Towards Best Practices for Speech User Interface Design

Bernhard Suhm
Speech and Language Processing
BBN Technologies, Cambridge (MA)
bsuhm@bbn.com

Abstract
Designing speech interfaces is difficult. Research on spoken language systems and commercial application development has created a body of speech interface design knowledge. However, this knowledge is not easily accessible to practitioners. Few experts understand both speech recognition and human factors well enough to avoid the pitfalls of speech interface design. To facilitate the design of better speech interfaces, this paper presents a methodology to compile design guidelines for various classes of speech interfaces. Such guidelines enable practitioners to employ discount usability engineering methods to speech interfaces, including obtaining guidance during early stages of design, and heuristic evaluation. To illustrate our methodology, we apply it to generate a short list of ten guidelines for telephone spoken dialog systems. We demonstrate the usefulness of the guidelines with examples from our consulting practice, applying each guideline to improve poorly designed prompts. We believe this methodology can facilitate compiling the growing body of design knowledge to best practices for important classes of speech interfaces.

1. Introduction
Speech user interface (SUI) design is difficult, despite decades of research on spoken language systems and years of commercial application development. One reason is the fact that knowledge relevant to speech interface design is not easily accessible to practitioners. Most published guidelines for speech interface design either cover the basics (e.g., [1-3]), or are in book form and thus not quickly accessible (e.g., telephone speech applications [4], dictation [5]). Consequently, very few experts understand both speech recognition and human factors well enough to avoid the pitfalls of SUI design and development. For lack of access to experts or for cost reasons, many projects rely on in-house developers who may have little or no experience in SUI design.

This paper aims to help practitioners design better speech recognition applications. We begin by reviewing our framework of SUI design [6] that is the foundation for this work. This framework allowed us to compile a broad range of knowledge relevant to SUI design into a database of good designs, based on a survey of relevant literature. Using this database, we can compile lists of design guidelines in two steps. First, the framework’s database is queried to obtain design solutions that apply to a specific class of speech interfaces. Second, expert judgment is employed to reduce the list to a short list of guidelines. We applied this methodology to generate ten guidelines for (telephone) spoken dialog design. Such guidelines can guide inexperienced designers to avoid common pitfalls in early stages of design. Furthermore, practitioners can use the guidelines to conduct heuristic evaluations of deployed speech recognition applications. We demonstrate the validity of the guidelines with examples from our telephone voice interface consulting practice.

2. A Framework for SUI Design
The framework for SUI design presented in [6] helps designers leverage known solutions to overcome limitations of speech and design better speech interfaces. Known solutions and best practices are organized based on two key dimensions of speech interface design: limitations of speech as an interface modality on the one hand, and application requirements on the other hand. Below we review the taxonomy of limitations of speech interfaces that forms the foundation of the framework. The second subsection describes the instantiation of our “framework” as a database of design “solutions” and best practices, which was compiled based on a survey of relevant literature and expert knowledge.

<table>
<thead>
<tr>
<th>Limitation Category</th>
<th>Definition</th>
<th>Specific Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech Recognizers</td>
<td>Limitations of (current) speech recognizers</td>
<td>Errors, finite vocabulary, language model, acoustic models, quality of input signal</td>
</tr>
<tr>
<td>Spoken Language</td>
<td>Limitations inherent in speech and spoken language</td>
<td>“Spontaneous” speech, public nature, turn-taking protocol, ambiguity, anthropomorphism, limited expressive power, sequential</td>
</tr>
<tr>
<td>Environment</td>
<td>Influences from the user’s environment</td>
<td>Noise, multiple voices, interruptions</td>
</tr>
<tr>
<td>Human Cognition</td>
<td>Properties of the human cognitive system</td>
<td>Low persistence, competition with verbal processing, limited working memory capacity</td>
</tr>
<tr>
<td>User</td>
<td>Differences and preferences between users</td>
<td>Prior beliefs and expectations, motivation, limited task knowledge, limited speech competence</td>
</tr>
<tr>
<td>Hardware</td>
<td>Properties of the hardware</td>
<td>Channel bandwidth, microphones, CPU</td>
</tr>
</tbody>
</table>

Table 1: Taxonomy of speech interface limitations
2.1. Limitations of Speech User Interfaces (SUIs)

Extending previous taxonomies of speech [7, 8], our taxonomy distinguishes six broad categories of limitations: speech recognition, spoken language, environment, human cognition, user, and hardware. Table 1 shows our limitation categories, their definition, and specific examples for each limitation. While the speech recognition community should be very familiar with the limitations of speech recognizers and the hardware used to deploy speech applications, this taxonomy makes obvious that there is much more to SUI design than the speech recognizer.

