Proc. of EuroSpeech, Budapest, Hungary, Sep. 1999.

MULTIMODAL SYSTEMS FOR CHILDREN: BUILDING A PROTOTYPE

Shrikanth Narayanan, Alexandros Potamianos[†] *and Haohong Wang*

AT&T Labs–Research, 180 Park Ave, P.O. Box 971, Florham Park, NJ 07932-0971, U.S.A.

shri@research.att.com, potam@research.bell-labs.com

ABSTRACT

In this paper, we describe our efforts in designing and building a prototype multimodal system for children users. Data collection efforts and user experience results from a WoZ study using a popular computer game are reviewed first. Automatic speech recognition and spoken language understanding technology for children speakers are discussed next. A multimodal prototype is designed for a personal agent and a gaming application. Emphasis is placed on a modular architecture, handling of multimodal input and multimedia output, and providing an engaging user interface. Informal evaluation by children users was very positive especially for the animated agent and the speech interface.

1. INTRODUCTION

The CHildren's Interactive Multimedia Project (Agent CHIMP) aims at providing design guidelines for building successful multimodal-input multimedia-output applications for children users. A special emphasis is placed on the spoken dialog interface. There are many factors that motivated and served as axioms for this study: (1) Children form a crucial market segment for interactive multimedia systems. (2) Children are eager and quick to embrace new technologies. (3) The dynamics of man-machine interactions are different for children compared to adults. For example, children's speech exhibits greater degree of acoustic and linguistic variability and is highly age-dependent [2]. The problem solving skills and strategies also differ widely with age.

Prototypes of spoken dialog systems for children can be found in the literature [3, 11]. There are also some commercial products (toys and computer games) for children with limited speech recognition capabilities (small vocabulary, isolated word recognition). Building a conversational natural language interface, especially for children, is a challenging problem and needs to be carried out in several stages. First, a proof of concept needs to be established for using voice as a viable means of children interacting with a machine, in terms of both its feasibility and usability. Second, data from children need to be collected for quantifying the variability present in their speech and to train and test models for automatic speech recognition (ASR). This is necessary for ensuring acceptable levels of ASR and understanding performance across all ages and environments. The design of the prototype system follows based on the results and intuition gained from analysis and modeling of the collected data [6].

An important feature of our system involves the integration of multiple input and output modalities such as voice, audio, keyboard, mouse, graphics and animation. Recently there has been increasing interest in the design of multimodal applications that combine and utilize a variety of input modalities such as speech, handwriting, gestures and touch [10, 12]. Results of these investigations, although to a large extent preliminary, suggest that the use of multiple modalities leads to more efficient and natural communication and enhances user experience (for example, [1]). Moreover, the actual type of device involved – desktop, telephones, PDAs, wearable computing devices – plays a crucial role in dictating the mix of output presentation modalities that should be provided to the user. There are numerous open research issues in terms of input data integration and interpretation, multimodal dialog design, output presentation strategies and performance evaluation that need to be addressed. Realistic case studies and prototype designs are crucial to further our understanding of multimodal interactions and the study presented in this paper represents an effort in this direction.

The organization of the paper is as follows. First, an overview of our work in [6] is presented: the collection and analysis of acoustic, spoken language and dialog data from a multimodal (speech, keyboard and mouse) Wizard of Oz (WoZ) experiment. Results from a user experience study are also presented and compared for different input modalities. Age-dependent and modality-dependent trends in the application are computed by analyzing and modeling the (generalized) dialog flow. In addition, the effect of speech recognition errors/rejections is discussed in relation to the user interface design. A brief overview of the state-of-the-art in speech recognition and understanding performance for children is presented next. Using the knowledge obtained from the WoZ experiments, a prototype system is designed. The agent CHIMP prototype combines speech, keyboard and mouse input modalities, and uses text, graphics, speech and animation for presentation. The application is controlled by animated agents. We conclude with an informal user evaluation of the prototype.

2. WOZ EXPERIMENTS

Acoustic, linguistic and dialog data were collected and labeled in a Wizard of Oz experiment from 160 children, ages 8-14 playing an interactive computer game ("Where in the U.S.A. is Carmen Sandiego" by Broderbund Software) using voice commands, or keyboard and mouse [6]. To successfully complete the game, i.e., arrest the appropriate suspect, two subtasks had to be completed, namely, determine the physical characteristics of the suspect to enable an arrest warrant and track the suspect's whereabouts (in one of fifty states in U.S.A). The player could talk to characters on the game screen seeking clues that can be correlated with information in a geographical database. Information could be obtained from the database either by single or multiple word search. The player had to travel through five states in the U.S. tracking the suspect, before he could identify him (using the profile information) from among several cartoon characters on the screen.

