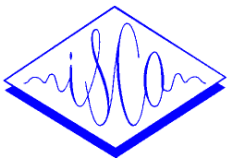

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

WEB-BASED MONITORING, LOGGING AND REPORTING TOOLS FOR MULTI-SERVICE MULTI-MODAL SYSTEMS

G. Di Fabbrizio and S. Narayanan
AT&T Labs – Research
180 Park Avenue, Florham Park, NJ, 07932, USA
{pino,shri}@research.att.com

ABSTRACT

This paper describes **MILER** (**M**ulti-modal data **L**ogger for **E**valuation and **R**eport), a web-based multi-service monitoring, logging and reporting tool for advanced multi-modal dialog systems. MILER has been designed to directly arrange and synchronize logging data collected from live services and to provide real-time reports about service usage and system performance. Special attention has been given to the architecture design in order to achieve service and access-device independence and reliable synchronization of data from distributed logs. MILER allows researchers to analyze multi-modal interactions, analyze the call flow, reconstruct the system/user dialogue turns, play the recorded user utterances, and provide a preliminary dialogue performance evaluation. It also supports labeling and annotation of the dialogue turns for further offline analysis. Once the user inputs (i.e. speech and other input modalities) are manually transcribed and labeled, along with detailed log events from each dialog, MILER derives a set of objective measures, which includes word and concept accuracy, number of attempts per concept, dialog turn counts and duration, and task completion rates. Subjective measures extracted from user's surveys, including perceived task success and ease of use measures, can be combined with the objective measures and the results used later for accuracy computation.

1. INTRODUCTION

One of the bottlenecks in the design and the deployment of spoken dialogue systems is conducting rapid and efficient performance evaluation. Spoken dialogue systems involve integration of several disparate technology components typically distributed over the network. In many situations there are different applications that are running in the same environment sharing resources that need to be evaluated. It is crucial to log and monitor data from the various dialogue system components in a centralized manner. For example in the DARPA communicator program, considerable effort was dedicated to specify the content and the XML format for logging high level dialogue metrics [11]. However, the infrastructure for monitoring and reporting system level activities together with details of logging data management were kept relatively undefined. We proposed a methodology and developed a set of web-based tools to facilitate and unify the evaluation process. In this paper we describe the details of our methodology using the example of a multi-modal spoken dialogue directory access application.

2. SYSTEM ARCHITECTURE

MILER was developed around an ECTF-compliant [7] Computer Telephony (CT) architecture, CT Media, which was extended to support IP-based telephony services and mixed-initiative spoken dialogue systems as well as multi-modal input/output [8]. Continuous speech recognition [5] and Text-To-Speech synthesis [9] are based on AT&T's Watson technology. A Generic Multi-modal Interface (GMMI) system is used to provide graphical user interface for presenting information to the user and collecting input stimuli [8]. An application independent middleware, called the Application Resource Manager (ARM), was developed to control both the I/O devices (telephony, multi-modal interfaces) and I/O resources (ASR, TTS, GUI managers). Dialog manager, spoken language understanding, back-end database and other services talk to I/O resources and devices through the ARM, which also instruments the service data logger. Collected data are immediately made available over the Intranet via several CGI scripts, which retrieve and summarize them into readable HTML reports. Anomalous system latencies, such as dialog manager processing time and barge-in occurrences are automatically detected and highlighted in different colors. Dialogue data, user and system turns are available for analysis, including the user's speech, which can be played within the browser.

CT Media gives the user access to services via both the Public Switched Telephone Network (PSTN) and IP by means of H.323 clients. It routes calls to specific services based on the ISDN PRI (Primary Rate Interface) DNIS (Dialed Number Identification Service) number. Multiple services can run at the same time on the same server. Applications are distributed in a client-server fashion. The system processes spoken, mouse, pen, keyboard and touch screen-based inputs. Output modes are audio only

for the voice telephony interfaces, and, audio, graphics, text, and video for PC and PDA interfaces [1]. Although complementary modalities are integrated, *mutual disambiguation* [6] is currently under development.

When a phone call comes in, either over IP or from the traditional PSTN, detailed information about the user's sessions are automatically logged by ARM in a plain text file from the active live services. Multi-modal inputs, including user speech (e.g. recorded utterance, speech detection and barge-in timing), pen taps, mouse clicks, and keyboard keystroke events are recorded with accurate timing information.

