

cobotDS User Manual*(Version 1.0b)

September 22, 2000

1 Introduction

Many thanks for agreeing to be one of the first users of cobot's new dialogue system, cobotDS. We appreciate your participation and will take your comments and suggestions seriously. The design of any spoken dialogue system is a complex and difficult task that usually takes many iterations and lots of trial and error. Because cobotDS will tackle a number of issues that have not been faced before in dialogue systems research, your help could prove especially valuable and influential in the direction we take and the solutions we adopt. We also hope you will find using cobotDS interesting and fun!

What exactly is cobotDS? In brief, cobotDS is a computerized telephone system that uses speech recognition, text-to-speech and artificial intelligence technology. It allows users to call cobot from any domestic US telephone on a toll-free number, interact with cobot over the phone using spoken English, and have cobot relay messages and information back and forth between the phone user and LambdaMOO.

Using cobotDS, you will be able to interact with the LambdaMOO environment and users over the phone in many of the same ways you already do when in the MOO itself. You will be able to wave, grin, poke, hug, etc. from the phone to a room in general or to a specific user, and LambdaMOO users will be able to pass such emotes to the phone user as well. The phone user will also be able to pass certain (restricted but hopefully sufficient) natural-language utterances to LambdaMOO, and LambdaMOO users will be able to pass arbitrary utterances to the phone user. Additionally, cobotDS will

*cobotDS is a system under active development, and will be undergoing rapid growth and change. We will periodically revise this document accordingly.

provide a number of informational commands for the phone user that allow them to find out what is going on in LambdaMOO, and will eventually replicate at least some of the existing functionality of cobot (social statistics such as verb-o-meter commands, general chat functionality, etc.).

Thus, cobotDS will provide rich multiparty interaction between a phone user, cobot, and the denizens of LambdaMOO. We anticipate that in the early stages of development, the primary use of cobotDS will be to communicate via phone with other human LambdaMOO users, but eventually some phone users may call to interact primarily or partially with cobot himself.

2 A Sample Session

Let's make this a little more concrete by examining both sides (phone user and LambdaMOO) of a typical interaction with cobot. Imagine that cobot is going about his business as usual in the Living Room, and a person (such as yourself) with LambdaMOO user name PhoneUser calls cobot using the toll-free number. The following initial interaction would be typical (where we have introduced numerical labels on the turns for ease of exposition) ¹ :

- 0. cobot: Hi, who is this?*
- 1a. PhoneUser: Phone user.*
- 1b. cobot: Secret password?*
- 2a. PhoneUser: Five one two seven.*
- 2b. cobot: Hi, PhoneUser. What would you like to do?*

Here the utterances of the human PhoneUser are, of course, spoken to cobot over the phone, while the utterances of cobot are spoken using text-to-speech (synthesized speech). At this point, cobot has accepted the password of PhoneUser, and is ready to serve. Meanwhile, back in the Living Room of LambdaMOO, users will have seen an announcement from cobot that he has received a phone call, as in the following sample transcript (again, we introduce numerical labels on the acts, which are not intended to be synchronized with the phone conversation labels):

¹We are still tinkering with the exact wording of the phone-related prompts to both the phone user and LambdaMOO users, so don't be surprised if the actual system differs slightly.

...

1. Buster [to HFh]: We should definitely add the script of the Coen brothers Unabomber biopic.
2. HFh [to Buster]: Cool dat. Also the Blair Witch JonBenet thing.
3. cobot exclaims, ‘‘PhoneUser has just called me on the phone!’’
4. Buster nods to HFh.

...

At this point, users in the LambdaMOO Living Room have been made aware that PhoneUser is on the phone with cobot (line 3), and may expect cobot to soon be conveying messages from PhoneUser into the Living Room.

We return to PhoneUser, who now engages cobot in a series of requests and queries:

- 3a. *PhoneUser: Wave.*
- 3b. *cobot: You wave.*
- 4a. *PhoneUser: Where.*
- 4b. *cobot: I am in the Living Room.*
- 5a. *PhoneUser: Who.*
- 5b. *cobot: I am with Buster, HFh, Betwixt, Natto, and GreenBean.*
- 6a. *PhoneUser: Hug Buster.*
- 6b. *cobot: You hug to Buster.*
- 7a. *PhoneUser: Poke HFh.*
- 7b. *cobot: You poke to HFh.*

In 3a, PhoneUser waves to all those present in cobot’s current location (the Living Room). In turns 4a/b and 5a/b, PhoneUser asks cobot where he is and what other users are there, and receives the answers from cobot. These two turns, being exchanges of information entirely between PhoneUser and cobot, have no visible consequences for the Living Room occupants. However, the acts by PhoneUser in 3a, 6a and 7a might appear to the Living Room as follows:

...

