I Tell You Something

Dirk Schnelle-Walka
Telecooperation Group
Darmstadt University of Technology
Hochschulstraße 10
D-64283 Darmstadt, Germany
dirk@tk.informatik.tu-darmstadt.de
phone: +49 (6151) 16-64321

June 24, 2011

Abstract

The designers of voice based systems must carefully think about the way to deliver the information to the user to ensure a good user experience. However, the invisible and transient nature of audio makes it hard to accomplish this goal. In this paper, we introduce a pattern language to design the system output in spoken dialog systems.

1 Introduction

Fundamental parts of an interactive system are the human user on the one hand and the computer on the other hand. ”Traditionally, the purpose of in interactive system is to aid a user in accomplishing goals from some application domain” [8] For that purpose interfaces are needed. “An *interface* is a layer that stands between two other things. . . . an interface can be thought of as a double sided object or artifact - each side designed to *face* its mating medium in a way that allows the two to comfortably connect or join together" [3]. Hence an interface is a sort of agreement on how to collaborate. This is also true for user interfaces in the field of Human Computer Interaction (HCI). Hence, the human's capacity limitations to process information have to be considered [8] as well as the input and output capabilities of the computer. Developers must consider the interaction between the user and the computer for every use case, and define the interface to process the input or output. The interaction is much like a dialog between the user and the computer. Hence, user interface design, especially in the domain of voice based interaction is often referred to as *dialog design*.

Definition 1 *Voice User Interfaces (VUI)s are user interfaces using speech input through a speech recognizer and speech output through speech synthesis or prerecorded audio.*

The basic elements of VUI, also shown in figure 1, are

- ASR (**A**utomated **S**peech **R**ecognition) with its grammars that define the possible things users can say in response to each prompt and which are understood by the system.
- Dialog logic or call flow in the terminology of telephony based systems define the actions taken by the system.
- Prompts or system messages are the recordings or TTS (**T**ext **T**o **S**peech) played to the user during the dialog.

Here, a dialog means a spoken exchange of words and not a *dialog* in the sense it is used by some software designers to refer to a box containing written words. The dialog manager as

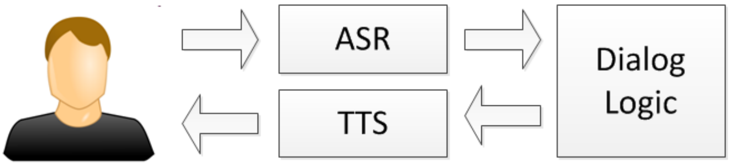

Figure 1: Information flow in voice user interfaces (based on [19])

the core component creates a link between the user input and the system output. Both, input and output must be carefully designed for a good user experience and are ". . . by far more than the sum of both"[9]. In [29] we already introduced a pattern language that was related to the input side. In this paper, we concentrate on the design of the system output, the prompts. In general output can be created using synthesized speech or by concatenating audio fragments. Here, we do not distinguish between these two but concentrate on general design aspects that can be applied in both approaches.

A key factor in the design of the system output is to personify the interface. The discussion about that has been started with the appearance of the AT&Ts famous *How may I help you?* [10]. This technique, which is also known as *anthropomorphism* or PERSONA [26] has been discussed very controversary in voice user interface design. On the one hand it seems to be strange talking to a machine. For instance Harris [15] comes up with the following statement of Woofit et al.:

> It is by no means unusual to find subjects saying "please" and "thank you" in their exchanges with what they thought was a machine [in a wizard study]. In one sense this is comparable to thanking a kettle for boiling. [34]

On the other hand, we do talk to computers. Harris concludes his thought about it with

> But it's not just the design of this machinery; in fact, it's not even just machinery. It's our design, and it's virtually everything. As far back as our species goes, we've been talking to things - sometimes worshipping, sometimes fearing, sometimes just shooting the breeze; it's the way we're wired. When kids are learning to talk, when elderly people reach the point where they forget there are other humans observing, or just don't care, when psychotics enter their own world, this penchant is more noticeable by the circumspect rest of us. But we all do it.

According to these thoughts the pattern language presented in this paper provides a language to discuss how *I [can] tell you something.*

2 The Difference with Audio

Development of audio based applications is different to development of graphical oriented applications. In [27] we grouped the challenges with audio into the two categories *technical challenges* and *audio inherent challenges*. It can be assumed that the technical problems, like speech recognition performance and speech synthesis quality, can be solved as technical progress is being made. The challenges inherent to audio will be impossible to solve completely, but it is important to know them and to find workarounds, probably with the help of our pattern language. The following sections are taken from [27].