2.1 SYSTEM COMPONENTS

Primarily MILER consists of six modules divided into two categories: the data logger, which includes: (1) online data logger (ONDL) agent; (2) offline data logger (OFFDL) agent; (3) relational database management system (RDMS); and the presentation modules, which include: (4) experimental design wizard; (5) labeling module; (6) report generator. Figure 1 depicts the first three modules.

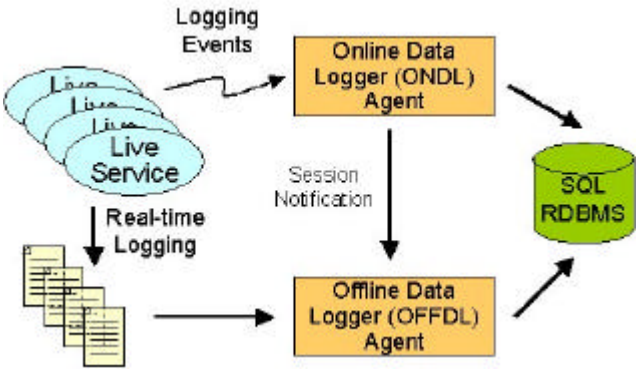

Figure 1. System Architecture

Online Data Logger Agent

The online data logger agent is a PERL program, which is listening for UDP/IP notification messages from the active services. It stores salient data into a relational database and communicates session information to the offline agent. A Java applet, located at a specific intranet URL address, can be loaded and executed within a web browser to periodically poll the database data and show the status of the active lines. Figure 2 shows a browser snapshot of the applet monitoring activities with three active lines and one idle line.

Services Status

CT Server	Service ID	Line ID	Line Status	ANI	DNIS	Date	Time
clio	CMTR	B0Trunk1PRI7	IDLE	9733608515	8026	2000-07-11	07:57:23.810
ursula	W99	B0Trunk1PRI	CALL	9733608999	8023	2000-07-11	08:28:10.150
thalia	mVPQ	B0Trunk1PRI20	CALL	9733608111	8019	2000-07-11	08:30:11.189
clio	CMTR	B0Trunk1PRI13	CALL	9733608110	8026	2000-07-11	08:31:27.691

Figure 2. Live services status

Offline Data Logger Agent

The offline data logger agent is also a PERL program, which is listening for UDP/IP notification messages from the ONDL agent. When a session is completed (e.g. when the user or the system hangs up), the offline agent, upon online agent's request, parses the text log file and populates a relational database with detailed login events (Figure 1). This two-pass approach is necessary to reduce the network traffic with the live services since several telephone lines could be active at the same time.

Relational Database Management System

Logging data is collected and stored in a Relational Database Management System (RDMS) that support SQL query language for data access and manipulation. The database schema is quite simple: events are stored in a table containing *timestamp*, *event type*, *event key* and *event value* fields along with other system information. Additional database tables store *session*, *labeling* and *tag schema* information. Basically, dialogue data are retrieved by selecting a session ID and accessing the event *type-key-value* tuple sorted by the *timestamp* field. Although this design solution gives the maximum flexibility for adding and removing fields, efficiency and data retrieval capabilities are penalized because most of the queries are based on the table fields' content instead of table attributes. Dialogue turns information is fragmented in several sequential record entries ordered by timestamps. The dialogue turn concept has to be reconstructed combining SQL query results and PERL script post-processing. Moreover, database temporal constrains are not well captured by this solution. For example, searching

for a specific event e_{t_i} followed by the event e_{t_j} cannot be expressed in SQL. A typical situation is: find all the dialogues **where** the system prompt **is** "*Were you able to complete your travel task successfully?*" (event `type=TTS`, `key=phrase`, `value="Where you …"`) **and is followed** by an affirmative user's answer (event `type=ASR`, `key=result_string`, `value="_YES(.*)_YES"`). These temporal constrains are resolved, again, combining SQL queries and PERL-based text processing. Future releases will improve the database design schema.

Experimental Design Wizard

In order to create an experimental design setup, MILER provides a web-based wizard that guides the researcher through three simple setup steps. Figure 3 shows the first wizard page where the user defines the experimental data set name and description.