5. cobot holds up a sign that reads: PhoneUser passes on a wave from the phone!
6. Betwixt [to HFh]: The scariest part of Blair Witch JonBenet was the mother.

7. cobot [to Buster]: This is from PhoneUser:
 8. cobot hugs Buster warmly.
 9. HFh [to Betwixt]: Really? It was the tiara collection scene
 I couldn't watch.
 10. Buster [to cobot]: phone: hug
 11. Buster [to cobot]: phone: say How did your interview go?
 12. cobot pokes HFh.
 13. cobot [to HFh]: That was from PhoneUser.
 14. HFh [to cobot]: phone: smile
 ...

Here we see that the emotes dictated by PhoneUser to cobot have now been announced by cobot in the Living Room (lines 5, 7, 8, 12 and 13). Further, Buster and HFh have invoked new functionality on cobot (lines 10, 11 and 14) in order to return PhoneUser's emotes, and also to pass an utterance to PhoneUser. Thus, PhoneUser would now hear something like:

8a. PhoneUser: Cackle GreenBean.
8b. cobot: You cackle to Greenbean. You got a hug from Buster.
9a. PhoneUser: Grin Natto.
9b. cobot: You grin to Natto. You got a smile from HFh. Buster
 says "How did your interview go" to you.
10a. PhoneUser: Say Buster.
10b. cobot: What message would you like to pass?
11a. PhoneUser: No point crying over spilt milk.
11b. cobot: You say "No point crying over spilt milk" to Buster.

Shortly afterwards, cobot will announce PhoneUser's cackle, grin and utterance to Buster in the Living Room.

This interleaved example conveys the spirit and basic functionality of the system we are building. In the remainder of this document, we provide a detailed guide for the use of cobot for the phone user. We emphasize that we have implemented and describe here only the most basic features on this service, and will be rolling out many others over the months to come. Again, we invite suggestions.

3 Talking to the Phone User from LambdaMOO

Before we describe how to use cobot as a phone user, we briefly explain how any LambdaMOO user can pass a verb or message to the phone user through cobot. There are two separate cases, both invoked by directed speech at cobot (via **-cobot** ²). The syntax of the first instance is:

-cobot phone: <verb>

Here **phone:** is a required field telling cobot that the following text is intended for the current phone user, and the **verb** is any verb (such as **hug**, **wave**, etc.). cobot will (privately) acknowledge the LambdaMOO user's command ³, and momentarily the phone user will hear the announcement of the verb from cobot (along with which LambdaMOO user passed the verb).

In the second instance, the syntax is:

-cobot phone: say <utterance>

In this case, the **utterance** is an arbitrary English sentence, word, or phrase. The utterance will be read to the phone user by cobot, along with the identity of the LambdaMOO user sending it. LambdaMOO users should remember that the phone user must process the message over a telephone in real time, and should be careful not to overload the phone user with verbs and messages.

4 Calling cobot on the Phone

Your phone user name for cobot will be the same as your LambdaMOO user name. In addition, we will provide you with a four-digit password code for phone access to cobot; please keep this code private. We will also provide you with cobot's telephone number. This number is a toll-free call from within the United States.

Upon calling cobot, you will be prompted for your user name and password, as in the example given in the Introduction. Please speak your responses to these in a clear and natural manner. (See also Section 7, "Hints

²Other directed speech verbs (such as **page** and **whisper**) may also be invoked with the analogous syntax. The use of such silent direct speech verbs will obviously prevent the passed verb from being publicly announced in LambdaMOO.

³Note that such acknowledgements did not appear in the LambdaMOO transcripts provided in the example of the Introduction, since these transcripts were from the viewpoint of an unseen user.

on Using Speech Recognition”.) Most importantly, when giving your password, please speak each digit individually. Thus, the password 5127 must be spoken “five one two seven”, and not variants such as “fifty one twenty seven” or “five thousand one hundred and twenty seven”. The digit 0 must be pronounced as “zero”, and not “oh”.

Following a successful login (which is attempted a few times before cobot gives up and ends the call), cobot will prompt you with a greeting confirming your user name, and then ask you what you want to do. At this point you are in the main command loop of cobot.