Overall, the game is rich in dialog subtasks including navigation and multiple queries, database entry, and database

[†]Alexandros Potamianos is currently with Bell Labs, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974-0636, U.S.A.

search. Further, the fact that (during a substantial part of the game) the child conversed with cartoon characters on the screen made the dialog more natural and human-like. As a result spontaneous speech could be elicited.[1]

The structure of the game was not changed (no adaptation to voice modality). The only modification was to provide some degree of automation (by the wizard) when navigating in the top level and scrolling in the database, and the addition of four text-to-speech generated dialog error control/clarification messages.

2.1. Subjective user evaluation

All the children who took part in the WoZ experiment participated in an exit interview wherein subjective impressions about the game and the interface were elicited. Sample questions that they were asked included: (1) What did you like about using voice activation? (2) What did you like/dislike about the game? (3) Would you like to use voice with keyboard and mouse? The participants were also asked to rate on a scale from 1 to 5 (5 being the highest) the following: voice interface, game, use of headset, TTS-generated error messages, multimodal inputs.

The children gave very high marks to the speech interface (93% rated the interface 4 or 5). The game also received high marks but somewhat lower than the interface (only 81% rated the game 4 or 5). The speech interface ratings degraded only slightly when 5% misrecognitions and 5% rejections were randomly introduced into the game by the wizard. It is interesting to compare the relation between the number of games won and the ratings. Losing a game had a significant negative effect on the rating of the game. However, the outcome of the game did not seem to affect the children's rating on the voice input in the same fashion. The 11-12 year olds gave the highest ratings for the voice input. Gender effects were negligible.

Other results showed that the dislike for TTS generated error messages and for spelling (for the purpose of "ASR ambiguity resolution") decreased with age. The enjoyment of the use of a headset microphone roughly correlated with the enjoyment of the game. Finally, about two-thirds of the children preferred having a multimodal interface to a voice-only interface.

In summary, user experience results were promising for the inclusion of voice as one of the interaction modalities in the design of interactive applications for children.

2.2. Dialog data analysis

Speech utterances were manually assigned to dialog states according to the game actions they triggered [6]. Age and speaker dependencies in dialog state transitions were analyzed. Some key observations regarding spoken interactions included: (1) queries seeking multiple attributes were far less common than those seeking a single attribute (2) frequency of skipping states in the canonical game structure was low (3) frequency of superfluous commands such as "goodbye" was relatively high. While there were no noticeable differences in the dialog patterns of male and female children, the older children (11-14 years) tended to complete the game faster, did fewer database lookups, used more advanced dialog patterns, and had fewer out-of-domain utterances.

Comparison of voice versus keyboard-mouse modalities showed similar dialog and problem solving strategies and roughly the same number of commands for navigation and database entry tasks. However, children took fewer turns using keyboard and mouse than voice to carry out the relatively high-perplexity database search and retrieval tasks. Finally, there were fewer superfluous commands such as greetings in the keyboard and mouse mode.

[1] The children players were not informed of the existence of a wizard and an observation room. Further, for approximately half of the experimental runs the player was alone in the game room.

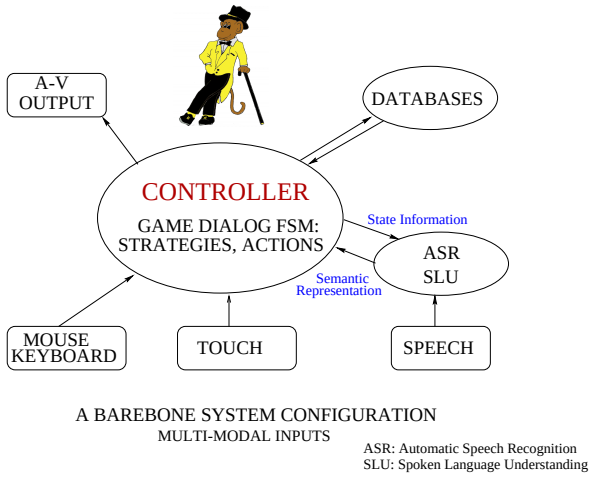

Figure 1: Functional block diagram of CHIMP prototype.

2.3. Performance of ASR and SLU

Due to inherent variabilities in children's speech [2], acoustic modeling for automatic recognition of children's speech is a challenging task. Word recognition accuracy as a function of age was reported in [7] wherein it was shown that there was a systematic degradation in ASR performance with decreasing age for a given model training condition. It was also demonstrated that using children-specific acoustic models and adaptation schemes such as vocal tract normalization and MLLR helps improve overall recognition rates.