2.1 Technical Challenges

Technical challenges include *speech-synthesis quality*, *speech recognition performance* and *flexibilty vs. accuracy*.

Speech synthesis quality The quality of modern text-to-speech engines is still low. In general, people prefer to listen to pre-recorded audio because it sounds more natural. However, for this to work, the data to be delivered has to be known in advance and recorded as audio, which consumes additional memory. For dynamic documents, where the content depends on the user's actions, text-to-speech may be the only feasible solution.

Speech recognition performance Speech is not recognized with an accuracy of 100%. Even humans are not able to do that. There will always be a doubt in the recognized input which has to be handled somehow.

Flexibility vs. Accuracy Speech can have many faces for the same issue and natural language user interfaces must serve many of them. This has a direct impact to recognition accuracy. To illustrate this trade off between flexibility of the interface and its accuracy, consider the following example for entering a date. A flexible interface would allow the user to speak the date in any format the user desires (e.g., "March 2nd", "yesterday", "2nd of March 2004", etc.). Another possibility would be to prompt the user individually for each of the components of the date (e.g., "Say the year", "Say the month", etc.). Obviously, the first method is much more flexible for the user, but requires much more work from the recognition software, and is far more error-prone than the second approach.

2.2 Audio Inherent Challenges

Audio inherent challenges are *one-dimensionality*, *transience*, *invisibility*, and *asymmetry*.

One-dimensionality The eye is active whereas the ear is passive, i.e. the ear cannot browse a set of recordings in the same way as the eye can scan a screen of text and figures. It has to wait until the information is available, and once received, it is not there anymore. This is meant by one-dimensionality.

Transience Listening is controlled by the short term memory. Listening to long utterances has the effect that users forget most of the information that was given at the beginning. This means that speech is not an ideal medium for delivering large amounts of data. Transience has also the effect that users of VUIs often have the problem to stay oriented. They describe a phenomenon, which is called *lost in space problem*, which is also known in web based application.

Invisibility It is difficult to indicate to the user what actions she may perform and what words and phrases she must say to perform these actions. In contrast to graphical environments, where the means to enable user interaction are directly related to capturing the user input, the presentation of a voice user interface is completely independent to the evaluation of the entered data. Moreover, invisibility may also leave the user with the impression that she does not control the system. Note that there is a difference between *feeling to be* in control and actually *being* in control.

Asymmetry Asymmetry means, that people can speak faster than they type, but can listen much more slowly than they can read [16]. This has a direct influence on the amount of audio data and the information being delivered. This property is extremely useful in the cases, where we have the possibility of using additional displays to supplement the basic audio interface. We can use the displays for delivering information, which is unsuitable for audio due to its length, and focus on using the audio device for interaction and delivering short pieces of information.

3 Prompt Categorization

In this paper we describe a set of patterns for spoken output that extend our pattern language that was described in [28, 27, 26, 29] with a focus on the system output. In voice-only interfaces spoken output can be categorized into

Opening Greetings Greet the caller and introduce the application

Prompt Indicates it is time for user input, and thus serves as a turn-taking cue

Feedback Presents the application state that results from user input, allowing the user to compare original intent with result

Application Data Information presented to the user as part of the task: e.g. weather, stock information, flight times

Instructions Provide information about operating the user interface or understanding the task

The opening greeting is the very first time the user gets into contact with the system. At this time the designer does not know anything about possible previous knowledge of the user about the system. However, it builds the basis for all coming interaction. To handle greetings we introduce the GREETING pattern.

Prompts are the turn taking queues of dialogs and serve mainly two purposes:

1. cause the user to speak
2. convey to the user what may be spoken (optionally)

Nicole Yankelovich discussed this in [35] and found that prompts fall along a continuum from implicit to explicit. This is shown in the following examples, taken from the speech interface design course in 2004 of Markku Turunen at the University of Tampere, where the same intention is prompted by three different computers.

