Towards Efficient Human Machine Speech Communication: The Speech Graffiti Project

STEFANIE TOMKO, THOMAS K. HARRIS, ARTHUR TOTH, JAMES SANDERS, ALEXANDER RUDNICKY, and RONI ROSENFELD
Carnegie Mellon University

This research investigates the design and performance of the Speech Graffiti interface for spoken interaction with simple machines. Speech Graffiti is a standardized interface designed to address issues inherent in the current state-of-the-art in spoken dialog systems such as high word-error rates and the difficulty of developing natural language systems. This article describes the general characteristics of Speech Graffiti, provides examples of its use, and describes other aspects of the system such as the development toolkit. We also present results from a user study comparing Speech Graffiti with a natural language dialog system. These results show that users rated Speech Graffiti significantly better in several assessment categories. Participants completed approximately the same number of tasks with both systems, and although Speech Graffiti users often took more turns to complete tasks than natural language interface users, they completed tasks in slightly less time.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Voice I/O; Natural language; Interaction styles*

General Terms: Human Factors, Design, Experimentation

Additional Key Words and Phrases: Human-computer interaction, speech recognition, spoken dialog systems

1. INTRODUCTION

As the most common mode of human-human interaction, speech can be considered an ideal medium for human-machine interaction. Speech is natural, flexible and most humans are already fluent in it. Using speech allows users to simultaneously perform other tasks which may or may not be related to the

This work was supported by grants from the Pittsburgh Digital Greenhouse, a National Defense Science and Engineering Graduate Fellowship, and Grant No. N66001-99-1-8905 from the Space and Naval Warfare Systems Center, San Diego, CA. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.
Authors' address: Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213; email: roni@cs.cmu.edu.

spoken task. Machine speech requires modest physical resources and can be scaled down to much smaller and much cheaper form factors than visual or manual modalities.

Technology now exists for allowing machines to process and respond reliably to basic human speech, and speech is currently being used as an interface modality in several commercially available applications such as dictation systems, web browsers, and information servers. However, we believe that speech would achieve even higher adoption as an interface technology if certain fundamental limitations were addressed, particularly

(1) recognition performance,
(2) language habitability (for users),
(3) ease of development (for implementers).

Natural interaction with computers has often been cited as a primary benefit of speech. The concept of talking with a machine as fluently and comfortably as with another human being has attracted funding and interest. However, from the user's perspective, fully natural communication may not be the most desirable option. For instance, Shneiderman [1980] suggests that natural communication may actually be too lengthy for frequent, experienced users who expect a system to give them information as quickly as possible, and other studies have suggested that users' natural inclination for talking to computers is to be "short, succinct and task specific; using simple imperative commands and . . . a restricted vocabulary" [Baber 1991]. Following these observations, our research focus has been on exploring speech as an efficient input/output modality rather than as a medium for natural communication. We still believe however that natural language speech interaction is a worthwhile challenge and feel that our research can provide an alternative to natural language communication in certain situations as well as suggest potential strategies for improving natural language systems (e.g., a restricted language might be used as a "back-off" strategy in natural language systems experiencing high error conditions).

The Speech Graffiti interface is the result of our research into these issues. This article presents further motivation for creating such an interface, describes its general characteristics, shows examples of its use, summarizes user study results, and describes other aspects of the system such as the development toolkit and a user tutorial.

2. BACKGROUND

2.1 Speech User Interface Styles

Although speech recognition technology has made spoken interaction with simple machines (in which high-level intelligent problem-solving is performed by the human user as opposed to the system) feasible, no suitable universal interaction paradigm has been proposed for facilitating effective, efficient, and effortless communication with such machines. In general, approaches to speech interfaces for simple machines can be divided into three categories: *command-and-control*, *directed dialog*, and *natural language*. One way to differentiate

these categories is in terms of what language can be used when interacting with the system and how easy it is for the system to process user input.

Command-and-control systems severely constrain what a user can say to a machine by limiting its vocabulary to strict, specialized commands. Since such systems do not require overly complicated grammars, these can be the simplest types of systems to design and can usually offer low speech recognition word-error rates. Command-and-control systems can be difficult to use, however, since interaction skills learned in one application do not necessarily transfer to other applications.

Directed dialog interfaces use machine-prompted dialogs to guide users to their goals, but the interactions can be slowed by requiring the user to follow a certain path, generally inputting one piece of information at a time. From a designer's perspective, such systems can be difficult to build because they require breaking down an activity into the form of a dialog graph; maintenance is difficult because it may require rebalancing the entire tree as new functions are incorporated.

In natural language interfaces, users can pose questions and give directives to a system using the same open, complex, conversational language that they would be likely to use when talking to another human about the same task. Allowing the user such a substantial degree of freedom alleviates the need for the user to learn specialized commands or to work within a rigid access structure. However, a natural language interface puts a heavy burden on system developers who must incorporate a substantial amount of domain knowledge into what is usually a very complex model of understanding and who must include all reasonably possible user input in the system's dictionary and grammar.

More importantly, although the inherent naturalness of natural language interfaces suggests that they should be quite simple to use, this apparent advantage can, at the same time, be problematic: the more natural a system is, the more likely it is for users to overestimate its bounds and form unrealistic expectations about this system [Perlman 1984; Glass 1999]. That is, although the goal of natural language systems is open, flexible communication, there are in practice significant limits to what any current system can understand in terms of vocabulary, syntax, and functionality, and users will find that input that is acceptable in one natural language interface may be rejected in another.

2.2 The Speech Graffiti Approach

Given the current state-of-the-art of natural language dialog systems, we believe that the optimal style for speech communication with simple machines lies somewhere between natural language and command-and-control. The Speech Graffiti paradigm is more structured than natural language, yet more flexible than hierarchical menus or strict command-and-control. Speech Graffiti was modeled after two very successful nonspeech interaction paradigms: Macintosh-style graphical user interfaces (GUIs) and the Graffiti® writing system for personal digital assistants (PDAs).

With GUIs, once a user learns the basic set of interactive behaviors (double-clicking, dragging, the Edit menu, etc.), these behaviors can be transferred to

almost any other GUI application. We believe that a universal speech interface like Speech Graffiti can have the same benefit. If a speech interface user knows, for instance, that the system will always confirm whatever parts of the user input it understood, or that they can always say **`options`** to find out what they can talk about at a given point, learning how to use new Speech Graffiti applications should be significantly easier. The existence of a standardized look-and-feel is particularly advantageous for spoken interfaces because the input behavior is invisible and must be remembered by the user.

The Graffiti® alphabet for PDAs requires users to slightly modify their handwriting in a standardized way in order to improve recognition performance. Although this requires that users invest a modest amount of time in learning the alphabet, the increase in handwriting recognition accuracy compared to that of systems designed to recognize users' natural handwriting is so significant that the use of Graffiti has been posited as one of the main reasons for the commercial success of the Palm handheld [Blickenstorfer 1995]. Similarly, in the Speech Graffiti interface, users are asked to phrase their spoken input in a certain way in order to improve speech recognition accuracy and reduce dialog complexity. Speech Graffiti users need to spend a short amount of time learning the interface but this is a one-time cost that is amortized by increased recognition accuracy.

3. RELATED WORK

The restriction on the form of Speech Graffiti user input means that it can be considered a *subset language*—an artificially constructed subset of natural language, designed for a specific purpose (though not necessarily for a specific domain). Sidner and Forlines [2002] investigated the use of a subset language for a home entertainment center and found that subjects were able to complete all given tasks successfully with the subset language and that their performance did not decline when performing tasks the following day, demonstrating that users were able to retain their knowledge of the language.

Zoltan-Ford [1991] investigated the use of restricted languages in both spoken and typed input and found that users generally did not mind having to use a restricted language. In fact, she found that study participants believed that computers naturally require consistency in input and that even in human-human communication, some amount of adaptation to a partner's speaking style is necessary.

The idea of universalizing commands in order to facilitate error recovery and decrease cross-application user training requirements has been promoted by industry groups and studies have been conducted to determine appropriate standard keywords for speech interfaces [Telephone Speech Standards Committee 2000; Guzman et al. 2001].

