing so, to create a bot that is worth talking to.

Acknowledgments

I thank Tamar Gutman for her Muspell wizardry and programming assistance. I thank Jon Handler, Julie Baher, and Lyn Walker for fruitful conversations. I thank Lauri Blouke for typing this manuscript. And I thank Ken Forbus for sponsoring this project. This research is sponsored by the Advanced Research Projects Agency of the US Department of Defense, under the Computer Education and Training Initiative.

References

Allen, J.F. Miller, B.W., Ringger, E.K., & Sikorski, T. (1996). "A Robust System for Natural Spoken Dialogue." In: *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, CA. pp. 62-70.

Biber, D. (1988). *Variation across Speech and Writing*. Cambridge: Cambridge University Press.

Bruckman, A. (1994). "Programming for Fun: MUDs as a Context for Collaborative Learning." National Educational Computing Conference, International Society for Technology in Education, Boston, MA.

Cherry, E.C. (1953). "Some experiments on the recognition of speech, with one and two ears." In: *Journal of the Acoustical Society of America*, **25**. pp. 975-979.

Curtis, P. (1992). "Mudding: Social Phenomena in Text-Based Virtual Realities." DIAC, Berkeley, CA.

Foner, L.N. (1993). "What's an Agent, Anyway? A Sociological Case Study." Cambridge, MA: Agents Group, MIT Media Lab.

Novick, D. G. and K. Ward (1993). "Mutual Beliefs of Mutual Conversants: A Computational Model of Collaboration in Air Traffic Control." In: *Proceedings* *of the Eleventh National Conference on Artificial Intelligence*. Cambridge, MA: The MIT Press.

Schegloff, E. (1993). "Multi-Party Discourse." In: *Burning Issues in Discourse*. NATO Advanced Research Workshop, Maratea, Italy. p. 82.

Sibun, P. (1992). "Generating Text without Trees." In: *Computational Intelligence: Special Issue on Natural Language Generation*, Volume **8**(1). pp. 102-122.

Sibun, P. (1991). *Locally Organized Text Generation.* COINS Technical Report 91-73, Department of Computer and Information Science, University of Massachusetts. Also Report SSL-91-21/P91-00159, Xerox Palo Alto Research Center.

Walker, M.A. (1993). *Informational Redundancy and Resource Bounds in Dialogue*. Ph.D. thesis, University of Pennsylvania.