EVALUATION FOR DARPA COMMUNICATOR SPOKEN DIALOGUE SYSTEMS

Marilyn Walker*, Lynette Hirschman† and John Aberdeen†

*AT& T Labs - Research
180 Park Ave, Florham Park, N.J. 07932, U.S.A.
{walker@research.att.com}
† The MITRE Corporation
202 Burlington Rd., Bedford, MA 01730, U.S.A.
{lynette,aberdeen@mitre.org}

Abstract

The overall objective of the DARPA COMMUNICATOR project is to support rapid, cost-effective development of multi-modal speech-enabled dialogue systems with advanced conversational capabilities, such as plan optimization, explanation and negotiation. In order to make this a reality, we need to find methods for evaluating the contribution of various techniques to the users' willingness and ability to use the system. This paper reports on the approach to spoken dialogue system evaluation that we are applying in the COMMUNICATOR program. We describe our overall approach, the experimental design, the logfile standard, and the metrics applied in the experimental evaluation planned for June of 2000.

1. Introduction

The overall objective of the DARPA COMMUNICATOR project is to support rapid, cost-effective development of speech-enabled dialogue systems. Current commercial technology for speech-enabled interfaces has made rapid progress over the past decade. There are increasing numbers of systems deployed in commercial applications that provide structured system-initiated interaction. These systems work by controlling the conversation, requesting that the user provide a specific kind of information at each turn. However, these systems do not yet have true conversational capability. Building robust systems that can engage in true mixed initiative interaction is still very much an open research issue. Conversational systems should be able to interact naturally with the user, supporting both user and system intiative, providing clarification, negotiation and ability to recover from user and system errors. Exploring the issues of mixed initiative conversational interaction are the focus of the DARPA COMMUNICATOR program.

There are several enabling goals for the COMMUNICATOR program. These are:

- To provide a common architecture, so that researchers can furnish subcomponents (dialogue management, or generation or synthesis) without having to build an entire system.
- To provide a testbed with sharable components that lower the entry bar to building speech-enabled dialogue systems.
- To provide a shared research environment, including common data and a common evaluation framework, to encourage cross-group comparison and rapid sharing of technological innovations.
- To further innovative research on dialogue management and interface design to support conversational systems.
- To encourage the transfer of this technology to real users, in particular, military users.

The program has chosen MIT's Galaxy II architecture (Seneff et al., 1999; Polifroni and Seneff, 2000) as its common architecture. This architecture uses a scriptable hub to provide routing and program control, in conjunction with servers that do the actual processing, such as speech recognition, natural language processing, dialogue management, generation, and synthesis.

A number of groups are now building systems using the Galaxy architecture and hub, coupled with in-house developed servers. These systems provide end-to-end functionality in the initial COMMUNICATOR challenge task, air travel planning, shown in Figure 1. To complete tasks such as this, COMMUNICATOR systems should engage the user in an intelligent conversational interaction, where both user and the system can initiate interaction, provide information, ask for clarification, signal non-understanding, or interrupt the other participant.

In order to make progress on the key research issues to sup-

You are in Denver, Friday night at 8pm on the road to the airport after a great meeting. As a result of the meeting, you need to attend a group meeting in San Diego on Point Loma on Monday at 8:30, a meeting Tuesday morning at Miramar at 7:30, then one from 3-5 pm in Monterey; you need reservations (car, hotel, air).
You pull over to the side of the road and whip out your Communicator. Through spoken dialogue (augmented with a display and pointing), you make the appropriate reservations, discover a conflict, and send an e-mail message (dictated) to inform the group of the changed schedule. Do this in 10 minutes.