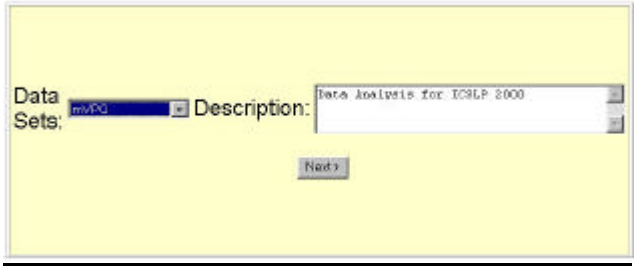

Figure 3. Experimental Design Wizard 1/3. Data Set Creation.

Next, a tagging schema is entered by tag name and description. In this phase, system and user turns tags are created along with other evaluation variables such user transcription, system and user concepts, word and concept system correctness and so on. Once the tag set is defined, labeling forms will be dynamically generated by each user/system turn pair. Figure 4 shows the tagging schema used for multi-modal directory access system described in [10].

Figure 5 shows the final wizard step where the dialogue sets are extracted from the collected data set. The selection criteria are a combination of queries on date/time and/or event values. A date range can be combined along with system/user dialogue turn constraints. The following predicates: *contains, doesn't contain, is, isn't, begins with, ends with*, are appropriate for event value filtering. The search form field allows regular expression pattern matching. Constraints between subsequent dialogue turns are also implemented.

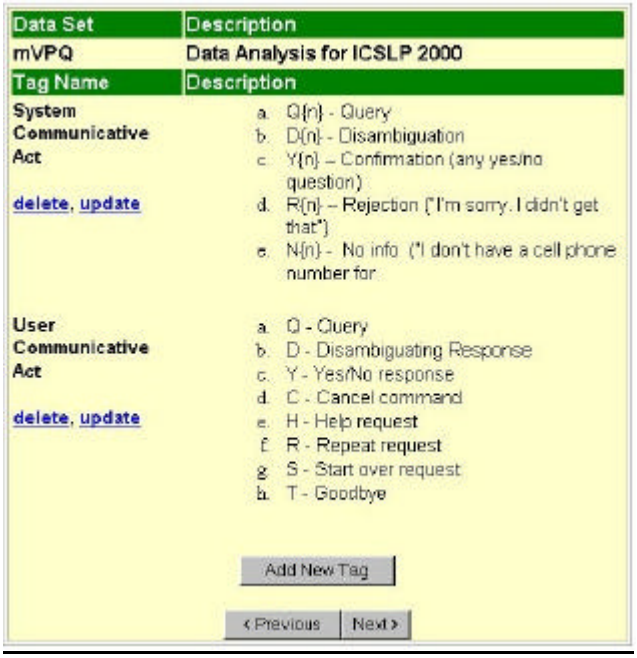

Figure 4. Experimental Design Wizard 2/3. Tags Definition.

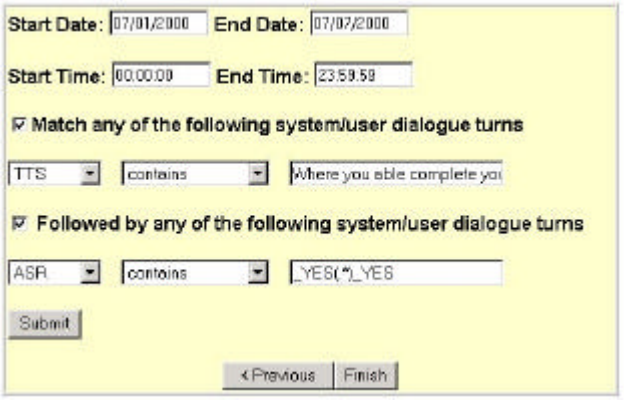

Figure 5. Experimental Design Wizard 3/3. Dialogue Set Selection.

Labeling Module

Based on the previously defined annotation schema, the labeling module generates an annotation web form for each dialogue turn. Turn by turn, the screen presents all the possible details of the interaction: the system output prompt, the message appearing on the screen (if the access device has a screen), the system recognized utterance, the system semantic representation (if available) and a hyperlink to the recorded speech utterance. The labeler can analyze, interpret, annotate and store the actual interaction for further analysis.