4. SPEECH GRAFFITI: DESIGN AND GENERAL CHARACTERISTICS

Speech Graffiti is designed to provide regular mechanisms for performing interaction universals. Interaction universals are actions which are performed by users at one time or another in nearly all speech user interfaces; the set of

universals addressed by Speech Graffiti was derived by analyzing several types of simple machines and application categories prior to developing the Speech Graffiti vocabulary.

These universals include actions involving help and orientation, speech recognition, basic system functions, and application-type-specific functions. For this last category, we recognize that different types of applications can have different interaction needs. For example, a transaction interface must include structures that allow the user to pay, bid, confirm, and so on, while a system for controlling a gadget must include structures for giving simple commands (e.g., rewind, toast, turn off) and setting continuous variables (e.g., adjusting volume level). Although these structures may vary by application type, they should be standardized for all applications of the same type.

In general, Speech Graffiti addresses these interaction universals by means of keywords and standard structures. Keywords are appropriate for some interaction universals which involve the performance of specific actions, while other universals, such as confirmation and error handling, require a standardized protocol for input and output rather than a single keyword. This section will discuss the specific ways in which Speech Graffiti addresses these interaction universals.

Since Speech Graffiti has so far been most fully developed in the domain of information access, the descriptions provided here are specialized for that type of application. A representative Speech Graffiti dialog, adapted from Speech Graffiti MovieLine interactions during a user study, is shown in Figure 1. References to line numbers from this dialog will be written as they are in the figure, for example, *S21*.

4.1 User Input

4.1.1 *Lexicon.* The lexicon of a Speech Graffiti application consists of two parts: a set of universal keywords and an application-specific vocabulary. The complete set of keywords, which will be discussed in the following sections, is summarized in Table I. Some synonyms are also allowed, such as **`where were we?`** for **`where was I?`**

Application-specific vocabulary. The size and contents of the application-specific vocabulary are naturally determined by the functionality and complexity of each application and will generally be quite a bit larger than the Speech Graffiti keyword set. The current Speech Graffiti MovieLine lexicon includes around 400 words, approximately 160 of which are movie titles. By comparison, a natural language system created to access the same movie information database contains around 800 words.

Keywords. For Speech Graffiti to be a truly universal interface, it must incorporate a small set of words that nontechnical users will feel comfortable with. Our original selection of keywords was based largely on our own intuitions about which words had simple unambiguous meanings and were, at the same time, relatively acoustically distinct. We later conducted an Internet-based user study to investigate the appropriateness of our keyword choices and to solicit possible alternative keywords from users [Shriver and Rosenfeld 2002]. Some

User1 **Start over.**
System1 START OVER.
U2 **Theater is the Manor, title is Abandon.**
S2 THE CINEMAGIC MANOR THEATRE, ABANDON.
U3 **What are the show times?**
S3 2 SHOW TIMES: 1:25, 5:30.
U4 **Start over.**
S4 STARTING OVER.
U5 **Theater is Maxi Saver, what is the address?**
S5 CARMIKE MAXI SAVER 12 THEATER: 2001 MOUNTAIN VIEW DR. IN WEST MIFFLIN.
U6 **Start over.**
S6 STARTING OVER.
U7 **Options.**
S7 YOU CAN SPECIFY OR ASK ABOUT TITLE, THEATER, SHOW TIME, {…}.
U8 **More.**
S8 DAY, GENRE, RATING, {…}.
U9 **More.**
S9 AREA, ADDRESS, PHONE NUMBER.
U10 **More.**
S10 NO MORE MATCHES.
U11 **Area options.**
S11 AREA CAN BE THE NAME OF A NEIGHBORHOOD OR CITY, OR A REGION, SUCH AS NORTH OR EAST, OR YOU CAN ASK, "WHAT IS THE AREA?"
U12 **Area is Monroeville.**
S12 MONROEVILLE.
U13 **What are the theaters?**
S13 SHOWCASE CINEMAS PITTSBURGH EAST.
U14 **Theater is Showcase Cinemas Pittsburgh East.**
S14 SHOWCASE CINEMAS PITTSBURGH EAST.
U15 **What are the titles?**
S15 11 TITLES: MY BIG FAT GREEK WEDDING, ABANDON, BROWN SUGAR,{…}.
U16 **Genre is comedy.**
S16 COMEDY.
U17 **Where am I?**
S17 AREA IS MONROEVILLE, THEATER IS SHOWCASE CINEMAS PITTSBURGH EAST, WHAT ARE THE TITLES?, GENRE IS COMEDY.
U18 **What are the titles?**
S18 5 TITLES: MY BIG FAT GREEK WEDDING, BROWN SUGAR, JONAH - A VEGGIETALES MOVIE, {…}.
U19 **Title is My Big Fat Greek Wedding.**
S19 MY BIG FAT GREEK WEDDING.
U20 **What are the show times?**
S20 4 SHOW TIMES: 3:45, 5:45, 7:45, 9:45.
U21 **Start over. Theater is Pleasant Hills.**
S21 {CONF!}, STARTING OVER.
U22 **Location is Pleasant Hills.**
S22 PLEASANT HILLS.
U23 **What is theater?**
S23 CARMIKE SOUTHLAND 9.
U24 **Theater is Southland Nine.**
S24 CARMIKE SOUTHLAND 9.
U25 **What is movie?**
S25 7 TITLES, THE RING, ABANDON, GHOST SHIP, {Σ}.

Fig. 1. Sample Speech Graffiti MovieLine dialog. The notation {. . . } represents a three-beep auditory icon signaling the continuation of a list.

Table I. Speech Graffiti Keyword Summary

repeat replays the system's last utterance	**more, next, previous, first, last** and **stop** navigate through items in a list	**scratch that** cancels the effect of the user's last utterance	**go ahead** sends the user's query to the application
start over erases all accumulated context	**where was I?** tersely restates the accumulated context	**options** lists what can be said next at this point	**what is ...?** queries the value of a specific item

of our original keyword choices (e.g., **start over** and **repeat**) performed well in the study, while others were replaced as a result (for example, **options** replaced **now what?** as the keyword for asking what can be said at any point in an interaction).

It is also desirable to keep the number of Speech Graffiti keywords small in order to minimize the effort required to learn and retain them. Currently the system incorporates seven main keywords plus another six navigation keywords, as shown in Table I.

4.1.2 *Phrases.* The primary action in the information access domain is a database query. In Speech Graffiti, database queries are constructed by joining together phrases composed of *slot* + *value* pairs. Phrases are order-independent and each user utterance can contain any number of them. The *slot* + *value* phrase format simplifies the work of the parser and roughly conforms to natural speaking patterns.

When a phrase is used to specify a constraint, its syntax is **<slot> is <value>**, as in *U14* (Figure 1). When a phrase is used to denote a slot being queried, its syntax is **what is <slot>?**, as in *U15*. In order to reduce the command-and-control feel of Speech Graffiti, these user input structures have been influenced by natural language. For instance, common synonyms can be accepted in many situations (e.g., **movie** and **title** both represent the same slot in a movie information system), and plural forms are accepted wherever they would naturally be used (e.g., **what are the theaters?** is equivalent to **what is theater?**).

It is worth emphasizing that our current choice to restrict phrases to this simple syntactic format is not driven by limitations of the parsing technology. The Phoenix parser we use can accept far more elaborate grammars, and, in fact, we have used such flexible grammars in many other dialog applications. Rather, our choice is based on arguments in support of a structured, simplified interaction language and the many benefits it brings: increased recognition and understanding accuracy, improved system transparency (including clear lexical, syntactic, and functional boundaries), and dramatically reduced development time. While we believe that a semistructured, seminatural-language interface style will ultimately become ubiquitous in human-machine speech

Theater is the Manor.
THE CINEMAGIC MANOR THEATRE.
What are the movies?
3 TITLES: CHICAGO, DAREDEVIL, DOWN WITH LOVE.
What are the show times?
THE CINEMAGIC MANOR THEATRE. 3 TITLES, 15 SHOW TIMES: CHICAGO: 5:30. DAREDEVIL: 4:45, 7:30, {…}
Genre is action, go ahead.
ACTION. 1 TITLE, 3 SHOW TIMES: DAREDEVIL: 4:45, 7:30,

Fig. 2. Sample MovieLine dialog illustrating context retention.

communication, we do not necessarily expect our current design choices to be the optimal ones. Our goal is to assess different designs in terms of the benefits they bring versus their cognitive cost.