Computer1: “Welcome to ABC Bank. What would you like to do?”
Computer2: “Welcome to ABC Bank. You can check an account balance, transfer funds, or pay a bill. What would you like to do?”
Computer3: “Welcome to ABC Bank. You can check an account balance, transfer funds, or pay a bill. Say one of the following choices: check balance, transfer funds, or pay bills.”

Obviously, *Computer1* is implicit and the following prompts become more and more explicit. In this paper we discuss the two extrema as the patterns IMPLICIT PROMPT and EXPLICIT PROMPT.

Language always occurs in a context. It’s structure is systematically context sensitive since it is essentially a communicative phenomenon. By means of cohesion devices [14] the whole is facilitated and comprehension is reinforced, but it is not responsible for creating meaning.

Definition 2 *Cohesion refers to explicit linguistic devices that help bind language into a coherent whole. [7]*

The effect of cohesion is illustrated by the following example:

User: “Dirk couldn’t wait to get to EuroPLoP. However, *he* didn’t attend the swimming BOF *there*”

The word *he* refers to *Dirk* and the word *there* refers to *EuroPLoP*. However, *there* is an adversarial or contrastive relation between the proposition to go to EuroPLoP.

Cohesion is relevant for VUI design, especially *i*) pronouns, *ii*) discourse markers and *iii*) special pointer words like *this* and *that*. These will be explored in the patterns PRONOUNS AND ADVERBS, DIALOG PROGRESS INDICATOR, END FOCUS PRINCIPLE, REGISTER and JARGON that deal with the aspects of discourse. Since the patterns that are introduced in this section are so general, they govern all other patterns of this language.

Feedback is a vital component of error prevention which has already been handled in [29]. Here, we described the patterns IMMEDIATE FEEDBACK, EXPLICIT CONFIRMATION and IMPLICIT CONFIRMATION.

Instructions are needed to inform the user how she can interact with the system. Since speech is invisible the user must be explicitly informed how to operate the application.

According to Cohen [7] possible implementations for instructions are *i*) reminder cards, *ii*) tutorials and *iii*) just-in-time Instructions.

The pattern EXPLICIT PROMPT is exactly what Cohen names *just-in-time instructions* [7]. The remaining patterns are described as the patterns REMINDER CARD and TUTORIAL.

4 Patterns

In this section the pattern language for system output in voice user interfaces is presented. An overview of the language with its relations is shown in figure 4. We stick to the format that we started in [28]. It is based on the GoF format [?] and also follows the suggested format of Tesanovic [32] which we find to be useful to talk about design issues.

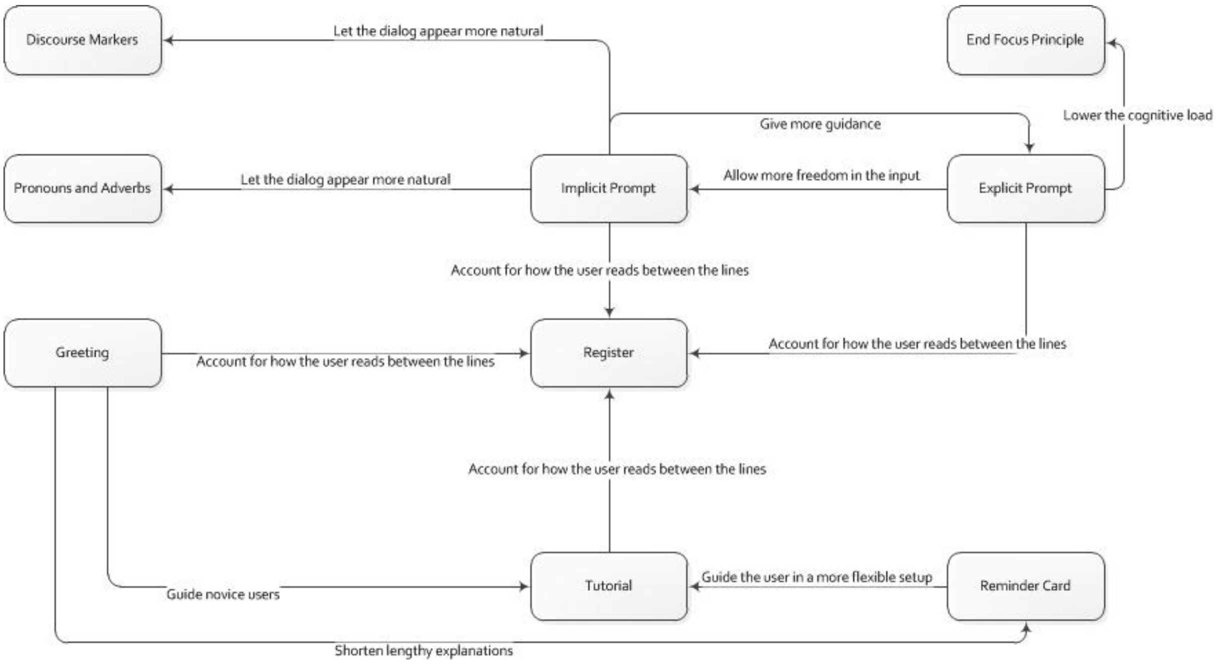
Discourse Markers
Let the dialog appear more natural
End Focus Principle
Give more guidance
Lower the cognitive load
Pronouns and Adverbs
Let the dialog appear more natural
Implicit Prompt
Allow more freedom in the input
Explicit Prompt
Account for how the user reads between the lines
Greeting
Account for how the user reads between the lines
Register
Account for how the user reads between the lines
Account for how the user reads between the lines
Guide novice users
Tutorial
Guide the user in a more flexible setup
Reminder Card
Shorten lengthy explanations