Figure 1: DARPA COMMUNICATOR Challenge Problem

port intelligent interaction, we need to measure the efficacy of different techniques. This means that we need to find methods for measuring the contribution of these techniques to the users' willingness and ability to use the system. Thus evaluation of advanced dialogue systems becomes a central research issue in its own right in the DARPA COMMUNICATOR program. The COMMUNICATOR research community has chosen to build on previous research in evaluation of conversational interaction by extending and refining the PARADISE framework for evaluating spoken dialogue systems (Price et al., 1992; Hirschman et al., 1993; Hirschman, 2000; Walker et al., 1997).

The PARADISE framework provides a methodology for learning general performance functions for spoken dialogue systems from experimental dialogue data. The framework posits that user satisfaction is the overall objective to be maximized and that task success and various interaction costs can be used as predictors of user satisfaction. PARADISE has been applied to data from several spoken dialogue systems performing different tasks; the results so far suggest that it is possible to learn a performance function on data for one system and use that as the performance function for another system (Walker et al., 2000b; Walker et al., 2000a). However, to date, PARADISE has only been applied to dialogue data collected in controlled experiments.

The evaluation program for DARPA COMMUNICATOR will extend current results applying PARADISE in several ways. First, to provide better insight into dialogue issues early on in the program, the COMMUNICATOR program is encouraging the use of real subjects who access real (useful) resources. This has been shown to be an effective way of quickly (and cheaply) collecting data from real users (Polifroni et al., 1998). We can use this approach to define an evaluation experiment consisting of dialogues collected with *open* tasks, i.e. tasks that the users define themselves, as well as predefined task scenarios. This will be the first opportunity we have had to apply PARADISE to open tasks. Second, we hope to draw on recent research to develop and utilize a broader set of metrics as the predictors of user satisfaction (Sparck-Jones and Galliers, 1996; Bernsen et al., 1996; Sanderman et al., 1998; Rudnicky, 1993; Eskenazi, 1999). These will include task completion, diagnostic and dialogue quality, and efficiency metrics instrumented in a consistent way across all systems. Third, a large numbers of systems are involved in the evaluation of the COMMUNICATOR travel task. Systems from eight current COMMUNICATOR sites are planning to participate in the evaluation.[1] Thus the evaluation will provide data for modeling cross-system performance to a much greater extent than in previous work. Fourth, the tasks will be run on systems with databases with potentially different content since all of the systems are running against (different) live databases.

The evaluation data will be analyzed with two goals in mind. First, we will use the task completion metric as the basis for comparing the systems, in order to verify that the systems are fundamentally capable of completing a representative set of travel tasks. Second, we will analyze the data using the PARADISE framework to learn an objective performance function and we will examine the generalizability of this performance function across different systems, user populations, and task types. The evaluation data collection will take place in June of 2000, with the results analyzed and reported out in September 2000.

There are several other facets to our evaluation efforts which we will not discuss in this paper. We are involved in efforts to evaluate (1) the portability of the techniques used by the systems; (2) the extent to which the systems are applying innovative techniques; (3) the learnability of the systems; and (4) how these systems compare on the travel planning tasks to human travel agents and other available technologies.

Section 2. explains the experimental design used for the data collection. Section 3. describes the logfile standard used by all the systems and discusses in detail the metrics that each system has been instrumented to log. Finally section 4. discusses our plans and future directions.

2. Experimental Design

The evaluation will be a controlled experiment in which a set of realistic subjects from the target population of frequent travelers will interact with each system to complete a set of 9 realistic scenarios of varying task complexity. Since we will evaluate 9 systems, we will recruit 81 subjects (9*9). The subject groups will be run in three clusters of three days each to balance the load on the systems. Over a three day period, each subject will call each system and use dialog interaction to plan travel tasks according to 9 different scenarios to be discussed in more detail below. Subjects will carry out the scenarios in a fixed order, with scenarios becoming progressively harder. The system factor will be counter-balanced; subjects will start the scenarios with different systems. Thus each system will get a total of 81 calls over 3 periods of 3 days, resulting in a corpus of 729 dialogues for evaluating and comparing systems, consisting of dialogues with 81 different users for each system, with 9 dialogues for each task scenario per system.