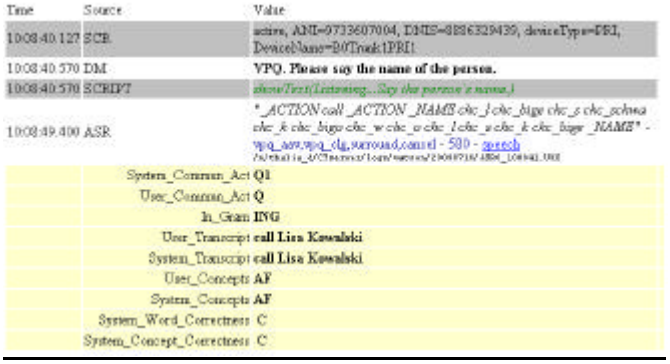

Figure 6. Example of Labeled Dialogue Turn

Report Generator

The report generator retrieves data from the database and produces HTML documents. The example in Figure 7 shows a dialogue for a multi-modal directory service where the user first requests the pager number for Cynthia Smith using speech input.

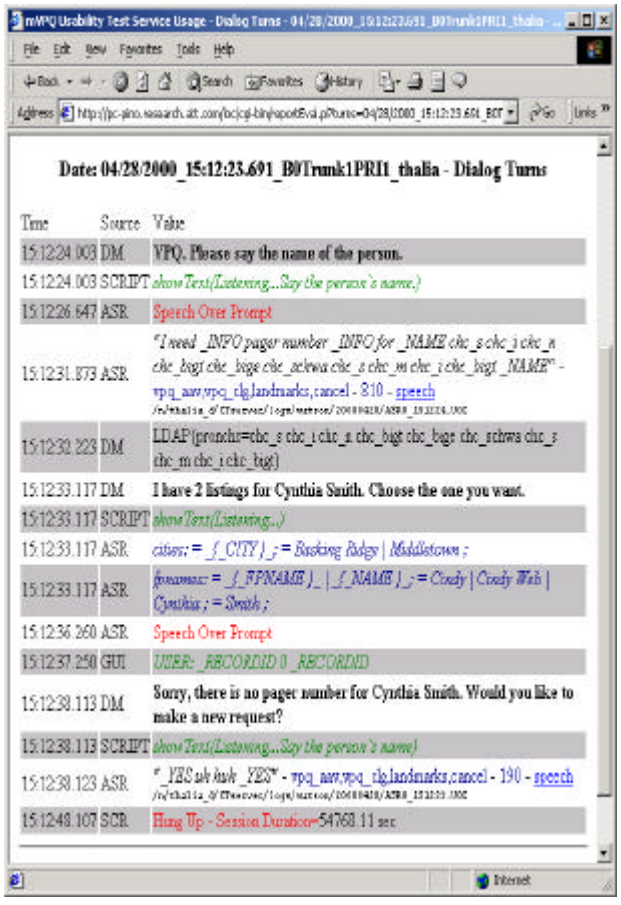

Figure 7. Dialogue Report Example

Then, in the second dialogue turn, she/he decides to select the first record that was displayed on the screen by touching the picture. The report also shows extra activities occurred during the interaction such as "speech-over-prompt" detection (i.e. user's barge-in), database look-ups and dynamic grammar compilation.

3. MULTI-MODAL DIALOGUE LABELING EXAMPLE

Table 1 shows an example of dialogue report generated by MILER after manual dialogue labeling. The application is a spoken dialogue system for accessing personnel directory information from a multi-modal interface (kiosk) [10]. The first table row reports the wizard pre-defined tags, respectively: user ID, task ID, system communicative action, user communicative action, speech recognizer grammar domain, user input transcription, system concepts in reaction to each user utterance, user expressed concepts, and concept correctness.

Usr ID	Task ID	Sys Act	Usr Act	ASR Grm	Usr Trans	Sys C	Usr C	Res C
SH	SM	Q1	Q	ING	Cynthia Smith	F	L	X
SH	SM	I1	S	ING	-	S-T	S-T	C
SH	SM	Q2	Q	ING	Cynthia Smith	F	F	C
SH	SM	D1	D	ING	-	W-T	W-T	C

Table 1. Labeled Dialogue Example

In the first interaction, the system plays the initial prompt (query tag Q1): *"VPQ. Say the name of the person"*. The user says an in-grammar (input tag ING) full name: *"Cynthia Smith"* (concept tag F), but the system misunderstands and responds with an incorrect last name (concept tag L). The concept is misunderstood too (concept tag X). In the second turn, the system is playing and showing the wrong information (I1) back to the user, but the user touches the *start over* button on the screen (S-T) to reformulate the request. The perceived concept is correct (C) and the system proposes the second time the initial prompt (Q2). The user repeats the full name (F), the response is correct (C) and the screen shows two different options for Cynthia Smith in two different locations. The user finally disambiguates (D) the request touching the city button on the screen (W-T).

