DATABASE MANAGEMENT AND ANALYSIS FOR SPOKEN DIALOG SYSTEMS: METHODOLOGY AND TOOLS

Chih-mei Lin, Shrikanth Narayanan, Russell Ritenour

AT&T Labs – Research
180 Park Avenue, Florham Park, NJ 07932, USA
email: {cmlin,shri,rit}@research.att.com

ABSTRACT

A methodology for creating and managing an integrated database for spoken dialog systems is proposed. Using an example of a telecommunication service application, details of organizing, maintaining, and visualizing the dialog system data are presented. Examples illustrating the use of the unified database structure for dialog reproduction and performance evaluation are provided.

1. MOTIVATION

Recent advances in automatic speech recognition technology and natural language processing have led to a spurt in the development of human-machine dialog (*agent*) systems for a variety of applications ranging from telecommunication services to remote information access and automated business transactions [1, 2, 3]. The spoken dialog system technology is, however, still far from maturity with several open research issues in areas such as system evaluation and validation [4], dialog management, and user adaptation strategies. To study these issues effectively and to evaluate the agent's performance, it is crucial to keep track of data from a variety of sources and in disparate forms[1] (such as speech data, user profile data, agent interaction log, system access data etc), in a manner that is readily accessible and/or usable (for example, off-line dialog scenario reproduction). Handling tremendous amounts of data from various sources and multiple users requires intelligent database organization and toolkits for database creation, management, manipulation and visualization. Links back to the agent to enable (off-line) dialog reproduction or recreation, and modifying the agent's behavior. Such database systems are also important in enabling an efficient shared research environment on spoken dialog agents.

In this paper, a methodology for managing and analyzing dialog system databases is presented with reference to a specific dialog system developed for a telecommunication service application.

2. DATABASE REQUIREMENTS

A spoken dialog agent system typically comprises four major components: speech recognizer, natural language processing module, dialog manager, and response generation unit. Although primarily voice activated, the agent may handle visual and/or text inputs/outputs. In addition to data from these agent modules, data from sources such as call agent access history (system access type, mode, frequency) and user surveys need to be organized and linked appropriately. Typically, all these data are generally in different physical locations and a systematic evaluation of the agent requires a centralized database access/control structure that links all the data together. Since the agent evolves dynamically, so should the database that tracks the agent.

In this section, some of the potential uses for the agent data are enumerated with a view toward defining the requirements for the database and toolkit design.

1. *Performance evaluation and validation:* This includes measurement of literal and semantic speech recognition accuracy, task completion success, system and dialog errors/repairs/recovery, and correlating user perception with system performance.
2. *Testing:* Enable off-line experimentation under new parameters and conditions and testing of new or modified features (dialog re-creation).
3. *Algorithm and system design:* Improved acoustic modeling and adaptation, domain-specific language modeling, user-dependent grammar and feature adaptation (dialog reproduction as a part of history tracking).
4. *Agent behavior learning:* Adaptive selection of dialog strategies.

Some of the agent data may be directly imported to the database (for example, system response logs), while others need intermediate data processing. For example, manual processing is required for speech data transcription and dialog tagging. Speech transcription , however, needs to be facilitated by information from the agent such as valid grammars and vocabulary that were active at the time of the interaction (note that grammars, especially the speaker-dependent ones, may evolve dynamically). In the next section, the organization of the data from the various sources is described.

3. DATABASE ORGANIZATION

A mixed-initiative, user-configurable personal agent that provides telephone call control and messaging services is considered in this paper. The database requirements for such a system are more complex than for a non-user configurable agent since dynamically-evolving speaker-dependent vocabulary and other feature selections need to be tracked. The data collection overview is shown in Figure 1. A multiple client-server architecture is used and speech data is collected from the ASR servers while the session logs and user profile data are recorded at the application level. Links from the ASR servers to the application's session log allow the system to maintain continuity between the user's speech data and the session it is recorded.