The task scenarios will consist of 7 *fixed* and 2 *open* scenarios. The *fixed* scenarios are designed to vary task complexity. We are exploring a definition of task complexity that consists of two components: a user-input component corresponding to the number of constraints that have to be communicated to the system, and a system-output component corresponding to the number of travel itineraries that match the constraint set which then must be filtered by interaction between the system and the user.

The *open* scenarios are tasks that are defined by the user. After completing 7 pre-defined tasks with 7 of the systems, the users will be instructed to use the system to "plan a recent or intended trip". By asking the users to define their own tasks, the *open* scenarios are intended to approximate the conditions

[1] An experimental system that NIST is putting together to explore issues with plug-and-play will also participate in the evaluation data collection, although it will not be evaluated.

under which these systems would actually be used by the intended user group in the field (Baggia et al., 1998). Recent work has argued that dialogue data collected with fixed scenarios is not realistic (Larsen, 1999). However, to our knowledge, no quantitative or qualitative assessment of the differences between these modes of data collection has ever been published. The combination of fixed and open scenario dialogue data from the same user will allow us to describe in detail any significant differences in dialogues collected with scenarios defined by the experimenters vs. those defined by the intended user population, and calibrate the extent of any differences found.

At the end of each call, each user will fill out a survey giving their subjective evaluation of the system's performance. The survey will be described in section 3. The dialogues will be recorded in full and each site will produce a logfile consistent with the COMMUNICATOR logfile standard (discussed below in section 3.), as well as a set of recordings of the user's utterances. After the data collection is completed, the recordings and logfiles will be redistributed to the sites for labelling and transcription.

This experimental setup provides a unique opportunity to push research on evaluation forward because of the large number of systems performing the same task and the desire of members of the COMMUNICATOR community to test their systems with realistic users doing realistic tasks. The systems participating in the experiments have been instrumented to provide much more data relevant to evaluation than will be used in the comparative evaluation based on task completion.

3. LogFile Standard and Metrics Used

Here we discuss in detail the logfile standard and the metrics that will be collected in the evaluation experiment. These metrics will provide the data necessary to apply PARADISE which will allow us to develop models of the relationship between a representative set of objective metrics and user satisfaction (Walker et al., 1997; Walker et al., 2000b).

As mentioned above, at the end of each call, users will fill out a web-based survey before going on to the next task. The web survey is the basis for calculating perceived Task Success and User Satisfaction measures. Users report their perceptions as to whether they have completed the task via the yes/no survey (**Perceived Completion**) question in Figure 2.[2] The User Satisfaction questions on the survey probe a number of different aspects of the users' perceptions of their interaction with the system in order to focus the user on the task of rating the system, as in (Shriberg et al., 1992; Jack et al., 1992; Love et al., 1994). The User Satisfaction questions are all stated in terms of positive dimensions of the system; the user has to state the degree to which they agree with these statements in terms of a 5 point multiple choice Likert scale. Each survey response is then mapped into the range of 1 to 5 and the values for all the responses are summed, resulting in a **User Satisfaction** measure for each dialogue ranging from 5 to 25.

[2] *Yes,No* responses are converted to *1,0*.

- Were you able to successfully complete your task? (**Perceived Completion**)
- In this conversation, it was easy to get the information that I wanted. (**Task Ease**)
- I found the system easy to understand in this conversation. (**TTS Performance**)
- In this conversation, I knew what I could say or do at each point of the dialogue. (**User Expertise**)
- The system worked the way I expected it to in this conversation. (**Expected Behavior**)
- Based on my experience in this conversation using this system to get travel information, I would like to use this system regularly. (**Future Use**)

Figure 2: User Survey assessing Perceived Task Completion and User Satisfaction

The objective metrics focus on measures that can be automatically logged or computed. They include diagnostic metrics that are comparable across systems for evaluation of component modules, as well as dialogue manager and whole dialogue metrics. Measures are summarized in Figure 3 and described in more detail below.