2.2. The Solution Database

To organize the existing body of knowledge relevant to SUI design, which is spread between vendors, consultants, speech recognition research and usability research, we developed a database for SUI design solutions and best practices. Realizing that (most) speech interface design problems are rooted in some limitation of speech, we organized this knowledge as "solutions" to specific design problems arising from limitations of speech. One index to the database is therefore the taxonomy of speech limitations described in the previous section. To be able to relate solutions to specific applications, we developed a set of solution attributes as orthogonal indices to the database.

Table 2 shows sample content of the database for the first few solutions to the limitation of "recognition errors". Based on our preliminary survey, the database currently contains 140 specific design problems and solutions. Due to space limitations, only the most important solution attributes are shown in Table 2: the input/output modalities required by a design or solution, the domain, the interaction style (dialog-oriented, interactive, or non-interactive), the solution type (recognition algorithm, recognizer configuration, interaction design, and user training), and the source (reference that describes the solution).

3. Compiling Design Guidelines

One application of the framework described in the previous section is the generation of lists of design guidelines for broad classes of SUIs. To illustrate our method for compiling design guidelines, this section describes how we queried the solution database to generate a list of guidelines for telephone dialog design. The framework can be applied in similar ways to generate design guidelines for other classes of SUI, such as desktop or handheld applications.

We first queried the solution database to obtain a complete list of design solutions and best practices that apply to telephone dialog applications. The query specified such applications in terms of our solution attributes as follows: Telephone dialog systems are characterized by their (predominantly) speech-only interface, constrained domains, and a dialog-oriented interaction style. The query resulted in a quite long list of design solutions, which represent candidates for design guidelines.

We paired down the list by eliminating all solutions that apply more generally than just speech-only dialog applications, by merging solutions that apply to multiple design problems, and by reformulating them as design guideline. Finally, to arrive at a short list of ten guidelines, we used expert judgment to select the ten most important ones. Table 3 below shows the guidelines, a specific design solution as an example for how to apply the guideline, and how it relates to limitations of speech.

Such lists of guidelines enable designers to apply discount usability engineering methods to SUIs, such as obtaining guidance in early stages of design, and conducting heuristic (usability) evaluations. Best practices emerge over time as widely accepted guidelines. The remainder of this paper illustrates the usefulness of our guidelines by applying them to heuristic evaluation of (telephone) spoken dialogs.

4. Heuristic Evaluation of SUIs

4.1. Heuristic Evaluation

Heuristic evaluation is a popular discount usability method. In a heuristic evaluation, an interface design is critiqued based on a checklist of design guidelines or heuristics. For lack of a suitable checklist, heuristic evaluations of speech interfaces have been limited to the few experts who understand both speech recognition and human factors really well. Nielsen’s ten usability heuristics [11] are too general to be useful for avoiding the specific pitfalls of speech interfaces design; and the few published guidelines for speech interface design are not compiled in a format that can be used in a heuristic evaluation (e.g., [4]).

<table>
<thead>
<tr>
<th>Solutions (for recognition errors)</th>
<th>Modalities</th>
<th>Domain</th>
<th>Interaction Style</th>
<th>Solution Type</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Careful configuration of the recognizer (vocabulary, grammars, …)</td>
<td>GUI, buttons, keyboard</td>
<td>Interactive</td>
<td>User training</td>
<td>Interactive Design</td>
<td>[10]</td>
</tr>
<tr>
<td>Design prompts to elicit brief responses</td>
<td>Interactive</td>
<td>User training</td>
<td>[5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adopt speaking style that minimizes error</td>
<td>Interactive</td>
<td>User training</td>
<td>[9]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adopt work style that is optimized for error correction</td>
<td>Interactive</td>
<td>User training</td>
<td>[5]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Sample from the solution database, showing some solutions to the limitation (recognition) errors
4.2. Heuristic Evaluation of Telephone Prompts and Dialogs: Case studies applying our Ten Guidelines

To illustrate our guidelines, below we apply them to conduct heuristic evaluations of telephone dialog systems. Using examples from our experience in deployed commercial call center Interactive Voice Response systems (IVRs), we apply our guidelines to improve specific prompts.