In [8, 9], speech understanding in a maximum likelihood probabilistic framework was proposed and applied to the "Carmen Sandiego" task. Improved language and dialog modeling greatly improved spoken language understanding (SLU) performance: a 20% understanding error-rate reduction was shown in [9] by incorporating dialog state dependencies in the language models. Explicit dialog modeling (modeling of user intent) also reduced error rates up to 25% in [8]. Finally, an additional 10% error-rate reduction in SLU was achieved by utilizing acoustic confidence scores in the understanding model. Overall, about 90% dialog action understanding accuracy was achieved for this task by combining the aforementioned SLU algorithms. The reported results show that it is feasible to build successful spoken dialog systems for children using existing ASR/SLU technology.

3. BUILDING A PROTOTYPE

A communications assistant (telephony, web access, email) and computer game application scenario was used as a vehicle to achieve the following goals:

- Define a general multimodal system architecture.
- Provide means to investigate merging of multimodal inputs (keyboard text, mouse clicks, voice) and multimedia presentation strategies.
- Demonstrate the concept of agent and sub-agents that handle different application modules.
- Demonstrate the role of intelligence and personality of the user interface through spontaneous conversation, audio, animation (gestures), and graphics.

3.1. Dialog system building blocks

Figure 1 shows the main functional building blocks of the system from a user's perspective. The central part of the

system is the controller. The user interface enables interactions using voice, typed text, mouse clicks or combinations there of. Output to the user is presented through audio, graphics, animation and text modalities. The speech and language processing unit comprises of ASR and SLU components that enable spoken language interactions. The dialog manager also communicates with information sources such as databases. Further details of the various modules are given in the following sections.

The prototype system consists of the following components: input/output (I/O) event handler, dialog manager, graphical user interface (GUI), spoken language understanding (SLU), speech recognizer (ASR), speech synthesizer (TTS), animator and database. The speech recognizer uses children-specific acoustic models that were built using the data obtained from the WoZ study described in Sec. 2. Application specific finite state grammars were hand crafted to boot-strap the language models for the prototype applications (described later). ASR is performed using the AT&T Watson speech recognition engine.

The dialog manager defines the strategies and actions to be taken based on the user's input and decides what to present to the user. The dialog manager used in the CHIMP prototype is based on AMICA (AT&T's Mixed Initiative Conversational Architecture) which provides a library of dialogue actions and a means for specifying dialog strategies either through a JAVA-based GUI or through a high-level scripting language [5]. The 'template' is a data structure that AMICA uses to maintain the dialog state information. The dialog manager can communicate with external modules such as ASR and databases through socket connections.

Language understanding is based on the CHRONUS system which uses a conceptual decoding scheme to derive meaning from an utterance [4]. The template structure that provides an 'attribute-value' mapping of concepts for use by AMICA was derived using a template generator. The template generator is based on a finite state machine (FSM) implementation wherein a concept is associated with one or more FSMs and a phrase that belongs to a concept is processed by the corresponding FSMs. For simplicity and consistency sake, all actions underlying button clicks on the GUI are also mapped to equivalent natural language expressions and are henceforth handled by SLU in a way similar to spoken or typed inputs.

The GUI consists of five main display areas: graphics/animation area, text area, buttons area, command line area and user command echo area. The GUI design aimed to provide a consistent look and feel across various applications. The personality and appearance of the animated agents provide orientation for the user. Function buttons provide an alternate means of accomplishing several key commands that could also be achieved through voice or typed inputs. A history of user inputs is maintained and any previous input can be easily repeated by highlighting and clicking on the desired entry.

In summary, the input event handler synchronizes the (asynchronous) input from the user (speech or keyboard or mouse events). All inputs are translated to an equivalent in-domain natural language expression and sent to the dialog manager and the understanding system which in turn returns a template. The template is parsed by the output event handler and is followed by a multimedia presentation of the system's response (text, graphics, animation, speech). Further details on the communication between the various modules is given in the next section.