- **Dialogue Efficiency Metrics**
  - Total elapsed time, Time on task, System turns, User turns, Turns on task
  - Time per turn for each system module
- **Dialogue Quality Metrics**
  - Word error rate, Reprompts, Error messages, Help messages, Response latency.
  - Mean word error rate, Reprompt %, Mean response latency, Variance reponse latency, Help %
- **Task Success Metrics**
  - Peceived task completion, Objective task completion
- **User Satisfaction**
  - Sum of TTS performance, Task ease, User expertise, Expected behavior, Future use.

Figure 3: Metrics collected for spoken dialogues.

To facilitate the evaluation, we have developed a logfile standard that all systems are using in their data collection. Our goal is to establish standards for both the content and format of the logs. By establishing minimum standards for content,

we can ensure that all sites collect the data necessary to calculate the desired objective metrics. Specifying the format has enabled us to develop automated tools for validating logs and calculating metrics. (The format uses XML, the Extensible Markup Language.)

The overall log corresponds to of a number of *sessions* (typically 1) with the system where each session is composed of a number of *system turns* and *user turns*. Each system and user turn contains some number of *operations* (commands executed by the system within a turn), *messages* (items sent by the various servers in a system, as well as their replies), and *events* (such as errors, locks and alarms). Operations, messages, and events may contain *data* in the form of key/value pairs. All elements are time stamped, to facilitate the calculation of durations. Below we describe the elements of the logfile standard that relate to the objective metrics we wish to calculate. Further details about the structure (as well as the XML format) of the logfile standard are available at http://fofoca.mitre.org/logstandard.

Consider the sample dialogue in Figure 4, which we will use to describe the logging.[3] This dialogue can be broadly divided into three sections. The region from utterance S00 through S01 is a prelude in which the system identifies the user, and the region from utterance S09 through S10 is a follow-on section in which the system asks the user a question about her experience with the system. The remaining utterances U01 through U08 represent the on-task portion of the dialogue. The logfile standard encodes these regions with attributes that mark the start and end of the task.

By logging system and user turns we can easily calculate the total number of turns in the session (21), as well as the number of system turns (11), number of user turns (10), and the number of turns on task (15). We also log what the system says at each turn of the dialogue and we have human transcriptions of each user turn (human transcriptions are kept in a file separate from the main logfile). From these two sources of information we can calculate the number of user words in a turn and the number of system words in a turn, and the mean number of user words per turn and system words per turn over the whole dialogue. Because we also have start and end of the task marked in the logfile, we can also calculate the latter metrics for just the on-task portion of the dialogue. The logfile standard also encodes the selected automatic speech recognition hypothesis for each user turn. This, coupled with the human transcription, allows us to calculate word error rate.

As mentioned above, all elements in a logfile are time stamped. This, along with the logfile characteristics described above, enables the calculation of several dialogue efficiency metrics, such as total elapsed time, time on task, mean length of system turn, and response latency. Response latency is calculated by subtracting the value of the end-time attribute of a user utterance tag from the start-time attribute of the following system utterance tag. An example of a COMMUNICATOR XML-based logfile is given in Figure 5 in the Appendix.

Utterance S05 is an instance of the system prompting the user for a specific piece of information. In the logfile standard such prompts are logged along with a type-prompt attribute whose value is the key being prompted for (departure time in this case). Notice that (due to some error) the next system utterance (S06) also prompts for the same key. A postprocess looks for consecutive instances of the same value for the type-prompt attribute, and infers that a system repeat/reprompt has occurred.

The logfile standard also has a provision for logging of various utterance types or speech act types such as help messages, confirmations, or error messages. We are currently asking systems to log as a help message or an error message any utterance that they consider to belong in one of those categories. In the future we hope to make this classification of utterance types more precise and extend the taxonomy to distinguish between different communication goals of the system.