Figure 2: Overview of the pattern language

Greeting

Intent

Greet the user and inform her about the actions she may use within the application.

Context

The user started to work with the application. The user has not been identified if the caller's phone number is not available and has been registered by the application.

Problem

The first contact of the user with the system is governed by a lot of unknown aspects. The system does not know much about the users background at this time but has to inform her about the capabilities of the application. While novice users need a more detailed description how to interact with the system, more experienced users know what to say and just want to go ahead. How to inform the user about the actions she may take without knowing who the user's capability and knowledge?

Forces

- Opening greetings must be short.
- A lot of information has to be conveyed, also known as the *Inverted Pyramid Approach* [24].
- Users want to interact with patient and polite systems.
- Users prefer talking to a real human.
- Users know that they are talking to a machine and don't want to be fooled.
- Users bring in their knowledge about machines when they know that they are talking to a machine [4].
- Persons with a spaking disablity must be served.
- Speech is invisible: First time users do not know if the system can help them to achieve their goals.
- Capabilities and knowledge of users is unknown.
- "First-time or infrequent users are likely to require instructions and/or guidance through a system to allow them to build a cognitive model of how the system works and how to interact with it" [18]
- "Experienced users who interact frequently with the system want ways to bypass the instructions and move through the interaction more efficiently" [18]
- The given information is the basis for all further interaction [6].

Solution

The solution deals with a prioritized order of the most important information pieces that have to be delivered to the user. To implement this strategy consider the following:

1. Identify the system immediately so that users are sure they reached the right system, e.g
   Computer: "Welcome to ABC Bank".

2. Use an audio logo if this is known from other media (e.g. commercials) or introduce a new one.

3. Let the user know they are interacting with a computer, e.g.
   Computer: "I am Berti, the automated bank assistant".

4. Name the next actions the user may initiate, e.g.
   Computer: "You can check an account balance, transfer funds, or pay a bill.".

5. Name universals that are available throughout the application, to provide shortcuts for more experienced callers so that can reach their goal faster the next time they call.

6. Allow the user to barge-in using a universal or a valid command to trigger the next action.
7. Inform the user about help if available and how to get help.
8. Users with a speaking disability must also be handled, therefore name the possibility to be transferred to a human operator but do not advertise it.

Consequences

- ☺ Users know that they reached the right system.
- ☺ Users know that they are talking to a machine.
- ☺ Users are prepared to reuse their knowledge about talking to a machine through the reset of the application.
- ☺ Users may talk to a human upon request or in case of speaking disabilities.
- ☺ Users know what actions they may perform.
- ☺ Users learn shortcuts to reach their goals.
- ☺ Experienced users can use universals to reach their goal right away without being forced to the whole greeting.
- ☹ The solution tends to become lengthy since it does not solve the *Inverted Pyramid Approach* completely.
- ☹ Does not solve the force of being polite.

Related Patterns

May be combined with Tutorial to guide novice users.

Reminder Card can be used to shorten lengthy explanations.

Gives hints to a Universal as shortcuts.

The Register gives hints how to avoid a user reading between the lines and thus points into the direction of being polite.