5 The Main Phone Command Loop

Under normal use, the core of a successful call to cobot will consist of a series of dialogue “turns”, alternating between utterances by the phone user and cobot. In a typical turn, the phone user makes a request of some sort to cobot, and in response, cobot confirms the command, and may also provide additional information (such as recent emotes or messages for the phone user from LambdaMOO).

The commands available to the phone user are of the form

`<command> [<user name>]`

where the command is a required utterance, and the user name argument is optional or absent. With a few exceptions (discussed later), all of the choices for the command are standard LambdaMOO social core verbs, such as **wave**, **smile**, **cackle**, and so on. (A list of the verbs currently supported is provided in the Appendix.) Until we have the chance to develop the speech recognition tools to handle a large set of LambdaMOO user names, the allowed user names are restricted to a small group that is listed in the Appendix. (However, the phone user can still emote and speak to the LambdaMOO room in general, and the context of conversation so far may be sufficient for LambdaMOO users to recognize phone user responses to their acts).

As shown in the example in the Introduction, the phone user might typically invoke a sequence of commands on cobot such as **wave buster**, **smile** and **cackle**. In each case, the command will cause cobot to announce the phone user’s actions publicly in whatever room of LambdaMOO cobot currently happens to occupy. When the requested command has an allowed user name as an argument, cobot will announce that the verb is invoked on

that user, while if a user name is absent, cobot will announce that the phone user has invoked the verb to everyone in the room.

Thus, spoken verb invocation on cobot has essentially the same syntax as it has in the textual environment of LambdaMOO, and the announcement of the resulting act is quite analogous as well. However, note that the phone user's verbs will always be conveyed to LambdaMOO via an announcement by cobot, and thus will *not* appear as if they were processed by the LambdaMOO server. Simply put, the metaphor is that the phone user is actually chatting with cobot on the phone, with cobot keeping LambdaMOO informed about the activity in the call where appropriate.

6 The **say** Command, Speech Recognition, and Grammars

An important exception to the basic command invocation on cobot described above is the **say** command. To begin with, when the phone user invokes the **say** command (by speaking, for example, either **say** or **say Buster** or some other user name), cobot will then prompt the user for the message they would like to pass to LambdaMOO. Following this prompt, the user should simply speak the English sentence they would like conveyed — subject to some important and nontrivial constraints.

Here is where some background on speech recognition is required. The goal of speech recognition is to automatically compute an accurate text transcription of a spoken utterance. Highly accurate, speaker-independent speech recognition of arbitrary natural language utterances is beyond current technology. Instead, all speech recognizers make use of some underlying *grammar* describing which utterances are “allowed” by the recognizer. For our current purposes, we may think of this grammar as simply an exhaustive listing of all those words, phrases and sentences that the recognizer will attempt to recognize. Any utterance not contained in this list should not be expected to be recognized. (A typical dialogue system, cobot's included, actually uses several different grammars, each appropriate in different dialogue contexts.)

Why not just set this grammar to be the entire English dictionary, or a phenomenally large corpus of English sentences? The main reason is that there is a trade-off between the size or complexity of the grammar and the accuracy of recognition. Simply put, a very small grammar will result in highly accurate recognition, but only of those items appearing in this lim-

ited grammar; while a very large grammar may in principle permit more expressive utterances, but with frequently poor accuracy (words being mis-recognized, dropped or added). For this reason, most applications of speech recognition attempt to balance the burden placed on users (by forcing them to use restrictive language) with the burden placed on the speech recognizer (by having to cope with many possible utterances). In many cases, this trade-off has a nice intermediate solution. For example, the grammar we have written for the main command loop of cobot is mercifully small and simple, since it exploits the fact that there are rather few verbs in the social core (and we have artificially limited the number of possible user names for now).

However, the **say** command as it is used in LambdaMOO is meant to be a powerful, general-purpose verb for social communication, and we would like to recapture at least some of this power for the cobot phone user. Our approach is to make several different grammars available for use with the **say** command, and to allow the phone user to choose which of these is active during any given invocation of **say**. We have initially implemented two grammars, and will allow phone users to provide us with a third “personal” grammar that can be modified by them periodically. We now describe each of these grammars in turn.⁴

6.1 The Cliche Grammar

The Cliche grammar is the default for the **say** command, and will always be active during a **say** unless another grammar is explicitly set by the phone user via the **grammar** command (described shortly). The Cliche grammar consists of approximately 3000 common English cliches, sayings and phrases. (A full listing of the items in the Cliche grammar will be provided to phone users in a separate document.) When the Cliche grammar is active, the phone user invoking the **say** command will be able to utter an item in the grammar, whereupon it will (hopefully) be recognized successfully and passed on to LambdaMOO by cobot.