The logfile standard also has a provision for encoding task completion. For fixed scenarios, this judgment is made by human transcribers and is encoded in the human transcriptions file. For open tasks, the judgment may be made by the user in the response to a follow-on question presented to the user either as part of the dialogue (S09, U09), or as part of the user satisfaction survey as in Figure 2.

Table 1 shows sample scorer output for a subset of the metrics that can be calculated by our automated scoring tools for the sample call in Figure 4. After human transcription and human assessment of task completion, all of the metrics can be calculated automatically using the logfiles generated in logfile standard format.

Finally, there are a number of concept accuracy metrics for which we have preliminary definitions, but which are not yet included in the logfile standard because their definitions are still under discussion. These metrics include mean number of user concepts per turn, mean concept efficiency, user repeats, and state of itinerary. However, some of these metrics are being calculated on a site specific basis. For example, MITRE annotates dialogues indicating, for each user utterance, the concepts in the utterance and the concepts captured by the system. In the dialogue in Figure 4, utterance U01 contains three concepts: a speech act (identify a flight), a departure location (Boston) and an arrival location (San Francisco). The system representation of this utterance also contains exactly these three concepts (determined by examining the logfile). By contrast utterance U05 contains a concept (departure time) which was not captured by the system. By annotating concepts in this way we can then compute concept precision and recall scores. Once a common definition of new metrics, such as concept accuracy, has been reached, these will be directly supported by the logfile standard and the associated software tools.

[3] This dialogue has been constructed from similar dialogues from several sites, to provide a simple example of human-system interaction for illustrative purposes. The figures provided in Table 1 are also illustrative.

S00: Welcome to the Communicator Travel Demonstration System. Please say your name.
U00: Jane Smith.
S01: Hello Jane, how may I help you?
U01: I'd like to fly from Boston to San-Francisco.
S02: Can you provide the approximate departure time?
U02: I'd like to leave in the morning on Friday.
S03: United Airlines flight one sixty nine departs Boston at seven A-M and arrives San-Francisco at ten thirty five A-M. Can I help you with something else?
U03: I need an American flight American Airlines.
S04: American Airlines flight one ninety five departs Boston at eight A-M and arrives San-Francisco at eleven thirty three A-M. Is there something else I can do for you?
U04: Yeah I need to come back I think the following Wednesday.
S05: Can you provide the approximate departure time?
U05: Um yeah I suppose I should leave sometime early in the morning.
S06: Can you provide the approximate departure time?
U06: I'd like to leave around seven in the morning.
S07: American Airlines flight one ninety six departs San-Francisco at seven thirty A-M and arrives Boston at four oh nine P-M. What else can I help you with?
U07: Please book the flights.
S08: I have booked the flights. Is there anything else?
U08: No thanks.
S09: Did you get the information you were seeking?
U09: Yes I did.
S10: Goodbye Jane. A copy of your itinerary has been emailed to you. Thank you for using the Communicator Travel Demonstration System.

Figure 4: Sample COMMUNICATOR Dialogue

Task completed	On-task duration (secs)	Total task duration (secs)	Turns to task end	Mean user words/turn	Mean system words/turn	Error messages
1	97.6	109.4	15	7.8	15.6	0

Help messages	Mean response latency (secs)	User words to task end	System words to task end	Number of reprompts	Mean user turn duration	Mean system turn duration
0	5.1	63	109	1	2.0	5.2

Table 1: Sample scores

4. Discussion and Future Work

A second evaluation of COMMUNICATOR travel systems is scheduled for February 2001. By this date, we plan to have new metrics for concept accuracy, as well as experimental metrics to assess output quality (both generation and synthesis). We also perceive a need for additional metrics that will help us calculate the quality of the solution that the system presents to the user, since users will evaluate the system's performance in terms of tradeoffs between price of the trip and convenience. In addition, we plan to refine our ability to collect and evaluate data from real users, as part of our overall evaluation.