4.1.3 *Grammar.* A valid user utterance in the Speech Graffiti language consists of any number of **<slot>**+**<value>** or **what**+**<slot>** phrases. Keywords such as **goodbye**, **repeat**, or **help** can occur by themselves or, less commonly, at the end of a string of **<slot>**+**<value>** phrases. A grammar describing the language more formally is included in Appendix A.

4.2 System Output

In many speech interface situations, no visual display is available so extra care must be given to the design of audio output to ensure that the system is able to convey information and express concepts to the user clearly and accurately. However, we believe that it should not be necessary to always present information in the most verbose manner possible. Indeed, doing so would be a violation of the Gricean maxim that conversational contributions should be no more and no less informative than necessary [Grice 1975]. Unnecessary verboseness and repetition in a system can become tiring; since we propose Speech Graffiti as a standard interface for systems that users might interact with several times a day, this effect is multiplied. Furthermore, one of the proposed advantages of Speech Graffiti is to facilitate efficient communication by allowing direct access to tasks that are too cumbersome for prompt- and menu-driven systems; using output that is too verbose could negate the effects of this strategy. Speech Graffiti implements its nonverbose output strategy via terse confirmation, segmented lists, and auditory icons.

4.3 Interaction Details

4.3.1 *Context.* Speech Graffiti can be set to retain or discard context depending on the requirements of individual applications. If context is turned off, parsed phrases are discarded after each query command. If context is retained, all parsed phrases since the last clearing of context are used to produce a database query string. Figure 2 shows an example of context retention. When the user asks about show times in the third utterance, context has not been cleared so the system returns all three movies (from the previous query) and show times for the Manor theater.

Departure airport is Pittsburgh, arrival airport is Detroit, arrival time is before 11:00am.
FROM PITTSBURGH, TO DETROIT, ARRIVING BEFORE 11:00AM.
Airline is Northwest, go ahead.
NORTHWEST. 2 MATCHES: FLIGHT 1921, NONSTOP FLIGHT, DEPARTING AT 7:02AM {…}.
More.
DEPARTING FROM GATE D83, ARRIVING AT 8:08AM, ARRIVING AT GATE A53, {…}.
More.
FLIGHT 1917, NONSTOP FLIGHT, DEPARTING AT 9:35AM {…}.

Fig. 3. Sample FlightLine dialog illustrating complete record retrieval.

Context is cleared using the **start over** keyword or individual slots may be overwritten via respecification. Our current implementation allows a slot to take only a single value at a time so restating a slot with a new value overrides any previous value of that slot. This behavior would be altered for other domains in which slots are allowed to take multiple values. For example, in a pizza-ordering system, the "topping" slot would be allowed to hold more than one value at a time.

Speech Graffiti will execute a query immediately upon processing an utterance containing a query phrase. If, after hearing the response, the user would like to reissue the same query (either with the same exact slots and values or after having respecified some slot(s)), the **go ahead** keyword is used as in the fourth user utterance in Figure 2.

In some applications, it may be appropriate for users to request a complete set of information rather than querying a specific slot. For instance, in the FlightLine application, a user might want to know *all* the pertinent information about a flight that matches the user's constraints. Rather than ask the user to issue a query phrase for each slot (e.g., **what airline, what is the arrival time, what is the departure time**, etc.), the speaker can simply use the **go ahead** keyword to effectively query all unspecified slots as shown in Figure 3. This approach is most appropriate for applications using simple databases containing a single table. Because our user studies have concentrated on the MovieLine application which contains three tables, we have not been able to assess how well users learn about the use of **go ahead**.

4.3.2 *List Presentation.* In database applications, information that is returned to the user often takes the form of a list. In keeping with the Speech Graffiti philosophy of presenting just as much information as is useful, our general strategy is to output information in small manageable chunks. Therefore, lists are presented with three items at a time (*S18*), or four if a split would result in a one-item orphan chunk (*S20*). The notation {. . .} in our text examples *S8* and *S15* represents an auditory icon played at the end of a chunk to indicate that the list continues beyond the current chunk. {. . .} is currently implemented as a brief three-beep signal intended to suggest the written punctuation for ellipsis (. . .). The initial list chunk is prefixed with a header indicating the size of the entire list, for example, 11 TITLES (*S15*). If the queried data is not available in the database, Speech Graffiti returns the string SORRY, THAT INFORMATION DOESN'T APPEAR TO MATCH ANYTHING IN OUR DATABASE.

4.3.3 *List Navigation.* Speech Graffiti includes a suite of keywords for navigating through lists: **more**, **next**, **previous**, **first**, **last**, and **stop**. **More** is used to access additional information of the same type, that is, the next chunk at the same level of information. In instances where the speaker has used **go ahead** to retrieve complete record information, the user can jump to the next item in the list (as opposed to the next chunk for the initial item) by saying **next** (in simple lists this keyword functions the same as **more**). This can be thought of graphically as navigating a two-dimensional table, with **more** continuing horizontally and **next** continuing vertically. **Previous** returns the previous chunk in the list, **first** returns the initial chunk, and **last** returns the final chunk in the list. Each navigation keyword can be followed by an integer which allows the user to customize the size of the list returned. For example, **last six** would return the six items at the tail end of the list.

In our current tutorial sessions, users are only told about the navigation keyword **more**. In the studies reported in this article, **more** accordingly had widespread use, while the other keywords were used sparsely. We suspect that in a longer-duration study, users would begin to use some of the other navigation keywords as they became more comfortable with the system.

Splitting complex output into chunks not only helps to avoid information overload, but also enables the **repeat** keyword to act on current, smaller segments of information that the user might be interested in hearing again.

4.3.4 *Turn-Taking.* Speech Graffiti responds to each user input with a terse standardized confirmation (*S2*, *S14*); the user can then correct this item if necessary or continue on with their input. The **repeat** keyword always replays the last system utterance.

4.3.5 *Question Answering.* As discussed in Section 4.1.2, queries are formed using the **what is <slot>?** structure. Our earliest Speech Graffiti implementations included a terminator keyword which would signal that the user's command was complete and ready to be executed by the system (e.g., **theater is the Manor, what are the movies? go!**). This eased the processing burden since the speech recognition component simply needed to spot a terminator and then pass along the recognized string to the parser. The terminator keyword also increased the flexibility of phrase order: users could state a query phrase and then add specification phrases in subsequent utterances before sending the complete command to the system. However, we found that users had difficulty remembering to use the keyword. Once a query phrase (e.g., **what are the movies?**) has been uttered, users naturally expect the system to provide an answer, and the system has been refined to accommodate this expectation. As discussed in Section 4.3.1, **go ahead** is used to re-execute a query phrase stored in context from a previous utterance.

4.3.6 *Session Management.* Each session begins with a brief recorded introduction to the system; experienced users can barge-in on this introduction and start their interaction. When Speech Graffiti recognizes **goodbye**, it replies with GOODBYE!, but the system remains active in case the input was misrecognized. If the user wants to continue, they can simply speak again; if not,

they can just hang up. Since our information access applications are currently telephone-based, sessions are initiated by the user calling the system.

4.3.7 *Undo.* See Section 4.3.9 (correcting ASR errors). Slots can also be cleared (i.e., set to no value) using the value **anything** as in **<slot> is anything**. The entire context can be cleared using the **start over** keyword.

4.3.8 *Help Primitives. What can the machine do?* Currently this is addressed by the **options** keyword which returns a list of slots the user and system can talk about. A slightly different functionality could be provided by a keyword like **what can you do?** which should return a list of high-level application functions. This is probably more appropriate for multifunction applications which we have not yet implemented in the information access domain. An example might be a movie information system which would provide information about movies and show times and also sell tickets.

What can I say? As noted above, the **options** keyword returns a list of what can be said at any given point in the interaction. If used by itself (*U7*), **options** returns list of available slots. If used as **<slot> options** (*U11*), a list of values that can be paired with **<slot>** is returned. If the values that a particular slot can take make up a standard class or make too long of a list to be enumerated efficiently (even when split into chunks), the response to **<slot> options** can be a description of the possible values instead. For instance, the system response to **show time options** is SHOW TIME CAN BE A TIME SUCH AS SEVEN THIRTY, OR A RANGE LIKE AFTER NINE O'CLOCK.