A relational database design mechanism is used to support the spoken dialog system data. The goal of a relational database design is to generate a set of relation schemes that allow people to store information without unnecessary redundancy and to retrieve information easily. The entity-relationship (E-R) data model is based on a perception of a real world which consists of a set of basic objects, entities, and relationships among these objects.

The six sub-databases that make up the bulk of the dialog system data, demonstrated by classes and instances diagrams in Figure 2 and Figure 3 are: (1) *User Profile Database* (2) *User Speech Database* (3) *Session Database* (4) *System Prompts Database* (5) *Dialog Database* (6) *System Logging Database*.

In addition, the database system may include results of subjective system testing such as user surveys (tagged by account and time stamp information).

3.1. USER PROFILE DATABASE

The *User Profile* database keeps track of user related changes in the system configuration and feature settings, and maintains a historical record of these changes. This database contains user preferences for system behavior (e.g. the number of rings before answering an inbound call), the user-configured personal dialing list information (enrolled either through an internet text interface or a voice interface; it contains standard rolodex information, speech data for names enrolled by voice, alias options associated with each name in

[1] Although many of the current applications are designed for implementation over the telephone network and are primarily restricted to voice access, several new agent systems currently being developed integrate visual, voice and text inputs, thereby demanding that the agent database system handle multimedia data.

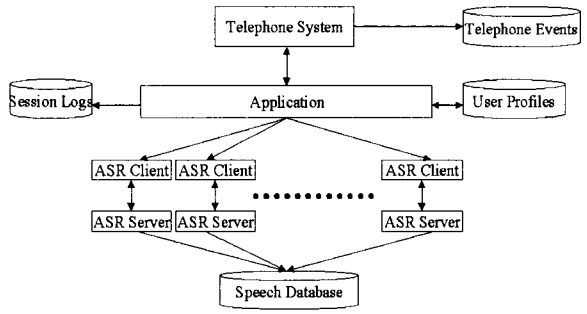

Figure 1: Database collection scheme for a user-configurable telephony agent.

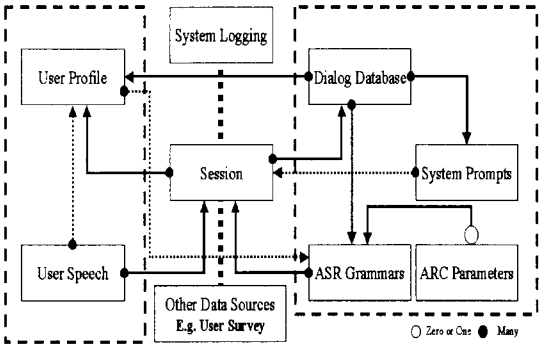

Figure 2: Database Class Diagram showing the six major data sources.

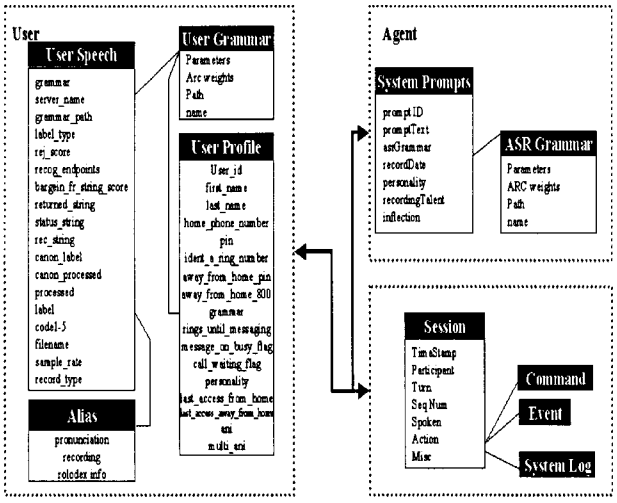

Figure 3: Database Instances Diagram focussing on the *Session*, *User Profile*, *User Speech*, and the *System Prompts* databases. Further details of the sessions and dialog databases are shown in Figures 4 and 5, respectively.