5. Acknowledgements

Thanks are due to the members of all the COMMUNICATOR sites (AT&T, BBN, CMU, University of Colorado, IBM, MIT, MITRE, SRI) who participated in the COMMUNICATOR evaluation committee and contributed to the workshops and discussions that resulted in the current plans for COMMUNICATOR evaluation.

6. References

Baggia, Paolo, Giuseppe Castagneri, and Morena Danieli, 1998. Field trials of the italian arise train timetable system. In *Interactive Voice Technology for Telecommunications Applications, IVTTA*.

Bernsen, Niels Ole, Hans Dybkjaer, and Laila Dybkjaer, 1996. Principles for the design of cooperative spoken human-machine dialogue. In *International Conference on Spoken Language Processing, ICSLP 96*.

Eskenazi, Maxine, 1999. User come back. Presentation to DARPA Communicator Compare and Contrast Meeting, June 16-17.

Hirschman, L., M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann, 1993. Multi-site data collection and evaluation in spoken language understanding. In *Proceedings of the Human Language Technology Workshop*.

Hirschman, Lynette, 2000. Evaluating spoken language interaction: Experiences from the darpa spoken language program 1990–1995. In S. Luperfoy (ed.), *Spoken Language Discourse*. Cambridge, Mass.: MIT Press.

Jack, M.A., J. C. Foster, and F. W. Stentiford, 1992. Intelligent dialogues in automated telephone services. In *International Conference on Spoken Language Processing, ICSLP*.

Larsen, Lars Bo, 1999. Combining objective and subjective data in evaluation of spoken dialogues. In *ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*.

Love, Stephen, R. T Dutton, J. C. Foster, M. A. Jack, and F. W. M. Stentiford, 1994. Identifying salient usability attributes for automated telephone services. In *International Conference on Spoken Language Processing, ICSLP*.

Polifroni, J. and S. Seneff, 2000. Galaxy-ii as an architecture for spoken dialogue evaluation. In *Second International Conference on Language Resources and Evaluation (LREC)*.

Polifroni, J., S. Seneff, J. Glass, and T. Hazen, 1998. Evaluation methodology for a telephone-based conversational system. In *Proc. First International Conference on Language Resources and Evaluation, Granada, Spain*.

Price, Patti, Lynette Hirschman, Elizabeth Shriberg, and Elizabeth Wade, 1992. Subject-based evaluation measures for interactive spoken language systems. In *Proceedings of the DARPA Speech and NL Workshop*.

Rudnicky, A. I., 1993. Factors affecting choice of speech over keyboard and mouse in a simple data-retrieval task. In *EUROSPEECH93*.

Sanderman, Angielen, Janienke Sturm, Els den Os, Lou Boves, and Anita Cremers, 1998. Evaluation of the dutchtrain timetable information system developed in the arise project. In *Interactive Voice Technology for Telecommunications Applications, IVTTA*.

Seneff, S., R. Lau, and J. Polifroni, 1999. Organization, communication, and control in the galaxy-ii conversational system. In *Proc. Eurospeech 99*.

Shriberg, Elizabeth, Elizabeth Wade, and Patti Price, 1992. Human-machine problem solving using spoken language systems (SLS): Factors affecting performance and user satisfaction. In *Proceedings of the DARPA Speech and NL Workshop*.

Sparck-Jones, Karen and Julia R. Galliers, 1996. *Evaluating Natural Language Processing Systems*. Springer.

Walker, M. A., D. Litman, C. A. Kamm, and A. Abella, 1997. PARADISE: A general framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, ACL/EACL 97*.

Walker, Marilyn, Candace Kamm, and Julie Boland, 2000a. Developing and testing general models of spoken dialogue system performance. In *Proc. Language Resources and Evaluation Conference, LREC-2000*.

Walker, Marilyn A., Candace A. Kamm, and Diane J. Litman, 2000b. Towards developing general models of usability with PARADISE. *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems*.