I need help. Currently, the **help** keyword allows users to get assistance on the keywords and the basic form of Speech Graffiti input. If a user says **help**, Speech Graffiti will return an example of how to talk to the system plus a short list of appropriate keywords for either general use or list navigation, depending on where the user is in the interaction. If the user asks for help again at the end of this message, the system returns a more detailed explanation of the system's features.

4.3.9 *Speech Channel Primitives. Detecting automatic speech recognition (ASR) errors.* Errors occur when the Phoenix parser cannot completely parse input passed to it. This may occur either because of a misrecognition of the speech input or because the user simply did not speak within the Speech Graffiti grammar. Speech Graffiti uses a succinct confirmation strategy in which the system confirms, with a short paraphrase, only those phrases which it has understood (*S2*, *S12*, *S14*). By responding this way, the system does not distinguish between different types of errors which may have occurred. If an error occurs in which user input is misinterpreted as acceptable input that does not match what the user said (e.g., the user says **area is Monroeville** and the system hears **area is Squirrel Hill**), the user can recognize the error from the explicit confirmation.

If the system receives an utterance that cannot be fully parsed due to either type of error, it prefixes its confirmation of any understood and parsable phrases with an auditory icon (*S21*). This icon is represented as {CONF!} (for *confusion*) in our text and is currently implemented as an error-signifying beep. The

system will respond with only {CONF!} if no part of the input was understood. On a third consecutive input that contains no parsable information, Speech Graffiti responds with the more verbose {CONF!} I'M SORRY, I'M HAVING TROUBLE UNDERSTANDING YOU. TRY AGAIN.

Correcting ASR errors. **`Scratch that`** is the primary Speech Graffiti keyword for correcting errors, although other strategies can be used as well. If used independently, **`scratch that`** clears a user's previous utterance. If used in conjunction with other input, **`scratch that`** clears all preceding input from the *same* utterance, thereby allowing users to self-correct disfluencies. As noted in Section 4.3.1, respecifying a slot will override any value already stored there so corrections can also be made this way. In the most extreme case, a user could opt to say **`start over`** and re-attempt their command from the beginning.

The phrasal structure of Speech Graffiti also helps to mitigate the effects of errors and reduce the amount of duplicated or unnecessarily reconfirmed information in subsequent utterances. That is, although expert users might enter several specification phrases and a query phrase in a single utterance, a user experiencing recognition problems can enter one phrase at a time, making sure it is successfully confirmed before moving on to the next phrase. This may slow down the overall interaction but can be used as a fallback strategy when the system's recognition rate is low (for instance, if the user is speaking in a noisy room or has a strong accent).

End of speech. Speech Graffiti plays a quiet beep when Sphinx has determined the endpoint of an utterance and has started to process this input.

5. SYSTEM ARCHITECTURE

The Speech Graffiti implementation is modular, with its various components residing on multiple machines spanning two platforms (Linux and Windows NT). The dialog manager consists of an application-independent Speech Graffiti engine and an application-specific domain manager. The Speech Graffiti engine interacts with a Phoenix parser [Ward 1990], and the domain manager accesses a commercial database package. These components together constitute a stand-alone, text-based version of the system which can be developed and tested independently of the speech recognition, speech synthesis, and telephony control components. In the experiments reported here, speech recognition was performed by the CMU Sphinx-II engine [Huang et al. 1993], using acoustic models based on Speech Graffiti applications and statistical language models created with the CMU/Cambridge SLM Toolkit [Clarkson and Rosenfeld 1997]. Unit-selection-based, limited-domain speech synthesis was generated using the Festival system [Black et al. 1998; Black and Lenzo 2000].

6. RELATED COMPONENTS

6.1 Application Generator

One of the acknowledged impediments to the widespread use of speech interfaces is the *portability problem*, namely the considerable amount of labor, expertise, and data needed to develop such interfaces in new domains. Speech

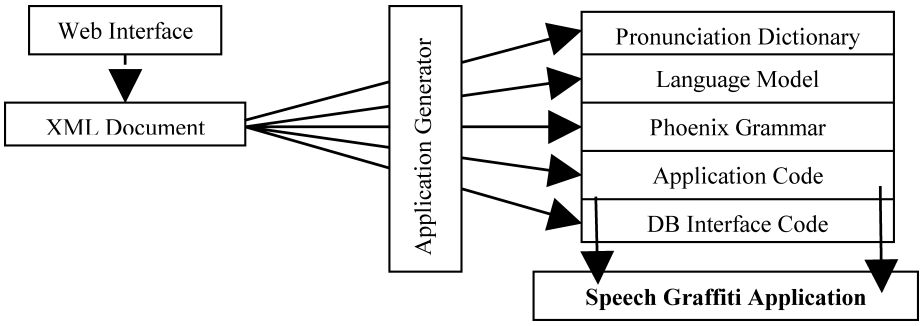

Fig. 4. Speech Graffiti application generation process.

Graffiti's semistructured interaction reduces the need for vast in-domain data collection, and the unified structure of Speech Graffiti interfaces also makes possible the automatic generation of new interfaces from a terse high-level specification. We have created a toolkit comprising all the necessary programs and files to create and run Speech Graffiti information access applications [Toth et al. 2002]. Together, these components

- generate code for the domain manager which accesses a given database;
- generate a grammar file for the Phoenix parser that enforces the Speech Graffiti interaction style and is consistent with the database content;
- generate a language model and pronunciation dictionary for the Sphinx speech recognition system which are consistent with the grammar; and
- properly cross-link these various knowledge sources so that multiple generated Speech Graffiti applications do not interfere with each other's operation.

The application-specific variables are collected for insertion into the various components via an XML document. Application developers can either create this XML document by hand using the Speech Graffiti Document Type Definition (DTD) as a template, or they can utilize the Speech Graffiti Web Application Generator. The Web Application Generator is an Internet-based program that allows the developer to describe their application via a series of Web forms from which an appropriate XML document is then derived. Regardless of whether the developer uses the Web interface or manually codes the XML document, a Perl script is available to convert the application-specific information from the XML file into all of the components previously discussed. Application developers can edit the resulting code to further customize the application to their needs. Figure 4 shows a schematic of this process, and Appendix B shows a screenshot and XML code fragment from the generation process.

In addition to the MovieLine system, we have also generated Speech Graffiti information-access applications for databases of airline flight information, rental property availability, bus travel information, and facts and figures for American states. This experience has shown us that Speech Graffiti's features appear to cover the interface requirements for information-access tasks, although the system would need to be modified to handle more complex database

tasks such as the insertion and updating of data. We have not yet formally evaluated the application generation process through user studies.

6.2 Appliance Control

As a framework for investigating the application of Speech Graffiti principles in the appliance-control domain, we built the Speech Graffiti Personal Universal Controller (SG-PUC). Its specification language and communications protocol effectively separate the SG-PUC from the appliances that it controls, enabling mobile and universal speech-based appliance control. The development of interfaces to numerous appliances and the results of user studies have demonstrated the usefulness of the SG-PUC, indicating that a high-quality, low-cost human-appliance speech interface can be largely appliance-agnostic [Harris and Rosenfeld 2004]. As in the information-access domain, the use of a universal control language provides the benefit of clear unambiguous semantics and low input perplexity. These factors translate into a more robust system with fewer errors than functionally equivalent natural language speech interfaces.

7. EVALUATION

Our most comprehensive evaluation of Speech Graffiti to date has been a user study comparing Speech Graffiti and a natural language interface [Tomko 2004]. Our main goal was to determine whether users would prefer a more efficient yet structured interaction over one that was more natural but perhaps less efficient. In this study, we compared various subjective and objective measures, including user satisfaction and task completion rates and times, between a Speech Graffiti system (SG-ML) and a natural language system (NL-ML) that both accessed the same database of information about movie showings.

7.1 Participants

Twenty-three users (12 female, 11 male) accessed the systems via telephone in our lab. Most were undergraduate students from Carnegie Mellon University, resulting in a limited age range representation. None had any prior experience with either of the two movie systems or interfaces, and all users were native speakers of American English. About half the users had computer science and/or engineering (CSE) backgrounds, and similarly, about half reported that they did computer programming "fairly often" or "very frequently."