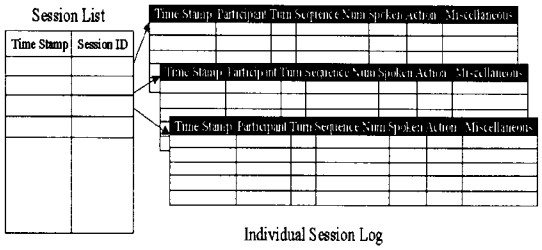

Figure 4: Session Database showing the fields that get populated during every agent-user interaction.

the list etc.), unique grammars for ASR that are modified based on configuration changes, usage profile, and subscription information (Figure 3). Each of the user's grammars are maintained in a source code control system so that the user's ASR grammars for any usage period can be exactly recreated.

3.2. USER SPEECH DATABASE

The *User Speech* database contains the speech data and all the results from the speech recognizer for every recognition attempt (Figure 3). Entries such as top recognition candidates and likelihood scores, vocabulary and grammar, unique user identification tag, system parameter settings (end pointer, rejection thresholds and scores), whether the user barged-in over the agent's prompt etc., are all stored in terms of a formatted header directly at the time of the interaction. Typically, the raw speech data, or some subset, are manually transcribed and any other unusual characteristics of the speech and/or the background (for example, background noise, details of out of vocabulary utterances) are recorded in the header to complete the speech data information. The header information is finally imported to the database with each speech file representing a record in the database, tagged uniquely by the (Unix system) time stamp of the interaction and the server name. Each line of the header populates a field in each database record. A special graphical user interface (GUI) toolkit was developed to facilitate the dialog speech transcription: Since different grammars are active along different points during the user's interaction with the agent, and since the contents of these grammars are change with time, the transcription tool was designed to dynamically link the appropriate vocabulary version pertaining to the account at the time of the interaction. User-configured vocabularies are constructed from the enrollment utterances (text and/or voice) in the *User Profile* database while system commands are generated from the call flow specification for the ASR grammar network in the *Dialog* database.

3.3. SESSION DATABASE

The *Session* database provides a time-sequence of call events typically generated from a formatted session log that provides a step-by-step trace through the dialog with the agent, with at least a 10 msec resolution of time stamping. This log embodies the details of all user queries, agent responses and successful/failed actions. Information is extracted from the session log and used to populate the fields of *Session* database. The *Session* database provides a complete picture of the agent's states and actions and when linked appropriately with the *User Speech* and *System Prompts* databases provides a complete reproduction of the dialog (See Figure 4). The link to the *User Speech* database is provided through unique file tags that use the (Unix system) time stamp of the interaction and the server name. The link to the *System Prompts* database is provided by unique prompt identification tags.

Using this database, we can completely reproduce any particular agent-user interaction (including an exact audio transcription). The database can be queried under several complex conditions, some of which will be illustrated in Section 6.

3.4. SYSTEM PROMPTS DATABASE

The *System Prompts* database maintains a complete listing of all the prompts that are available to the system. These prompts are indexed by a unique

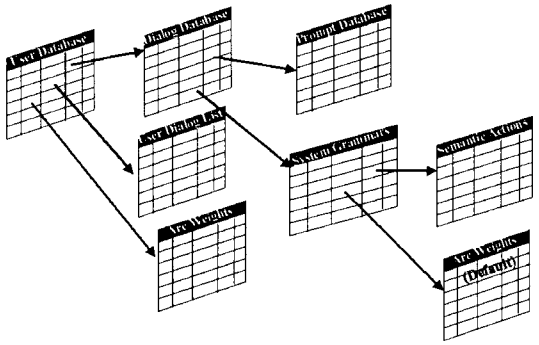

Figure 5: Dialog Database keeps details of the system grammars and user interaction specifications.