Explicit Prompt

Intent

Provide instructions relevant to the task immediately at hand just before the user needs to perform the task

Context

The user has to provide some input or make a choice. The system expects a limited set of commands that the user may utter.

Problem

Since speech is invisible the user will not be able to provide the expected data without the prior knowledge what can be entered in the current field of a form or a choice in a menu. How to provide the required information at the right time?

Forces

- Speech is invisible: Users need guidance to know what to say.
- Audio is transient: Users forget the exact wording of the choices by the time.
- Novice users need more guidance to know what they might say.
- Experienced users know what to say and need a fast way to enter the data.
- Users want to enter the data as free as possible.
- Listing the possible commands appears to be rigid.
- Directive prompts can significantly increase user compliance to system restrictions [17].
- Recognizers perform better for a restricted vocabulary.
- Users must know the allowed vocabulary and grammar of the system.
- In order to minimize cognitive load the number of words and choices must be limited to a small set [7].
- Voice interfaces must be designed for efficiency so that the number of steps a user must take should be minimized [7].
- An increasing verbosity of the prompt leads to shorter user responses [2].

Solution

In the solution, the information that the user needs to enter the data or make a choice is given right before a command is expected from the user. This enables the user to know what she is expected to say in time so that the grammars can remain very restricted. To implement this strategy consider the following:

1. Restrict the vocabulary to some basic commands.

2. Tell the users the exact words they should say.

3. Enable barge-in to enable the user to interrupt the playing of the prompt.

Consequences

☺ Users are enabled to operate the system and can say the option as soon as they hear it. This is especially helpful for novice users.

☺ Offers instructions in small pieces at that time when the information is relevant.

☺ Restricts the user to short and expected utterances, so grammars can be very small to allow for a better recognition accuracy.

☹ Users may not hear all possible choices.

☹ Does not effect the number of steps a user must take but is not perceived as very efficient due to a verbose output.

- ☹ Users are restricted in entering their data so the overall system appears to be rigid which is especially true for experienced users.
- ☹ Cannot be applied for a large set of options.

Related Patterns

Usually, prompts fall along a continuum from explicit to implicit. Consider using Implicit Prompt if you want to allow more freedom in the input. Follow the End Focus Principle when designing the prompts to lower the cognitive load to remember the possible inputs. Cooperative Principle accounts for how and why a user reads between the lines. Can be applied for Form Filling or Menu Hierarchy (cf. [27]) to provide the user with instructions what she may say.

IMPLICIT PROMPT

Intent

Encourage the user to speak sentences to a constraint grammar.

Context

The user has to utter a command to make a choice in a menu or enter some data in a field of a form.

Problem

Speech is often regarded as a natural way of communication. Therefore, users want to mimic the conversation as they know it from their communication with other humans. However, an automated system does not have the conversational capabilities of a user and can accept only a limited set of commands. How to shape user input in conversational systems imitating natural ways of communication?

Forces

- Speech is invisible: Users need guidance to know what to say.
- Audio is transient: Users forget the exact wording of the choices by the time.
- Novice users need more guidance to know what they might say.
- Experienced users know what to say and need a fast way to enter the data.
- Users want to enter the data as free as possible.
- Flexible grammars tend to increase error rates.
- Directive prompts can significantly increase user compliance to system restrictions [17].
- People adapt to the way that the computer speaks and use both the same style and words which occur in the computer's turns [33].
- Users must know the allowed vocabulary and grammar of the system.

Solution

The solution deals with giving subtle hints in the system output to direct the input from the user. To implement this strategy consider the following

1. Use a limited vocabulary for each input
2. Design the sentences that lets the user speak in a constraint manner. E.g.,
   Computer: "There are 100 convertibles. Can you specify a make or make and model?"
3. If the user does not know what to say suggest a query that the system is likely to understand
4. Use subtle discourse cues in the prompt to direct the user's input. E.g.
   User: "What's on Bob's calendar tomorrow?"

   Computer: "Did you mean Bob Kuhn or Bob Sproull?"

   User: "I meant Bob Sproull"

Consequences

- ☺ Users have the impression to speak freely. Especially experienced users will be able to provide their data fast.
- ☺ Guides the user to operate the application at that time when the information is relevant.
- ☺ Users will mostly use a restricted vocabulary that can be handled by a small and less flexible grammar. Hence recognition rates will be still high.
- ☺ Does not rely on the users ability to memorize the valid commands.