7.2 The Natural Language MovieLine

The NL MovieLine (NL-ML) was derived from the CMU Scheduler architecture [Eskenazi et al. 1999]. As in the Speech Graffiti system, Speech recognition is performed by Sphinx-II, speech synthesis is generated using Festival, and parsing is handled by Phoenix. To keep the nondialog components of the systems as similar as possible, the acoustic models used in the NL-ML for this experiment were the same as those used in the Speech Graffiti system; tests showed that they performed comparably to the most up-to-date Communicator-based acoustic models. The language models used by the NL-ML are class- and grammar-based and were designed specifically for this system.

WHAT MOVIE OR THEATER WOULD YOU LIKE TO HEAR ABOUT?
Squirrel Hill.
CINEMAGIC MANOR THEATRE AND CINEMAGIC SQUIRREL HILL ARE LOCATED IN SQUIRREL HILL. WHICH THEATER WERE YOU INTERESTED IN?
What is playing at the Cinemagic Squirrel Hill theater?
THERE ARE 6 MOVIES PLAYING AT CINEMAGIC SQUIRREL HILL. THEY ARE: MOONLIGHT MILE, RED DRAGON, SECRETARY, SWEET HOME ALABAMA, THE RING, AND WHITE OLEANDER. WHAT ELSE CAN I HELP YOU WITH?
Southland Nine cinema.
THERE ARE 8 MOVIES PLAYING AT CARMIKE SOUTHLAND 9. THEY ARE: MY BIG FAT GREEK WEDDING, ABANDON, KNOCKAROUND GUYS, RED DRAGON, SWEET HOME ALABAMA, THE RING, THE TRANSPORTER, AND WHITE OLEANDER. WHAT ELSE CAN I HELP YOU WITH?
What is the address of the theater?
THE CARMIKE SOUTHLAND 9 IS LOCATED AT: RTE. 51 S. - SOUTHLAND SHOPPING CENTER IN PLEASANT HILLS.
Showcase West.
THE SHOWCASE CINEMAS PITTSBURGH WEST IS LOCATED AT: PARK MANOR DR. IN PITTSBURGH.
What time is The Tuxedo playing at Showcase West?
THE TIMES FOR THE TUXEDO AT SHOWCASE CINEMAS PITTSBURGH WEST ARE 5:10 PM AND 10:00 PM. WHAT ELSE CAN I TELL YOU?

Fig. 5. Sample natural language MovieLine (NL-ML) dialog from the user study.

Figure 5 presents a sample NL-ML dialog from the user study, showing its natural language prompts and response patterns. The NL-ML permits both longer and shorter input, although this allows for the type of ambiguity demonstrated in the first user utterance of Figure 5 which requires a system clarification in order to determine whether the user wants to know about the Squirrel Hill Theater or the Squirrel Hill neighborhood. Recognition problems in the NL-ML are handled with a generalized response which does not allow partially-recognized input fragments to be retained.

7.3 Training

Users learned Speech Graffiti concepts prior to use during a brief, self-paced, Web-based tutorial session. Speech Graffiti training sessions were balanced between tutorials using examples from the MovieLine and tutorials using examples from a database that provided simulated airline flight information. Regardless of the training domain, most users spent ten to fifteen minutes on the Speech Graffiti tutorial.

A side effect of the Speech Graffiti training is that, in addition to teaching users the concepts of the language, it also familiarizes users with the more general task of speaking to a computer over the phone. To balance this effect for users of the natural language system, which is otherwise intended to be a walk-up-and-use interface, participants engaged in a brief natural language familiarization session. They were shown a Web page that provided a brief description of the system and a few examples of the types of things one could say to the system and were then asked to spend a few minutes experimenting with the actual system. To match the in-domain/out-of-domain variable used in the Speech Graffiti tutorials, half of the natural language familiarization sessions used the NL-MovieLine and half used MIT's Jupiter natural language

system for weather information [Zue et al. 2000]. Users typically spent about five minutes exploring the natural language systems during the familiarization session.

7.4 Tasks

Upon completion of the training session for a specific system, each user was asked to call that system and attempt a set of eight tasks (e.g., "list what's playing at the Squirrel Hill Theater," "find out & write down what the ratings are for the movies showing at the Oaks Theater"). Participant compensation included task completion bonuses to encourage users to perform each task in earnest. Regardless of which system they were using, all users were given the same set of eight tasks for their first interactions and a different set of eight tasks for their interactions with the second system. System presentation order was balanced.

7.5 Assessment

After interacting with a system, each participant completed a user satisfaction questionnaire scoring 34 subjective response items on a 7-point Likert scale. This questionnaire was based on the Subjective Assessment of Speech System Interfaces (SASSI) project [Hone and Graham 2001] which sorts a number of subjective user satisfaction statements (e.g., "I always knew what to say to the system" and "the system makes few errors") into six relevant factors: system response accuracy, habitability, cognitive demand, annoyance, likeability, and speed. User satisfaction scores were calculated for each factor and an overall score was calculated by averaging the responses to the appropriate component statements.[1] In addition to the Likert scale items, users were also asked a few direct comparison questions, such as "which of the two systems did you prefer?" For objective comparison of the two interfaces, we measured overall task completion, time- and turns-to-completion, and word- and understanding-error rates.

7.6 User Satisfaction

After using both systems, 17 out of the 23 subjects (74%) stated that they preferred the Speech Graffiti system to the natural language interface. Mean scores for subjective user satisfaction assessments were significantly higher for Speech Graffiti overall and in each of the six user satisfaction factors as shown in Figure 6 (by one-sided, paired t-tests: overall $t = 3.20$, $\mathrm{df} = 22$, $\mathrm{p} < .003$; system response accuracy $t = 3.36$, $\mathrm{df} = 22$, $\mathrm{p} < .002$; likeability $t = 2.62$, $\mathrm{df} = 22$, $\mathrm{p} < .008$; cognitive demand $t = 2.39$, $\mathrm{df} = 22$, $\mathrm{p} < .02$; annoyance $t = 1.94$, $\mathrm{df} = 22$, $\mathrm{p} < .04$; habitability $t = 2.51$, $\mathrm{df} = 22$, $\mathrm{p} < 0.01$; speed $t = 5.74$, $\mathrm{df} = 22$, $\mathrm{p} < .001$).

All of the mean SG-ML scores except for annoyance and habitability were positive (i.e., >4), while the NL-ML did not generate positive mean ratings in

[1]Some component statements are reversal items whose values were inverted for analysis so that high scores in all categories are considered good.

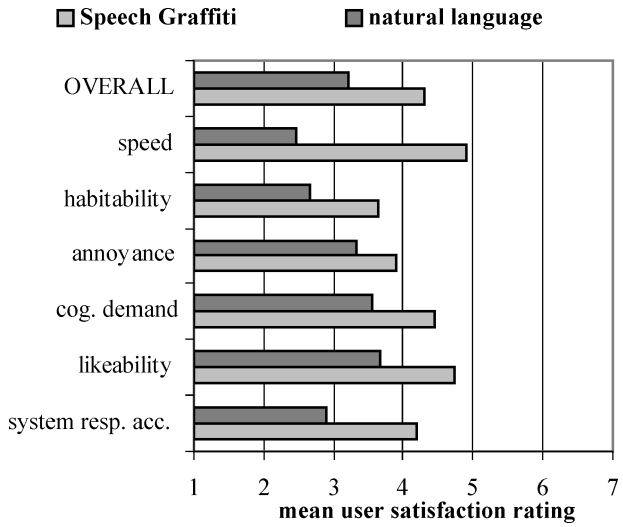

Fig. 6. Comparison of user satisfaction ratings.

any category. The SG-ML's lowest user satisfaction rating was in the habitability category which involves factors related to knowing what to say to the system, a predictable issue with a subset language interface. For individual users, all those, and only those, who stated they preferred the NL-ML to the SG-ML gave the natural language system higher overall subjective ratings. Participants confirmed our suspicions that programmers and users with CSE backgrounds might be more amenable to the Speech Graffiti approach. In all categories, CSE/programmer subjects gave the SG-ML higher user satisfaction ratings, although the differences were significant in fewer than half of the categories.

We also compared users' subjective assessments of the Speech Graffiti MovieLine based on whether they had used the tutorial system that used the SG MovieLine or the SG FlightLine and found that training domain had a negligible effect on satisfaction ratings.