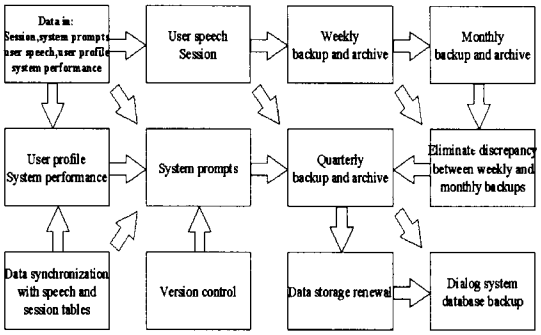

Figure 6: Database backup flow diagram

identifying number, the text of the prompt, version numbering, comments (such as inflection, stresses, prosody), speaker/personality, ASR grammars that may reference the prompt, and recording time/date. Also included are coding information, file name and location, and sampling rate.

3.5. DIALOG DATABASE

The dialog manager controls the interactions with the user. It uses a template of desired information (specified by the application developer) to guide its queries. The *Dialog* database is analogous to the *User Profile* database; it maintains the configuration information for the agent. Once the user has been identified, the appropriate dialogs are retrieved from the *Dialog* database.

Once the current context and the user identification are known, the *Dialog* database provides the indices into the *System Prompts* database for the sequence of prompts to be played, an index into the system's ASR grammars for the ASR engine, and a reference to the semantic grammar for understanding the user's response (Fig. 5). The system's ASR grammars are combined with the user's individual dialing lists to create a custom ASR grammar that incorporates the user-configured items. In addition to the default settings, the database can store optional ASR grammar network arc weights on a per-account basis for all the system grammars based on the *User Profile*.

In addition to the typical agent/user interaction, the *Dialog* database maintains transition information to guide the system to the next interaction. The dialog database also maintains a list of possible actions that can be taken and the transitions to/from those actions. Thus, the dialog database maintains the connections between user interactions (prompts, prompts + recording, or prompts + recognition) and system actions.

3.6. SYSTEM LOGGING DATABASE

Finally, the system maintains a *System Logging* database that contains information from the telephone monitoring program (the details about telephony events, message set-up and their time-stamps), the ASR client/server communication, and the Unix activity levels.

4. DATABASE MANAGEMENT

Each of the component databases change at a different rate. In this section, details of the spoken dialog system database management are discussed; emphasizing the disparate characteristics of the component databases. While the *System Prompts*, *Dialog*, and the *User Profile* databases change and grow slowly, the *Session* database and the *User Speech* database grow at a tremendous rate with ongoing usage by multiple users. On the other hand the information in the *User Profile*, *System Prompts* and *Dialog* databases have more 'permanence' in the value of the information they bear when compared to that provided by the *Session* and *User Speech* databases. Hence, different strategies need to be adopted for data archival purposes. Furthermore, from a data analysis point of view, one needs to ensure synchronization between the contents of the different component databases.

The *Session* and *User Speech* databases demand efficient storage and retrieval. Weekly updates and weekly, monthly and periodic backups are essential to ensure data continuity. Version controls are used to determine which database chunks reside in the currently active structure. A flowchart describing the various stages involved in the database management is given in Figure. 6.

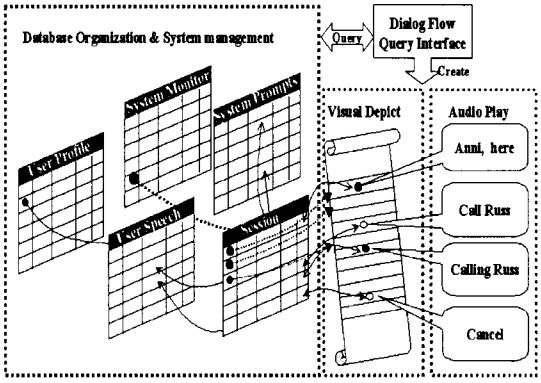

Figure 7: Database visualization scheme showing an audio and visual reproduction of an agent-user dialog.