- ☹ Does not hinder the user to speak sentences outside the vocabulary which is pushed by the impression to speak speak freely.

Related Patterns

Usually, prompts falls along a continuum from explicit to implicit. Less restrictive than Explicit Prompt which gives the user more guidance. Dialog Progress Indicator and Pronouns and Adverbs help to let the dialog appear more natural. Register accounts for how and why a user reads between the lines. Can be applied for Form Filling or Menu Hierarchy (cf. [27]) to give lightweight guidance to the user what she might say.

Pronouns and Adverbs

Intent

Present information in one unit of talk often presupposes information presented in a previous one to let the dialog appear more natural.

Context

A piece of information can be referred to by a name and occurs frequently in the system output.

Problem

In our daily communication it occurs frequently that the same piece of information has to be referred to in subsequent utterances. Repeating this information let the dialog appear rigid and contradicts the aim for a natural interaction. How to present this piece of information without repeating it while aiming for a more natural interaction?

Forces

- The same information must be referred to frequently within a the system output.
- Repeating the name makes the user interface rigid.
- Users prefer a more natural way of interaction.
- Cohesive text contains related words in close proximity [14].
- Linguistic analysis showed that a conversation's "deep structure" is expressed in terms of structural relations between discourse elements [12, 20].

Solution

The solution deals with imitating the way humans structure information in our daily communications. Therefore consider the following:

- Use pronouns and adverbs to present information in one unit of talk often. that presupposes information presented in a previous one.

Pronouns substitute for a noun (or a noun phrase) and can be used e.g. as follows

User: "A large pizza please."
Computer: "OK. A large pizza. Do you want to have *it* with ham, salami or mushrooms?"

Here, the second occurrence of *pizza* is replaced by the pronoun *it*.

Adverbs can modify a verb, an adjective, another phrase or clause to indicate e.g. manner, time and place. In the following example, the adverb *unfortunately* modifies the entire clause of the first sentence:

Computer: "Unfortunately, all agents are busy, You will be transferred as soon as the next agent is available."

Consequences

☺ The system output is more natural and is preferred by most users.

☺ The same information is referred to by linguistic concepts that we know from the conversation with other humans.

☺ The system output does not appear to be rigid.

☹ Users are forced to resolve the pronouns and adverbs even if they are not in very close proximity.

☹ Puts a lot of work to the VUI designers to take care about the deep structure of the conversation and to have the cohesive text in close proximity.

☹ Hard to apply if the output is generated.

Related Patterns

This pattern is very basic and hence effects all patterns in the language.

Example

The use of pronouns can be used to make the dialog appear more natural. The following examples for the use of pronouns are taken from [7] and is a good example to show the same dialog with and without the use of pronouns. Without the use of pronouns, a voice based bookmarking management system may offer the following dialog:

System: "You have five *bookmarks*. Here's the first *bookmark*... Next *bookmark*... That was the last *bookmark*."
User: "Delete a *bookmark*."
System: "Which *bookmark* would you like to delete?"
<...>
System: "So you want to delete another *bookmark*?"

The same system that make use of pronouns could be realized as

System: "You have five *bookmarks*. Here's the first *one*... Next *one*... That was the last *one*."
User: "Delete a *bookmark*."
System: "Which *one* would you like to delete?"
<...>
System: "So you want to delete another *one*?"

Although the user input is the same. the second dialog is preferred by most users.

Dialog Progress Indicator

Intent

Bracket sequentially dependent units of talk to let the dialog appear more natural and inform the user on how the dialog is progressing.

Context

The user has to enter a sequence of information items to complete a request.

Problem

A dialog step may require that the the user has to provide a sequence of information items. Since speech is invisible she does not have a clue how the dialog will be progressing and will be less willing to provide the information as the system keeps on asking for more information. How to inform the user on how the dialog is progressing.

Forces

- Sequentially dependent elements must be tied together although they appear in different prompts.
- Users want to be aware about how the dialog is progressing.
- Speech is invisible: The user does not know how many questions will be asked.
- The conversational process within a voice user interface can be viewed as a series of moves carrying the user to another stage of discourse [11]
- Asking for one information item after the other may be perceived as rigid.