7.7 Objective Assessments

7.7.1 *Task Completion.* Task completion did not differ significantly for the two interfaces. In total, just over two thirds of the tasks were successfully completed with each system: 67.4% for the NL-ML and 67.9% for the SG-ML. Participants completed on average 5.2 tasks with the NL-ML and 5.4 tasks with the SG-ML. As with user satisfaction, users with CSE or programming backgrounds generally completed more tasks in the SG-ML system than non-CSE or programming users, but the difference was not statistically significant. Training domain had no significant effect on task completion for either system: users who trained on the SG-ML completed an average of 5.45 tasks correctly, while users who trained on the FlightLine system completed an average of 5.42 tasks correctly. (Similarly, the NL-ML familiarization system variable had no significant effect on users' subjective assessments of or task completion rates for the

NL-ML interface.) Considered along with the lack of difference in user satisfaction ratings, this indicates that users are generally able to transfer concepts from one Speech Graffiti application to another, and that although application-specific training systems might be preferred if available, they are not absolutely necessary for acceptable performance.

7.7.2 *Time-to-Completion.* To account for incomplete tasks when comparing the interfaces, we ordered the task completion measures (times or turn counts) for each system, leaving all incompleted tasks at the end of the list as if they had been completed in "infinite time," and compared the medians.

For completed tasks, the average time users spent on each SG-ML task was lower than for the NL-ML system, though not significantly: 67.9 versus 71.3 seconds. Accounting for incomplete tasks, the SG-ML performed better than the NL-ML with a median time of 81.5 seconds compared to 103 seconds.

7.7.3 *Turns-to-Completion.* For completed tasks, the average number of turns users took for each SG-ML task was significantly higher than for the NL-ML system: 8.2 versus 3.8 ($F = 26.4$, $p < .01$). Including incomplete tasks, the median SG-ML turns-to-completion rate was twice that of the NL-ML: 10 versus 5. This reflects the short-turn, one-concept-at-a-time style adopted by most users in the SG-ML which flexibly supports both short and long turns.

7.7.4 *Word-Error Rate.* The SG-ML had an overall word-error rate (WER) of 35.1%, compared to 51.2% for the NL-ML. When calculated for each user, WER ranged from 7.8% to 71.2% (mean 35.0%, median 30.0%) for the SG-ML and from 31.2% to 78.6% (mean 50.3%, median 48.9%) for the NL-ML. The difference in error rates can be partially attributed to a difference in out-of-vocabulary rates. Utterances containing out-of-vocabulary words occurred more than twice as often in the NL-ML system than in the SG-ML system. This demonstrates a difficulty with natural language systems: it is difficult to design grammars that will fully cover user input, and it is difficult to get users to understand what is accepted by the grammar and what is not.

The six users with the highest SG-ML WER were the same ones who preferred the NL-ML system, and four of them were also the only users in the study whose NL-ML error rate was lower than their SG-ML error rate. This suggests, not surprisingly, that WER is strongly related to user preference.

To further explore this correlation, we plotted WER against users' overall subjective assessments of each system, with the results shown in Figure 7. There is a significant, moderate correlation between WER and user satisfaction for the Speech Graffiti interface ($r = -.66$, $p < .01$), but no similar correlation for the NL-ML system ($r = .26$).

7.7.5 *Understanding-Error Rate.* Word-error rate may not be the most useful measure of system performance for many spoken dialog systems. Because of grammar redundancies, systems are often able to understand and process an utterance correctly even when some individual words are misrecognized. Understanding-Error Rate (UER) may therefore provide a more reliable idea of the error rate that a user actually experiences. For this analysis, the

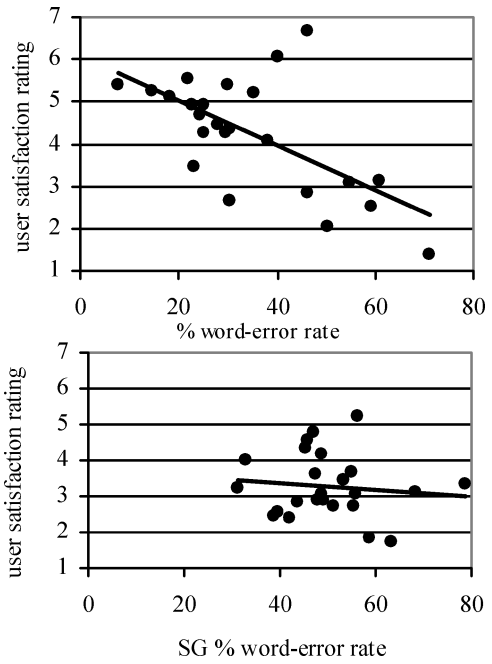

Fig. 7. Word-error rate vs. overall user satisfaction for Speech Graffiti and natural language MovieLines.

understanding-error rates were hand-scored, and as such represent an approximation of actual UER. For both systems, we calculated UER based on an entire user utterance rather than individual concepts in that utterance. SG-ML UER for each user ranged from 2.9% to 65.5% (mean 26.6%, median 21.1%). The average change per user from WER to UER for the SG-ML interface was –29.2%. The NL-ML understanding-error rates differed little from the NL-ML WER rates. UER per user ranged from 31.4% to 80.0% (mean 50.7%, median 48.5%). The average change per user from NL-ML WER was +0.8%.

7.7.6 *Grammaticality.* Independently of its performance compared to the natural language system, we were interested in assessing the *habitability* of Speech Graffiti: how easy was it for users to speak within the Speech Graffiti grammar? Overall, 82% of user utterances were fully Speech Graffiti-grammatical. For individual users, grammaticality ranged from 41.1% to 98.6%, with a mean of 80.5% and a median of 87.4%. These averages are quite high, indicating that most users were able to learn and use Speech Graffiti reasonably well. No significant effects on Speech Graffiti-grammaticality were found due to differences in CSE background, programming experience, training supervision, or training domain.

The lowest individual grammaticality scores belonged to four of the six participants who preferred the natural language MovieLine interface to the Speech Graffiti one which suggests that proficiency with the language is very important for its acceptance. Indeed, we found a moderate, significant correlation

between grammaticality and user satisfaction for Speech Graffiti ($r = .37$, $p < .003$,) (a cursory analysis found no similar correlation for the natural language interface).

Users' grammaticality tended to increase over time. For each participant, we compared the grammaticality of utterances from the first half of their session with that of utterances in the second half. All but four participants increased their grammaticality in the second half of their Speech Graffiti session with an average relative improvement of 12.4%. A REML analysis showed this difference to be significant ($F = 7.54$, $p < .02$). Interestingly, only one of the users who exhibited a decrease in grammaticality over time was from the group that preferred the natural language interface. However, although members of that group did tend to increase their grammaticality later in their interactions, none of their second-half grammaticality scores were above 80%. A more thorough longitudinal analysis over multiple sessions is needed to further assess changes in grammaticality over time.

7.8 Discussion

It could be argued that the good performance of Speech Graffiti as compared to the natural language interface in this study may be the result of either the more intensive Speech Graffiti tutorial given to users or the lower word-error rate of the system.

To address the tutorial issue, we note that our experimental treatment reflects the underlying design assumptions of each system: a tutorial session is necessary to teach users the concepts of Speech Graffiti, whereas one of the purported advantages of natural language systems is that they should be so natural as to require no special training. In our case, the Speech Graffiti training may have provided a learning benefit; one could imagine that we might find different results overall if the experiment was revised to include a time constraint so that time spent on the tutorial session detracted from time available to work on actual tasks. One should note though that Speech Graffiti training need only be done once and is therefore amortized across future uses of all SG applications.

As for the word-error differences, this is one of the foundations of our argument. When the word-error rate of natural language systems can be reduced considerably, such systems become truly feasible options for speech interaction with computers. The difference in out-of-vocabulary rates between the two systems hints at the difficulty of solving the OOV issue for natural language interfaces. In the meantime, we have demonstrated that a simpler, more restricted language system can compare favorably to a more natural, yet more errorful, interface.