This labeled data are quickly generated from the web dialogue reports. The compact tag notation can be passed to a scoring program for word and concept accuracy evaluation.

4. SUMMARY

The described tool is a flexible and efficient system to monitor and evaluate multi-modal conversational systems. It is platform independent, distributed and it has been used to support several diverse services such as [3], [4] and the AT&T DARPA Communicator [12]. In contrast with other approach [2], MILER is service and platform independent and allows easy plug and play of Natural Language Understanding and Multi-modal Parser modules for evaluation. The DARPA communicator hub architecture [11] has a centralized approach similar to ARM, including logging. However, ARM has been designed to support a more flexible Computer Telephony environment for multi-channels, multi-modal applications. Detailed system's events logging (e.g. barge-in, resource latency detection, etc.) are better captured by ARM, rather than propagated to the hub and synchronized with the high-level dialogue events.

5. REFERENCES

[1] G. Di Fabbrizio, S. Narayanan, P. Ruscitti, C. Kamm, B. Buntschuh, J. Hubbell, J. Wright, and J. Hamaker, *"Unifying conversational multimedia interfaces for accessing network services across communication devices"*, IEEE International Conference on Multimedia and Expo, New York City, New York, USA, July 30 - August 2, 2000

[2] J. Clow and S. L. Oviatt, *"Stamp: A Suite of Tools for Analyzing Multimodal System Processing"*, Proc. ICSLP 98, 1998.

[3] B. Buntschuh, C. Kamm, G. Di Fabbrizio, A. Abella, M. Mohri, S. Narayan, I. Zeljkovic, R. Sharp, J. Wright, S. Marcus, J. Shaffer, R. Duncan, and J. G. Wilpon. - *"VPQ: A Spoken Language Interface to Large Scale Directory Information"*, Proc. ICSLP 98, 1998.

[4] M. Rahim, R. Pieraccini, W. Eckert, E. Levin, G. Di Fabbrizio, G. Riccardi, C. Lin, C. Kamm *"W99 - A Spoken Dialogue System For The ASRU' 99 Workshop"*, ASRU99 Workshop, 1999

[5] R. D. Sharp, E. Bocchieri, C. Castillo, S. Parthasarathy, C. Rath, M. Riley, and J. Rowland, *"The Watson speech recognition engine"*, Proc. ICASSP 97, 4065-4068, 1997.

[6] S. Oviatt, *"Mutual Disambiguation of Recognition Errors in a Multimodal Architecture"*, Proceeding of the CHI 99 conference on Human factors in computing systems, May 15-20, 1999, Pittsburgh, PA USA

[7] *"ECTF Architecture Framework"* - Revision 1.0, http://www.ectf.org/ectf/pubdocs/arch_fr.pdf

[8] G. Di Fabbrizio, C. Kamm , P. Ruscitti, S. Narayanan, B. Buntschuh, A. Abella, J. Hubbell and J. Wright, *"Extending a Standard-based IP and Computer Telephony Platform to Support Multi-modal Services"*, Proc. of Interactive Dialogue in Multimodal systems, Kloster-Irsee, Germany, pp.9-12, Jun. 1999

[9] M. Beutnagel, A. Conkie, J. Schroeter, Y. Styliano and A. Syrdal, *"The AT&T Next Gen TTS System."* Proc. Joint Mtg. ASA, EAA and DEGA, Berlin, 1999

[10] S. Narayanan, G. Di Fabbrizio, C. Kamm, J. Hubbell, B. Buntschuh, P. Ruscitti, J. Wright – *"Effects of Dialog Initiative and Multimodal Presentation Strategies on Large Directory Information Access"*, to appear in 6th International Conference on Spoken Language Processing (ICSLP2000/ INTERSPEECH 2000), Beijing, China, October 16 –20, 2000

[11] DARPA Communicator Testbed - Log Standard Proposal (v14) - http://fofoca.mitre.org/logstandard/log-standard-v14.html

[12] E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, M. Walker, *"The At&T-Darpa Communicator Mixed-Initiative Spoken Dialog System"*, to appear in 6th International Conference on Spoken Language Processing (ICSLP2000/ INTERSPEECH 2000), Beijing, China, October 16 –20, 2000