1. **Keep it Simple**

Due to the limited capacity of working memory and the sequential and non-persistent nature of speech, the dialog must be kept simple. For example, it is widely accepted to limit spoken menus or lists to 3-7 options. If the user can barge in, up to 9 items are acceptable.

**Problematic prompt:** You can say voicemail information, coverage area, credit for a dropped call, problem placing or receiving calls, handset problems, questions about 3G, or other options. To hear this list again, say "repeat". For more information, say "help".

**What's the problem:** Too many options (10) are offered.

**Improved Prompt:** You can say "credit for dropped call", "handset problems", or "other options". (send other options straight to agent)

2. **Carefully control the amount of spoken output**

The amount of spoken output to the user must be carefully controlled, for the same limitation that motivated guideline #1. Many current telephone IVRs violate this guideline by opening the dialog with lengthy instructions, including the bad, yet common practice of referring callers to the company’s website. Research shows that callers “zone out” after 10 seconds of introductory verbiage.

**Guideline #2** (in conjunction with guideline #1) also suggests that answering frequently asked questions does not lend itself to telephone dialogs, because caller simply are unable to absorb lengthy instructions over the phone.

3. **Word options the way users think**

Wording menu options and information readouts is notoriously difficult. Too easily, call center jargon creeps in, options are confusing to the caller due to overlap between them, or due to the ambiguity that’s inherent in spoken language.

**Problematic prompt:** For your account balance or past or future payments say "account information". If you want to make a payment say "payment information". If you're calling about your bill or statement, say "billing information". Or you can ask for "other options.". So what'll be?

**What’s the problem:** The distinction between payment and billing information is unclear and confusing.

**Improved Prompt:** Please tell me what you'd like to do: check your balance, make a payment, or discuss a billing question. For other requests say "other."

4. **Minimize acoustic confusability of vocabulary**

While obvious to anyone with some knowledge about speech recognition, this guideline is still frequently violated.

**Problematic prompt:** Say "one referral number" or "multiple referral numbers".

**What’s the problem:** Large overlap between options offered – the distinction is between "one" and "multiple".

**Improved Prompt:** Do you want to specify multiple referral numbers?

5. **Provide carefully designed feedback**

The importance of feedback is a well-known guideline for interface design in general. For speech interfaces, this task is particularly difficult because the amount of output needs to be controlled (guideline #2).

6. **Abide by natural turn-taking protocol**

Turn-taking in human conversation is determined by certain rules; for example, that interrupting someone is considered rude. Due to the anthropomorphism that speech interfaces elicit, users generally abide by the same turn-taking protocol when interacting with dialog applications. Therefore, prompts should.

<table>
<thead>
<tr>
<th>#</th>
<th>Guideline</th>
<th>Example specific design solution</th>
<th>(corresponding) Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Keep it simple</td>
<td>max. 5 items in non-interruptable lists (or menus), max. 9 items in interruptable ones</td>
<td>Limited capacity of working memory (human cognition)</td>
</tr>
<tr>
<td>2</td>
<td>Carefully control the amount of spoken output</td>
<td>Keep prompts short, especially opening instructions.</td>
<td>Sequential nature of speech (spoken language)</td>
</tr>
<tr>
<td>3</td>
<td>Word options the way users think</td>
<td>Word menu options such that they are clearly distinguished in the users’ mind</td>
<td>Ambiguity (spoken language)</td>
</tr>
<tr>
<td>4</td>
<td>Minimize acoustic confusability of vocabulary</td>
<td>Minimize the number of shared syllables between prompt options</td>
<td>Acoustic models (recognizer)</td>
</tr>
<tr>
<td>5</td>
<td>Provide carefully designed feedback</td>
<td>(Almost) never verbatim feedback - express in terms of actions instead</td>
<td>N/A (general UI design principle)</td>
</tr>
<tr>
<td>6</td>
<td>Abide by natural turn-taking protocol</td>
<td>Design prompts that encourage natural turn-taking (instead of relying on barge-in)</td>
<td>Turn-taking (spoken language)</td>
</tr>
<tr>
<td>7</td>
<td>Coach a little at a time</td>
<td>Use examples in error/timeout reprompt, especially after open-ended prompts</td>
<td>Spontaneous speech (spoken language)</td>
</tr>
<tr>
<td>8</td>
<td>Offer alternative input modalities</td>
<td>Offer touch-tone keypad after errors, or for input that’s sensitive to privacy</td>
<td>Repeated errors (recognizers), public spoken language</td>
</tr>
<tr>
<td>9</td>
<td>Yes/no queries can be very robust</td>
<td>Employ yes/no queries to stabilize dialog after errors or ambiguous responses</td>
<td>Errors (recognizers), ambiguity (spoken language)</td>
</tr>
<tr>
<td>10</td>
<td>Carefully select the appropriate persona</td>
<td>Professional applications should employ professional personas</td>
<td>Anthropomorphism (spoken language)</td>
</tr>
</tbody>
</table>