8. FUTURE WORK

The results of our user studies have shown that, compared to users of a particular natural language speech interface, Speech Graffiti users had higher levels of user satisfaction, lower task completion times, and similar task completion

rates, at a lower overall system development cost. We also found that task success and user satisfaction with Speech Graffiti were significantly correlated with grammaticality. This indicates that it is very important to help users speak within the grammatical bounds of voice user interfaces (particularly subset language ones). However, even after training, some users had difficulty speaking within a restricted grammar. In our comparative experiment, 6 of 23 participants preferred the natural language system. The experience of these 6 users provides a picture of frustrating interaction. In the Speech Graffiti system, they accounted for the highest word- and understanding-error rates, the lowest task completion rates, and the four lowest grammaticality rates. (These users also accounted for the four lowest task completion rates for the natural language system which suggests that working with speech interfaces in general may pose problems for some users.) One defining characteristic of these 6 participants was that all but 1 of them belonged to the group of study participants without computer programming backgrounds.

Based on our results to date, we plan to refine the Speech Graffiti system to improve the interaction experience and efficiency for all users. However, we will specifically consider the experience of the 6 NL-ML-preferring participants in our improvements. One interesting feature of the user input collected in this study was that when users spoke to the natural language system, their inputs distilled into nearly 600 syntactic patterns. However, when users were ungrammatical in the Speech Graffiti interface and spoke natural language to it rather than using the restricted syntax, their input distilled into less than 100 patterns. Noticeably absent from this latter set were conversational, nontopic items such as “could you please tell me” or “I would like to know about.” This indicates that simply knowing that one is speaking to a restricted-language system is enough to affect the type of input a user provides to a system.

We plan to exploit this observation in our future work in the information access domain by implementing a system of *intelligent shaping help and adaptivity*. The goal is to create a system which can understand input that is less than conversational but more accepting than canonical Speech Graffiti. Since interaction at the Speech Graffiti level is expected to be less error-prone and more efficient, system prompts can then be used to shape user input to match the more efficient Speech Graffiti style. We propose that the implementation of such a shaping scheme can virtually eliminate the pre-use training time that is currently required to learn the Speech Graffiti system. This system should benefit both long-term users, who will learn strategies for making their interactions more efficient, and one-time users, who should be able to complete tasks using the expanded language without necessarily having to learn the Speech Graffiti style.

In addition to helping first-time users learn the basics of the system, we also plan to implement adaptive strategies that can help novice users improve their skills and have even more efficient interactions. Such strategies might include making suggestions about more advanced keyword use such as using

the **<slot> is anything** construction to clear individual slots, or combining navigation keywords with integers to customize the length of query result presentation. The system's own interaction style could change for experts as well. As part of our shaping strategy, future system confirmations will echo the **<slot> is <value>** format of Speech Graffiti input in order to be more lexically entraining than repeating just the value. However, once a user has achieved proficiency with the **<slot> is <value>** input format, confirmations could switch back to value-only to make the interactions faster and less repetitive.

Since our results reported here were generated by a fairly specific user population (undergraduate students), further evaluations of Speech Graffiti will focus on users who are older than college age and who do not have experience with computer programming. We also plan to conduct a longitudinal study in which participants interact with Speech Graffiti applications several times over a period of a few months. We expect this to help us better understand the learning curve for the Speech Graffiti language.

Another potential area for future work with the Speech Graffiti approach is in the interactive guidance domain where the system leads the user through a series of steps to complete a task such as repairing a mechanical part or baking a cake. Another area would be transaction systems such as those that would allow users to make restaurant reservations or purchase movie tickets. Even in the information access domain, Speech Graffiti functionality could be expanded to include the addition, modification, and deletion of database records.

9. CONCLUSION

We have found Speech Graffiti to be a promising step in increasing the efficiency of human-machine speech interaction. Our system was designed to make recognition more reliable by regularizing the interaction style, and the lower word- and understanding-error rates generated in our comparison study verify this approach. User study results demonstrated that speakers can use Speech Graffiti well enough to complete tasks successfully and prefer the system to a less efficient natural language interface. However, our studies also demonstrated that learning and using Speech Graffiti successfully can be challenging for some users. Our future research directions are aimed at reducing this challenge, opening the possibility for Speech Graffiti–like systems to be integrated into a variety of publicly-accessible applications. With this aim, our future evaluations of the system will focus on a more representative adult population.

Information access applications provide perhaps the greatest opportunity for Speech Graffiti systems. Requiring only a telephone for access, they generally access text databases which easily support mappings to Speech Graffiti slots and values. Transaction systems would be the natural next extension to such systems. The implementation of other types of systems such as gadget control and interactive guidance introduces an interesting area of research questions on the idea of skill and learning transference not just across domains, but across forms.

As a modality, speech interaction is celebrated for its accessibility, portability, and ease of use. It is usually an extremely efficient mode of communication for human-human interaction. However, the current state-of-the-art in speech and language processing and artificial intelligence does not allow for equally efficient human-computer speech communication. Speech Graffiti offers a step towards improving such interaction.

APPENDIXES

Appendix A.

This Phoenix grammar [Ward 1990] describes the Speech Graffiti language using two slots (**`title`** and **`start time`**) from the MovieLine application.

```
## ---------- valid Speech Graffiti utterance ----------
## ---------- domain-independent except as noted -------
[Utt]
   ( +[PHRASES] *[KeyPhrase] )
   ( [KeyPhrase] )
   ( [NavPhrase] )
;
[PHRASES]
   ( [MOVIE=SLOT] [MOVIE=VALUE] ) ## domain-specific
   ( [SHOWTIME=SLOT] [TIME=VALUE] ) ## domain-specific
   ( [WHAT] SLOTS )
   ( SLOTS *anything )
   ( *SLOTS options )
   ( ERASER )
SLOTS
   ( [SHOWTIME=SLOT] )
   ( [MOVIE=SLOT] )
ERASER
   ( start=over )
   ( scratch=that )
;
[WHAT]
   ( what *IS-ARE )
   ( requesting )
IS-ARE
   ( is )
   ( are )
;
[NavPhrase]
   ( more )
   ( previous *[Hour] ) # [Hour] is a convenient small integer type
   ( next *[Hour] )
   ( first *[Hour] )
   ( last *[Hour] )
;
[KeyPhrase]
   ( go=ahead )
   ( repeat )
   ( goodbye )
   ( help )
   ( where=was=i )
```

```
   ( where=were=we )
   ( where=am=i )
   ( where=are=we )
;
## ------------------- example slots ---------------
## -- lexical items are domain-specific, but slot syntax is standard
[MOVIE=SLOT]
   *the MOVIE-SYNONYMS-SG *is
   *the MOVIE-SYNONYMS-PL *are
MOVIE-SYNOMYNS-SG
   ( movie )
   ( title )
MOVIE-SYNONYMS-PL
   ( movies )
   ( titles )
;
[SHOWTIME=SLOT]
   *the START-SHOW time *is
   *the START-SHOW times *are
START-SHOW
   ( start )
   ( show )
   ( starting )
;
## ------------------- example values ---------------
[MOVIE=VALUE] ## domain-specific
   ( [lost=in=translation] )
   ( [the=matrix=revolutions] )
;
[lost=in=translation]
   ( lost=in=translation )
;
[the=matrix=revolutions]
   ( the=matrix=revolutions )
   ( matrix=revolutions )
;
[Time=Constraint]              ## some value types (time, date,
   ( INTERVAL )                ## numbers, etc.) have standard grammars
   ( SEMI-INTERVAL [Time] )    ## which can be used across Speech Graffiti
   ( *at [Time] )              ## applications
INTERVAL
   ( between [Time] and [Time] )
   ( after [Time] before [Time] )
SEMI-INTERVAL
   ( before )
   ( earlier=than )
   ( after )
   ( later=than )
;
[Time]
   ( [Hour] *o'clock *AM-PM )
   ( [Hour] [Minute] *AM-PM )
   ( noon )
   ( midnight )
AM-PM
```