7. Appendix: Log Files

Figure 5 shows a sample of the structure of a logfile that conforms to the standard. The structure is hierarchical, with a top level GC-LOG element (line 1) enclosing a GC-SESSION element. (GC is a prefix indicating Galaxy Communicator.) Within the GC-SESSION element is a sequence of GC-TURNS, e.g., lines 3, 11, 21). Each GC-TURN may contain GC-OPERATIONS, GC-MESSAGES and GC-EVENTS, any of which may contain GC-DATA e.g., lines 5, 7, 10, 14, 16, 18, and 20. Several portions of the logfile have been elided (some GC-OPERATIONS, all GC-MESSAGES and GC-EVENTS, as well as several entire GC-TURNS) so that the overall structure can be illustrated in a small space. In addition, the file has been formatted and given line numbers for readability. Time stamps are given as start time (stime) and end time (end time) in seconds. The final logfile is a result of postprocessing to associate the end-times with the start-times of events.

```
1. <GC-LOG logfile-version="2.0">
2.   <GC-SESSION  id="None" stime="928870182.260000"
                            etime="928870243.640000">
3.     <GC-TURN id="-1" stime="928870182.260000"
                        etime="928870193.740000">
         ...
4.       <GC-OPERATION name="speak-output" server="tts"
                       location="localhost:15020" turnid="-1"
                       stime="928870182.290000"
                       etime="928870182.790000" tidx="238">
5.         <GC-DATA key=":reply-string" dtype="unknown">
             Welcome to the Communicator Travel Demonstration System.
             Please say your name.
           </GC-DATA>
           ...
         </GC-OPERATION>
         ...
6.      <GC-OPERATION name="" server="None" location="None" turnid="-1"
                     stime="928870183.460000" etime="928870183.460000">
7.         <GC-DATA key=":playing-has-begun" dtype="unknown"/>
8.       </GC-OPERATION>
9.       <GC-OPERATION name="" server="None" location="None" turnid="-1"
                       stime="928870193.740000"
                       etime="928870193.740000">
10.        <GC-DATA key=":playing-has-ended" dtype="unknown"/>
         </GC-OPERATION>
       </GC-TURN>
11.    <GC-TURN id="0" stime="928870193.750000"
                       etime="928870230.750000">
12.      <GC-OPERATION name="enable-input" server="audio"
                       location="localhost:15000" turnid="0"
                       stime="928870193.750000"
                       etime="928870194.010000" tidx="246">
           ...
         </GC-OPERATION>
13.      <GC-OPERATION name="" server="None" location="None" turnid="0"
                       stime="928870193.970000"
                       etime="928870193.970000">
14.        <GC-DATA key=":listening-has-begun" dtype="unknown"/>
         </GC-OPERATION>
15.      <GC-OPERATION name="" server="None" location="None" turnid="0"
                       stime="928870194.240000"
                       etime="928870194.240000">
16.        <GC-DATA key=":recording-has-begun" dtype="unknown"/>
         </GC-OPERATION>
         ...
17.      <GC-OPERATION name="" server="None" location="None" turnid="0"
                       stime="928870202.850000"
                       etime="928870202.850000">
18.        <GC-DATA key=":recording-has-ended" dtype="unknown"/>
         </GC-OPERATION>
         ...
19.      <GC-OPERATION name="create-frame" server="nl"
                       location="localhost:11000" turnid="0"
                       stime="928870218.050000"
                       etime="928870218.110000" tidx="297">
           ...
20.        <GC-DATA key=":input-string" dtype="unknown">
                    " <> <pause1> jane smith <pause2> <>"
           </GC-DATA>
           ...
         </GC-OPERATION>
         ...
       </GC-TURN>
       ...
21.    <GC-TURN id="19" stime="928870230.770000"
                        etime="928870243.640000">
         ...
       </GC-TURN>
     </GC-SESSION>
   </GC-LOG>
```

Figure 5: Example of Logfile standard logging