Although a phone user’s **say** utterances using exclusively the Cliche grammar would be highly stylized, the grammar is rich enough to contain appropriate remarks and responses for many conversational situations [Becker reference]. However, there are several difficulties in relying solely on the Cliche grammar to provide the phone user with a rich set of utterances:

⁴We emphasize that these grammars are simply our starting point, and that we are open to suggestions for additional grammar sources and types.

- Searching for the right phrase in such a large grammar in real time during a live conversation while on the phone is difficult.
- Being a list of cliches and common sayings, the grammar is missing some obvious conversational necessities, such as basic greetings, questions, and replies (How are you?, I am at work, yes/no, etc.).
- The grammar is also missing particular utterances that a particular phone user might find useful, either repeatedly or even for short periods of time.
- Even though the Cliche grammar is a highly restricted fragment of the English language, phone users will still find the speech recognition sometimes inaccurate.

For these reasons, we view the Cliche grammar as serving a mixture of practical and entertainment purposes, not a general-purpose conversational mechanism. We attempt to partially address the second and third items in the list above by the introduction of the Small Talk and Personal grammars.

6.2 The Small Talk Grammar

The Small Talk grammar is meant to provide a more basic and coherent set of useful utterances for common conversation than is provided by the highly stylized Cliche grammar. We are building the Small Talk grammar by hand, adding common phrases as we think of them, and solicit suggestions for additions. The Small Talk grammar contains staples such as “How are you”, “Hello”, “I am fine”, “Yes”, “I am at home”, and many others. A complete listing of the elements of the Small Talk grammar, along with a simple Perl script for searching the grammar by some indexed categories (such as greeting phrases, basic responses, phrases about domestic or work life, etc.), will be provided to phone users in a separate document. Again, any time the Small Talk grammar is active, all **say** utterances will be recognized using the Small Talk grammar only.

6.3 The grammar Command

We now have two different grammars (Cliche and Small Talk) that can be applied to the utterance of a **say** command. How can the phone user choose which grammar should be activated on any given **say**? The answer is the **grammar** command, which can be invoked at the main command loop by

simply saying “grammar”. cobot will then prompt the phone user for a grammar name, which currently must be one of the utterances “cliche”, “small talk”, or “personal”. The successful invocation of the grammar command will result in the active grammar being set to the designated argument. This grammar will be applied to *all* subsequent **say** commands until the grammar is changed again, or the call terminates. Again, the default grammar (active at the start of every call, or made active any time a **grammar** command fails) is the Cliche grammar.

6.4 Personal Grammars

The last grammars that we are providing in our initial system are those provided by the phone users themselves. Despite our best attempts to provide some basic coverage of common useful utterances through the Cliche and Small Talk grammars, we will inevitably miss many particular phrases that individual users may want to be able to pass using the **say** command. We thus invite each phone user to provide us with an evolving list of such personalized phrases. When a phone user successfully invokes the command **grammar personal**, cobot will set the active grammar to be the personal grammar provided by the particular user (as identified during login to cobot).

We will solicit source material for each personal grammar from phone users. This will typically be provided as a simple text file of English phrases, words, and sentences, all lower case, and one to a line. We will take this source material and compile it as a grammar, after which users should be able to pass utterances from the source using the **say** command with active grammar set to personal. The mechanism by which users can update and replace their personal grammars are still to be determined, as is the frequency with which we will perform such updates (updating grammars requires both a compilation process and for cobot to be taken down). We will keep our phone users informed of the necessary details, but it is safe to expect that phone users will be able to modify their grammars roughly weekly but not daily.

The use of personal grammars is one of the more interesting research aspects of cobot, and we invite creativity by phone users in choosing their source material. For example, some users may want to hand-craft a lengthy, slowly changing list of favorite utterances and sayings not covered by the other grammars, or to create specialized functional phrases appropriate to the details of their life (“I am at Wanda’s house”, “I am playing Ultimate

today”). Others may want to manage their grammar more dynamically, anticipating and swapping in phrases that will be useful only for a short time following the grammar update (“I am calling from my vacation in Wyoming”). Still others may want to experiment with adding text from favorite books, movies, poems, and so on.