The mechanism used to provide the most frequently used entries is the idea of data caching. The latest data is always kept in the database cache. When data remains untouched for one week, the system automatically archives and backs up the data onto the storage media, including digital data tapes and hard disks. After eliminating the discrepancy between data each month, data is stored onto permanent media, like CDroms. Quarterly backups are made for all of the databases whether they are recently changed, including the *User Speech* and *Session* databases, or remain unchanged for a long period of time, including the *User Profile* and *System Prompts* databases.

5. DATABASE VISUALIZATION

There are two main needs for agent data visualization: one is to reproduce the original dialog and the other is to recreate the dialog under new conditions and/or different parameter settings (Fig. 7). Because the database organization is built on the entity-relationship (ER) model, individual tables can be easily linked based on the predefined relationships. For example, time sequence information is used to track the user's speech in the interaction by linking the *Session* database with the *User Speech* database. The prompt file name is used as the linking tag connecting the *Session* and the *System Prompts* databases. Hence, when the call flow is reproduced, the grammars and vocabularies are dynamically linked to it (through the *User Speech* and *User Profile* databases). Graphical representation of the dialog paths is embedded in the database visualization scheme. Either the whole or parts of an interaction can be graphically visualized. An ensemble of interactions from a user over a period of time can be obtained in a straightforward manner and may be useful in designing long-term user adaptation techniques (an open research issue). An audio reproduction of the dialogs is also available. Dialog recreation (facilitated by a GUI) can be accomplished by reproducing the call flow but under new recognizer/system parameters (e.g. new rejection thresholds and timeout settings).

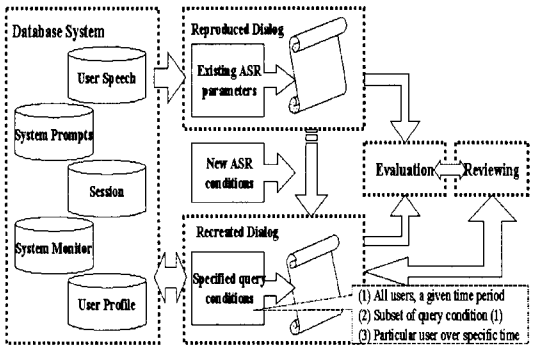

Figure 8: Database Organization: Example for recognition application.

6. DATABASE APPLICATIONS: EXAMPLES

Recall that the unified agent database comprises 6 major components. This section illustrates how the contents of these sub-databases can be easily accessed, brought together, and used in an integrated fashion for performing various tasks such as performance evaluation and algorithm and user interface design. Our GUI query toolkit provides a friendly environment to access, manipulate and analyze data, objects, and results.

6.1. REVIEWING THE DIALOG: AUDIO AND TEXT REPRODUCTIONS

As described in Section 5, the *Session* database, in conjunction with the *User Speech* and *System Prompts* databases, is used to regenerate an audio replay of any session (Fig. 7). This allows us to review the call flow as it happened and listen to the entire interaction. This helps the system designer to study the timing pattern of the interaction and optimize the user interface.

The *Dialog* database allows us to view the sum of all potential dialog paths a user may take through the system (recall that the underlying speech recognition is constrained by a finite state grammar network). However, we are also interested in analyzing specific instances of actual interaction between the agent and the user, most probably under some pre-specified condition (Sec. 5). This requires data from more than one component database. Some likely scenarios of requiring the dialog path reproduction are: (1) a composite summary for all users (or, a specified subset of users) for a given task (say, after first agent prompt following successful access to the system), (2) a summary of the entire interaction for a particular user over a specified period of time, (3) a summary for a sequence of events, for example, placing a call followed by canceling the call, and (4) a summary of user actions after system rejections over a specified period of time.

Using the *Session* database, with its link into the *System Prompt* and *User Speech* databases in conjunction with the *Dialog* database, a network of actual dialog paths can be created. For developing continuous adaptation and customization of the the ASR grammars for each user, the availability of such dialog recreation enables us to design stochastic grammars based on ongoing system usage. One way of achieving this is to maintain account-specific weighting of the grammar network arcs and store the weighted arcs in the *User Profile* database.