3.2. Architecture

A detailed architectural diagram is shown in Fig. 2, where the focus is on the communication between the controller

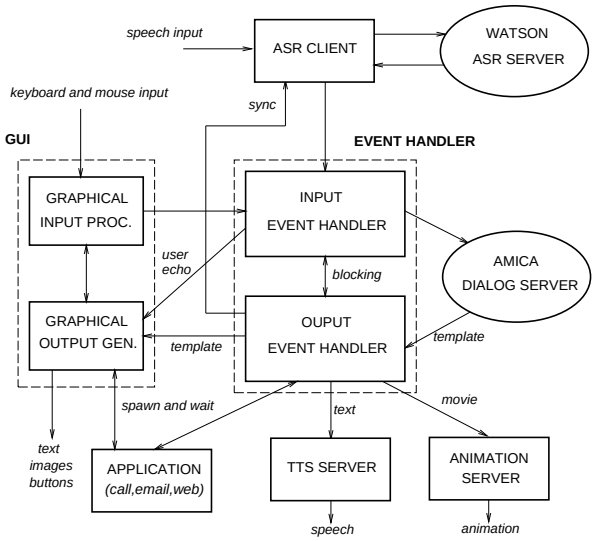

Figure 2: Architectural diagram of CHIMP prototype.

and various servers. The controller is an asynchronous I/O event handler and comprises of two separate event loops for the input and output modalities, respectively. The program flow of the controller event loops is very simple: (i) input events are sent to the dialog manager in the form of text strings (input loop), (ii) templates received from the dialog manager are parsed for multimodal output fields and passed on to the appropriate output modules (text-to-speech, animation/movie player or the graphical user interface). During processing of requests by the dialog manager all input events are queued up. If multiple events have been queued up in the input, only the latest event is sent to the dialog manager for processing. The output event handler has the additional functionality of being able to surrender control or simply start up external applications (e.g., pop up an e-mail reader or a web browser). Finally, the controller handles barge-in events (speaking- or typing-over voice prompts or animation sequences) from the ASR client by informing the text-to-speech and movie player modules to stop playback of speech and/or animation sequences.

The dialog manager processes incoming strings by following a control language. Strings are interpreted by calling the understanding module and by requesting information from a postgres-SQL backend database server. State information and (requested) state history are saved in a template (frame) that is passed around from control module to control module in the dialog manager and is finally surrendered to the controller as prescribed by the script in the dialog manager. Then the dialog manager pauses and waits for input from the controller. The ASR client communicates with the AT&T ASR WATSON server through a wireline protocol. The commands and audio flows from the client to the server while the results and notifications flow the other way. The interaction between the ASR client and server is through polling or notification modes.

An important design principle of the prototype is that all state information is kept internally in the template that is generated and updated by the dialog manager. However, currently state information is also kept locally at the GUI where parsing of the template is required. This discrepancy is often encountered in practical systems. In the near future, we intend to eliminate all state information from the GUI by using a dialog manager generated GUI scripting language with features from HTML and Tcl/Tk.

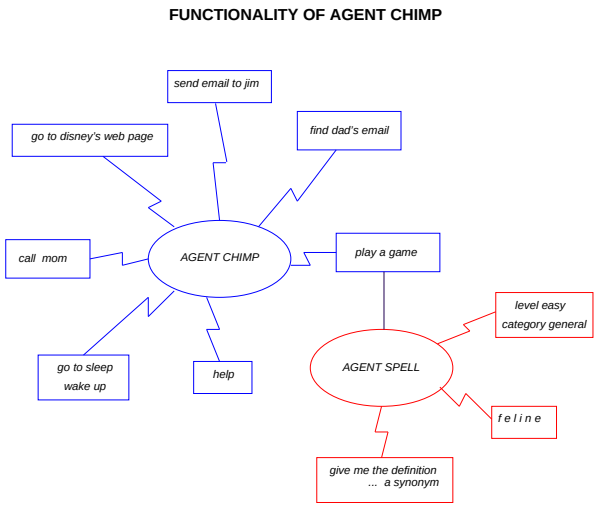

Figure 3: Overview of some of the application features.

3.3. Application details

The prototype consists of two distinct applications as shown in Figure 3: a communications agent application (information retrieval from a personal directory, placing phone-calls, accessing the internet, sending email) and a computer game (spelling bee). Both applications are controlled by an animated agent. The personality and appearance of the agent, however, are different across the two applications. The user can switch back and forth between the applications at any point during the interaction. The system also has 'go to sleep' and 'wake up' features by means of which the personal agent's attention can be controlled. The dialog manager design supports both mixed-initiative and system-initiative strategies. Ambiguity resolution (for example, in the case of the retrieval of multiple records) and error control (using confidence measures provided by utterance verification) are implemented as a part of the dialog strategy. Help messages are available through GUI and through audio prompts. Animation provides a key presentation modality. The agent uses appropriate gestures such as pointing, nodding, and shrugging to convey retrieved information, error conditions, and confusions, respectively. Animation also aims to provide an engaging interface for the child user. Sleep and wake agent modes are animated with vanishing and re-appearing animation sequences.