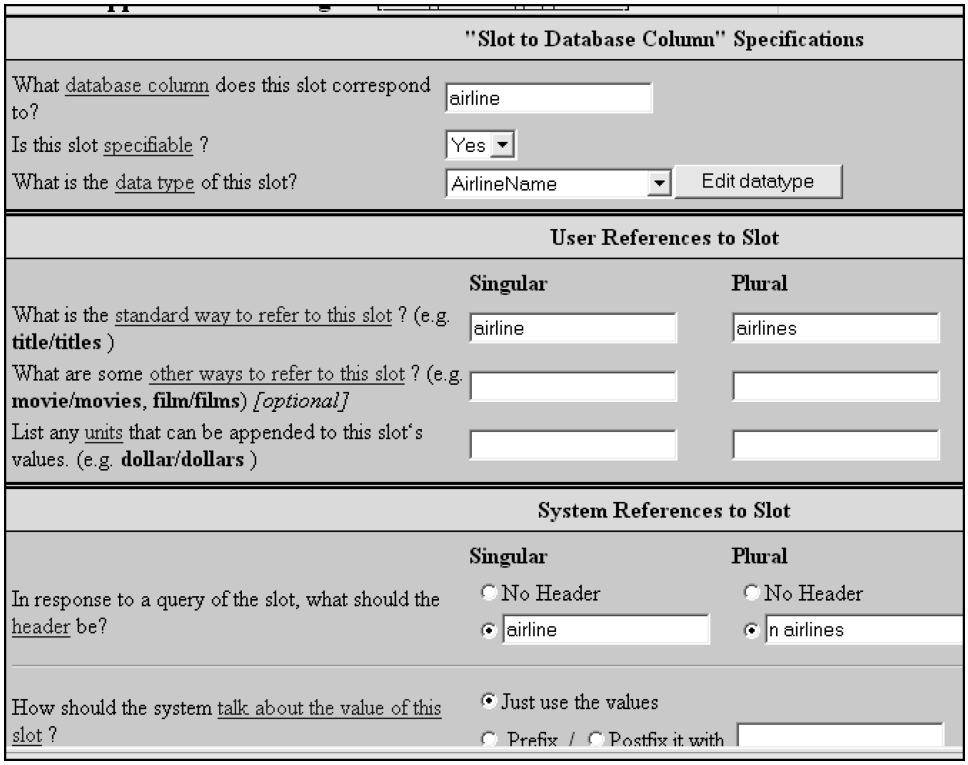

Fig. 8. A portion of the Speech Graffiti Web Application Generator for an airline schedule application.

```
   ( a=m )
   ( p=m )
;
[Hour]
   ( one )
   ( two ) # etc.
;
[Minute]
   ( ten )
   ( fifteen ) # etc.
;
```

Appendix B.

Figure 8 shows a screenshot of a portion of the Speech Graffiti Web Application Generator for an airline schedule application. The code fragment following it shows an extended piece of the .xml file produced by the generator.

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE Application *(View Source for full doctype...)*>
<Application name="**expflight**" preferred_page_size="**3**" what_required="**false**">
  <Database name="**usiroutes**" field_list="**airline, flight, depapt, arrapt, date,**

deptime, arrtime, connect, depgate, arrgate" query_tables="**allroutes**" />
<Slot name="**airline**" column="**airline**" now_what="**airline**"
single_result_header="**airline**" multi_result_header="**airlines**"
slot_now_what="**airline can be airtran, american,** continental, **delta, northwest, united, us air, vanguard or you can ask what is the airline**">
<slot_say_as>**airline is**</slot_say_as>
<slot_say_as>**airlines are**</slot_say_as>
<value_reference constraint="**true**">
<enum_reference type="**AirlineName**" />
</value_reference>
</Slot>
<value type="**AirlineName**">
<top name="**northwest**">
<value_say_as>**northwest**</value_say_as>
<value_say_as>**northwest airlines**</value_say_as>
</top>
</value>
<Slot name="**flight**" column="**flight**" now_what="**flight**"
single_result_header="**flight**" multi_ result_header="**flights**" slot_now_what="**flight number can be the number of a scheduled flight or you can ask what is the flight number**" sg_confirm="**flight %s,**" pl_confirm="**flight %s,**"
sg_in_result_string="**flight %s,**" pl_in_result_string="**flight %s,**">
<slot_say_as>**flight is**</slot_say_as>
<slot_say_as>**flights are**</slot_say_as>
<slot_say_as>**flight number is**</slot_say_as>
<slot_say_as>**flight numbers are**</slot_say_as>
<value_reference constraint="**false**">
<basic_reference type="**NumberString**" />
</value_reference>
</Slot>
</Application>

REFERENCES

BABER, C. 1991. Human factors aspects of automatic speech recognition in control room environments. In *Proceedings of IEEE Colloquium on Systems and Applications of Man-Machine Interaction Using Speech I/O*. 10/1–10/3.

BLACK, A. AND LENZO, K. 2000. Limited domain synthesis. In *Proceedings of the 6th International Conference on Spoken Language Processing (ISCLP'00)*. Beijing, China. 411–414.

BLACK, A., TAYLOR, P., AND CALEY, R. 1998. The festival speech synthesis system. Available at http://www.cstr.ed.ac.uk/projects/festival.html.

BLICKENSTORFER, C. H. 1995. Graffiti: Wow!!!! *Pen Comput. Mag.*, (Jan:30-31).

CLARKSON, P. AND ROSENFELD, R. 1997. Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of Eurospeech*. Rhodes, Greece. 2707–2710.

ESKENAZI, M., RUDNICKY, A., GREGORY, K., CONSTANTINIDES, P., BRENNAN, R., BENNETT, C., AND ALLEN, J. 1999. Data collection and processing in the Carnegie Mellon Communicator. In *Proceedings of Eurospeech*. 2695–2698.

GLASS, J. 1999. Challenges for spoken dialogue systems. In *Proceedings of IEEE Automatic Speech Recognition (ASRU) Workshop*. Keystone, CO.

GRICE, H. 1975. Logic and conversation. *Syntax and Semantics, Vol. 3: Speech Acts*. Academic Press, New York, NY. 41–58.

GUZMAN, S., WARREN, R., AHLENIUS, M., AND NEVES, D. 2001. Determining a set of acoustically discriminable, intuitive command words. In *Proceedings of AVIOS Speech Technology Symposium (AVIOS'01)*. San Jose, CA. 241–250.

HARRIS, T. K. AND ROSENFELD, R. A. 2004. A universal speech interface for appliances. In *Proceedings of the 8th International Conference on Spoken Language Processing (ICSLP'04)*. Jeju Island, South Korea.

HONE, K. AND GRAHAM, R. 2001. Subjective assessment of speech-system interface usability. In *Proceedings of Eurospeech*, Aalborg, Denmark.

HUANG, D., ALLEVA, F., HON, H. W., HWANG, M. Y., LEE, K. F., AND ROSENFELD, R. 1993. The Sphinx-II speech recognition system: An overview. *Comput. Speech Lang. 7*, 2, 137–148.

PERLMAN, G. 1984. Natural artificial languages: Low level processes. *Int. J. Man-Machine Studies 20*, 373–419.

SHNEIDERMAN, B. 1980. *Software Psychology: Human Factors in Computer and Information Systems*. Winthrop Inc, Cambridge MA.

SHRIVER, S. AND ROSENFELD, R. 2002. Keywords for a universal speech interface. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. Minneapolis, MN. 726–727.

SIDNER, C. AND FORLINES, C. 2002. Subset languages for conversing with collaborative interface agents. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP'02)*. Denver, CO. 281–284.

TELEPHONE SPEECH STANDARDS COMMITTEE. 2000. Universal commands for telephony-based spoken language systems. *SIGCHI Bull. 32*, 2, 25–29.

TOMKO, S. 2004. Speech Graffiti: Assessing the user experience. Carnegie Mellon University. LTI Tech Rep. CMU-LTI-04-185, Available at: www.cs.cmu.edu/∼stef/papers/mthesis.ps

TOTH, A., HARRIS, T., SANDERS, J., SHRIVER, S., AND ROSENFELD, R. 2002. Towards every-citizen's speech interface: An application generator for speech interfaces to databases. In *Proceedings of the 7th International Conference on Spoken Language Processing*. Denver, CO. 1497–1500.

WARD, W. 1990. The CMU air travel information service: Understanding spontaneous speech. In *Proceedings of the DARPA Speech and Language Workshop*. Hidden Valley, PA. 127–129.

ZOLTAN-FORD, E. 1991. How to get people to say and type what computers can understand. *Int. J. Man-Machine Studies 34*, 527–547.

ZUE, V., SENEFF, S., GLASS, J. R., POLIFRONI, J., PAO, C., HAZEN, T. J., AND HETHERINGTON, L. 2000. JUPITER: A telephone-based conversational interface for weather information. *IEEE Trans. Speech Audio Process. 8*, 1, 85–96.

Received July 2004; accepted March 2005 by Marcello Federico