Using the combination of the *Session*, *User Profile* and *User Speech* databases, the effectiveness of individual prompts can be examined and parts of the interaction where users may be getting confused can be identified. Correlating the time spent within a particular task in a session with the transcribed user speech data and looking for out-of-vocabulary utterances, along with requests for help, will indicate that users are confused at a particular point in the dialog.

6.2. ASR ANALYSIS AND ALGORITHM DESIGN

The availability of the unified database provides an increased scope for analyzing the ASR performance. For example, obtaining performance results for the agent's performance under query conditions such as specific interaction type, set of users, a specific access type, speech category (whether in vocabulary or out of vocabulary), and the time period of the dialog, becomes trivial. Performance results can be obtained for a sub-dialog or an entire interaction. A GUI query toolkit is used to facilitate such analyses.

Realistic speech recognition experiments can be run with new parameters using data from the agent database, and the results can be linked back to the database for enabling comparative analysis with original dialog. An overview of one such database organization for recognition performance analysis is shown in Fig. 8. This allows the system designer to optimize component parts, such as the recognizer and rejection performance, at the *system level.*

The database also allows us to easily specify data sets to build new acoustic models or refine existing ones, for example on a per-account basis or per-grammar basis.

6.3. REDUCING HUMAN INTERVENTION

Since there is a large amount of speech data collected during a service trial (approximately 15 MB/day), and since ongoing performance needs to be measured, we need to prioritize our manual speech transcription efforts. One method to accomplish this is to use the session database to identify problems with the interaction and automatically generate a list of the corresponding speech files to be manually transcribed for further probing. Note that the manual transcription is facilitated by vocabulary information automatically obtained from the relevant *User profile* and *Dialog database*.

6.4. AUTOMATIC REPORT GENERATION

With the availability of an integrated database, a comprehensive report of the ongoing performance of the spoken dialog system can be automatically generated, archived and distributed to the system developers. Example items in such a report are ASR acceptance/rejection numbers, statistics of duration of each interaction, details of calls placed, access records (successful/failed), system/server failure details etc. Flags can be raised to alert the system developers for any deviations from expected normal system activity.

7. SUMMARY

Testing and evaluation of spoken dialog systems is a daunting task that demands efficient integration of data from several disparate sources and forms. Furthermore, a unified database structure is essential for furthering our understanding of the human-machine dialogs and developing automated strategies for user adaptation of spoken dialog agents. Toward achieving such goals, a methodology for an efficient database integration and organization for a spoken dialog system was presented based on a case study of an agent providing telecommunications assistance. Toolkits were designed to facilitate analysis and testing of the dialog system in a systematic manner with the use of the unified database structure.

8. ACKNOWLEDGEMENTS

The authors are grateful to the members of AT&T's team on voice-enabled agents.

9. REFERENCES

[1] C. Kamm, S. Narayanan, D. Dutton, and R. Ritenour, "Evaluating spoken dialog systems for telecommunication services," in *Proc. of Eurospeech97*, (Rhodes, Greece), 1997.

[2] M. D. Sadek, A. Ferrieux, A. Cozannet, P. Bretier, F. Panaget, and J. Simonin, "Effective Human-Computer cooperative spoken dialogue: The AGS demonstrator," in *Proc. of the Intnl Conf. Spoken Lang. Processing*, vol. 1, (Philadelphia, PA), pp. 546–549, 1996.

[3] S. Bennacef, L. Devillers, S. Rosset, and L. Lamel, "Dialog in the RAILTEL telephone-based system," in *Proc. of the Intnl Conf. Spoken Lang. Processing*, vol. 1, (Philadelphia, PA), pp. 550–553, 1996.

[4] M. Walker, D. Litman, C. Kamm, and A. Abella, "PARADISE: A framework for evaluating spoken dialogue agents," in *Proceedings of ACL/EACL97*, 1997.