The spelling game provides a richer and more challenging application domain. One of the design objectives was to explore how to design interactive educational tutors for children. Agent Spell has a somewhat studious personality and gives appropriate feedback to the child user depending on how the game is progressing. A score meter on the GUI keeps track of the number of attempts per word, and overall scores with appropriate accompanying audio prompts. The child can select to play a spelling game of a difficulty level of choice (easy/medium/hard) and in a subject area of choice.

Informal evaluation of the prototype by children users was very positive, especially for the animated agent and the speech interface. Children enjoyed the naturalness and flexibility of the interface and communicating with the animated agent. Their main criticism was the limited range of interaction one could have with the animated agent (there were only about ten animated sequences) and the lack of understanding of out-of-domain user requests. Formal evaluation of the prototype remains to be done. The prototype can also serve as a testbed for investigating synchronization and merging of input modalities and multimedia presentation.

4. DISCUSSION

Since the system was designed for children users, heavy emphasis was placed on the interface. Indeed it was found that using animated sequences to communicate information and adding 'personality' to the interface significantly improved the user experience. In addition, the flexible choice of input modality (any of speech, natural language, commands or buttons) made the application easy to use even for novice users.

In addition to the user interface, the prototype serves as a testbed for a general multimodal system architecture. The main design principle of our system is a modular architecture, where the controller communicates with 'stateless' servers via text messages (all state information resides in the template). Seamless integration of all input modalities for our applications is achieved by translating all inputs into text strings that are in turn handled by the spoken language understanding system. Other features of our system not yet implemented include customizable application content and agent personality. Overall, the prototype represents a successful first effort in building a multimodal system for children with an emphasis on conversational speech.

Acknowledgments: The authors would like to express their sincere appreciation to Christa Lazarus, Matthew Einbinder and Roger Barkan for organizing the "Carmen Sandiego" Wizard of Oz experiments and for collecting and analyzing the user experience data, and to Selina Chu for the design of the Agent Spell module.

5. REFERENCES

[1] P. Cohen, M. Johnston, D. M. and S. Oviatt, J. Clow, and J. Smith, "The efficiency of multimodal interaction: A case study," in *Proc. ICSLP 98*, (Sydney, Australia), 1998.

[2] S. Lee, A. Potamianos, and S. Narayanan, "Acoustics of children's speech: Developmental changes of temporal and spectral parameters," *J. Acoust. Soc. Am.*, vol. 105, pp. 1455–1468, Mar. 1999.

[3] J. Mostow, A. G. Hauptmann, and S. F. Roth, "Demonstration of a reading coach that listens," *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 77–78, 1995.

[4] R. Pieraccini and E. Levin, "A spontaneous-speech understanding system for database query applications ," in *ESCA Workshop on Spoken Dialogue Systems - Theories and Applications*, 1995.

[5] R. Pieraccini, E. Levin, and W. Eckert, "AMICA: the AT&T Mixed Initiative Conversational Architecture," in *Proc. EuroSpeech*, (Rhodes, Greece), Sept. 1997.

[6] A. Potamianos and S. Narayanan, "Spoken dialog systems for children," in *Proc. ICASSP 98*, (Seattle, WA), May 1998.

[7] A. Potamianos, S. Narayanan, and S. Lee, "Automatic speech recognition for children," in *Proc. EuroSpeech*, (Rhodes, Greece), pp. 2371–2374, Sept. 1997.

[8] A. Potamianos, G. Riccardi, and S. Narayanan, "Categorical understanding using statistical N-gram models," in *Proc. EuroSpeech*, (Budapest, Hungary), Sept. 1999.

[9] G. Riccardi, A. Potamianos, and S. Narayanan, "Language model adaptation for spoken language systems," in *Proc. ICSLP 98*, (Sydney, Australia), pp. 2327–2330, 1998.

[10] R. Sharma, V. Pavlovic, and T. Huang, "Toward multimodal human computer interface," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 853–869, 1998.

[11] E. F. Strommen and F. S. Frome, "Talking back to big bird: Preschool users and a simple speech recognition system," *Educational Technology Research and Development*, vol. 41, pp. 5–16, 1993.

[12] T. Takezawa and T. Morimoto, "A multimodal-input multimedia-output guidance system: MMGS," in *Proc. ICSLP 98*, (Sydney, Australia), 1998.