Table 3: Ten Guidelines of Telephone Spoken Dialog Design
abide by the same turn-taking rules that apply to human conversation. Vice versa, we cannot expect callers to barge into long prompts, because most callers will listen to the complete prompt following turn-taking courtesy.

**Problematic prompt:** Feel free to interrupt me at any time. You can say something like ... (15 seconds of examples) So, what will it be?

**What’s the problem:** Most callers don’t intuitively interrupt long prompts, even if they are encouraged to do so.

**Improved Prompt:** Tell me, briefly, the reason for your call. (and provide examples only in the reprompt, upon timeout or recognition error).

7. **Coach a little at a time**

Coaching a little at a time is a well-known error recovery technique in spoken dialog design. For example, in extension of the example for guideline #6, it is a good practice to provide examples in the reprompt to an open-ended prompt, such as: **Open-ended prompt:** “Tell me, briefly, the reason for your call today”.

(Directed) Reprompt: “You can something like: ‘I need to make a payment’, or ‘my service is not working’”.

8. **Yes/no queries (generally) are very robust**

Well designed yes/no queries effectively stabilize dialogs after errors or ambiguous responses. Refer to [4] for more details on how to design robust dialogs with well designed yes/no queries.

**Problematic prompt:** Are you disconnecting all lines on your account, or just one?

**What’s the problem:** It is unclear how to interpret a “yes” or “no” response to this prompt; confusion about how to respond (“all lines” vs.“one”, or “yes” vss “no”)

**Improved Prompt:** Do you want to disconnect all lines on your account?

9. **Offer alternative input modalities**

Research has shown that offering alternative modalities for error correction dramatically increases correction success. Furthermore, alternative input modalities alleviate privacy issues if the user needs to provide sensitive information. The “improved prompt” below effectively leverages the keypad to avoid error spirals.

**Problematic prompt:** Please speak your account number again.

**What’s the problem:** No encouragement to switch modality.

**Improved Prompt:** Please re-enter your account number, and this time key it in if you spoke before.

10. **Choose persona judiciously**

While the value of personas for telephone dialog systems is debated, feedback from deployed applications and customer surveys indicate that callers prefer a professional persona for professional applications.

**Problematic design:**

I’m Sam your automated repair technician.

**Caller:** My phone’s not working.

Ooouu, sorry to hear that, what kind of problem are you having?

5. **Summary**

We presented a method for identifying design guidelines for broad classes of speech interfaces in two steps. First, the solution database of the framework of SUI design is queried to obtain lists of “solutions” to specific design problems of a broad class of SUIs. Second, expert knowledge is employed to reduce this list to a short list of best practices. We employed this method to generate a list of ten guidelines for telephone dialog design, and demonstrated their usefulness by critiquing various deployed telephone dialog systems. The contribution of this paper is not primarily in this specific set of guidelines, but in the theory-based approach that we have employed to generate them. Our framework can be applied in similar ways to generate guidelines for other classes of speech interfaces.

Good design requires creativity and an understanding of the intricate dependencies between conflicting design parameters, which no set of guidelines can replace completely. While the skillful application of guidelines still requires some experience, we believe that a better understanding of the limitations of speech, and the knowledge of best practices of SUI design will lead to better designs and more effective use of speech as an interface modality.

**Acknowledgements**

The author gratefully acknowledges the support of BBN Technologies in conducting this research.

6. **References**