7 Hints on Using Speech Recognition

As you might have guessed from the last section, speech recognition is an imperfect technology. Performance can vary widely depending on many factors: the particular utterance, the particular speaker, the particular phone called from. Here are some hints for trying to get the best performance:

- Try to minimize the amount of acoustic noise in your calls. Calls made from a quiet room or phone booth on a land-line phone are likely to do better than calls made from a cell phone at a busy intersection. (Nevertheless, feel free to try it anywhere — we are interested in the system being used over a range of conditions and environments — just try not to get frustrated.)
- Try to speak clearly and articulate your words carefully, without being unnatural. In particular, speaking deliberately slowly and drawing out the syllables may result in worse performance rather than better. (Nevertheless, do experiment around to see what seems to give the best performance for you.)

8 Informational Commands: where and who

The commands **where** and **who**, which take no arguments, will cause cobot to report to the phone user which room of LambdaMOO cobot currently occupies, and which other LambdaMOO users are also in that room, respectively.

9 Other Commands: repeat, sleep, help and quit

Other commands available at the main command loop of cobot include **repeat**, **sleep**, **help** and **quit**, none of which takes any arguments.

The **repeat** command will simply cause cobot to repeat the last prompt to the phone user, including any emotes or messages passed to the phone

user from LambdaMOO. Since the synthesized speech of arbitrary messages passed from LambdaMOO can sometimes be difficult to understand on first hearing, the **repeat** command allows the phone user to replay such messages before moving on to their next command. Note that past prompts and messages are lost once a non-**repeat** command is given. Later versions of cobot may provide more elaborate “audio scrollbar” functionality.

There may be times when the phone user needs time to contemplate their next action, or is awaiting a response to an earlier verb pushed to LambdaMOO. During such periods, it is inconvenient to have the main command loop continually prompting the phone user for the next action. The **sleep** command is designed to force cobot to idle at the phone user’s command. After uttering the “sleep” command, cobot will announce that he is now sleeping, and a silent period will ensue. During this period, cobot will not prompt the phone user for additional commands, but will continue to announce the arrival of verbs pushed from LambdaMOO. At any point during the silent period, the phone user may wake cobot by simply uttering their next command.

The **help** command will eventually provide on-line audio help for phone users, but is not implemented yet.

The phone user should utter the **quit** command to terminate a call to cobot. Please make your best effort to end calls explicitly with **quit**, rather than simply hanging up in the middle of a dialogue.

10 Appendix

10.1 Listing of Phone Commands

The following verbs from the LambdaMOO social core are currently available as cobot phone commands: **blush**, **bow**, **cackle**, **chuckle**, **comfort**, **cringe**, **cry**, **french**, **giggle**, **grin**, **hug**, **kiss**, **laugh**, **nod**, **poke**, **say**, **shrug**, **sigh**, **smile**, **smirk**, **wave**, **whisper**, **wink**, **yawn**. All accept an optional user name argument. As already discussed, the **say** command will cause cobot to prompt the user for an utterance. In addition to the social core verbs, cobot also recognizes the no-argument commands **grammar**, **help**, **repeat**, **sleep**, **where**, **who**, **quit**. The **grammar** command will cause cobot to prompt for a grammar name.

10.2 Current LambdaMOO Phone User List

The current list of LambdaMOO users with a cobot phone user account are **AcidHorse**, **Betwixt**, **Boiledhead**, **Buster**, **CrashLander**, **Cyberat**, **Gabaldon**, **Goat**, **GreenBean**, **HFh**, **Huey**, **LuciaLibra**, **Milly**, **Natto** and **Oedipus**. LuciaLibra should be referred to as “Lucia” due to trouble we had with the speech recognizer on this name. These are also the only LambdaMOO user names that cobot will recognize as arguments to the social core verbs.

10.3 Privacy Policy

The cobotDS system will of course keep system logs that include your dialogues with the system and the phone number you call from. All of this information will be kept absolutely confidential, and will not be accessible to anyone outside of our small research group of approximately five people, all of whom are directly working on the project. In general, we will not use this information for anything other than debugging and improving the system.

10.4 Contacting Us

We solicit your feedback and suggestions on the use, functionality and design of cobotDS, and invite you to keep in touch with us both in LambdaMOO and via email. We have monitoring tools that will automatically inform us when cobotDS is not working, but if you notice anything especially anomalous, please let us know. The contacts for cobotDS are Charles Isbell (LambdaMOO user HFh, RL email isbell@research.att.com) and Michael Kearns (LambdaMOO user Buster, RL email mkearns@research.att.com).