Solution

The solution deals with imitating how humans inform the other party in a human-to-human conversation about the status of a questionnaire. To implement this strategy make use of discourse markers to bracket sequentially dependent units of talk. They may be from one of the following categories as listed in [7]. Here, we present only a snippet from this list to give an impression about the nature of discourse markers.

Enumerative first, second, third, for one thing, and for another thing, to begin with, for starters, in the first place, in the second place, one, two, three, next, then, finally, last, lastly, to conclude

Reinforcing also, furthermore, moreover, then, in addition, above all, what's more

Equative equally, likewise, similarly, in the same way

Transitional by the way, incidentally, now

Summative them (all) in all, in conclusion, in sum, to sum up

Apposition namely, in other words, for example, for instance, that is, that is to say

Result consequently, hence, so, therefore, thus, as a result, somehow, for some reason or other

Inferential else, otherwise, then, in other words, in that case

Reformulatory better, rather, in other words

Replacive alternatively, rather, on the other hand

Antithetic ...

They can be used as follows to enter the dates when asking for a train connection:

Computer: "*First*, tell me from where you want to leave"
User: "I want to leave from Darmstadt."
Computer: "*Next*, I'll need the destination."
User: "Frankfurt"
Computer: "*Finally*, I need to know the departure date and time"
. **User:** "Tomoorow morning at 10:00."

Consequences

- ☺ The redundant nature of discourse markers reinforces the functional relationship between two units of discourse.
- ☺ The user becomes aware about the conversational process and on how the dialog evolves
- ☺ Is not perceived to be that rigid although a series of questions is been asked.
- ☺ Discourse markers suggest a human-like awareness of how the dialog is progressing.
- ☹ Customers may resist the use of discourse markers on the grounds that they are perceived as informal or slang.

Related Patterns

Can be used in FORM FILLING [27] to let the dialog appear more natural.

End Focus Principle

Intent

Structure the given information in a way that it is easier for the user to remember it.

Context

Multiple pieces of information are given to the user. This may either be the case if the user is requested to utter a command to take a specific action or if some follow-up application data are to be presented to the user.

Problem

Audio is no suitable means to deliver multiple pieces of information. The transient and one-dimensional nature poses high cognitive load onto the user. How to present information so that it is easy for the user to remember it?

Forces

- Audio is one-dimensional: The user has to wait for the relevant information to become available.
- Users want to have the relevant information at once.
- Audio is transient: The given information is gone once it is presented to the user.
- Old information is erased by new unless it has been transferred to the long-term store [5].

Solution

The solution exploits the recency effect [1]. It is easier for users to remember the last things that they hear. To implement this strategy consider the following:

1. Consider the placement of contextually determined old versus new information.
2. Follow the end-focus principle: End focus is given to the last open-class item or proper noun in a clause.
   - Old information at the start.
   - New information in the end.
3. If users are requested to use specific words, make that the last thing that they hear.
4. Order: **Function** and then **action**. For instance, the following prompt can be used to ask for the topping within a pizza ordering process:
   Computer: “To select a topping say Ham, Salami or Mushrooms.”

Consequences

☺ Exploits measured psychological effects to design for an order of information that is easy to remember.

☺ The user is put into context before so that it is easier to process the newly delivered information.

☹ The selected word order may not sound natural in a given context.

☹ The user has to hear to all the information before getting to the relevant information and does not have the relevant information at once.

Register

Intent

Use the right level of formality and account for how the user reads between the lines.

Context

The application has to deliver some information or indicate the user to enter some data. The target group is closed and known in advance.

Problem

Language is governed by meta information that influence what can be expressed within a particular context [13]. This is also true for the computer as the counterpart. Hence, prompt designers have to care about these social relationships.

Forces

- Contextual knowledge influences the way how a user interprets the delivered information.
- The contextual background of the user is not always obvious to the system designer.
- Too casual or to distant prompts hinder the acceptance of a user.
- The level of formality can only be guessed within the target group.
- The level of formality must match the target group.
- The prompt designer must be familiar with the interpretation of wordings within the target group.
- The way users within the target group communicate may differ from their expectations while talking to a computer.
- Speakers change their registers all the time and completely effortlessly, sometimes even in mid-sentence [25].
- The register of an application will be fixed.
- A conversational partner is expected to make her contribution to a conversation clear [11].

Solution

The solution deals with a detailed understanding of the target group. Therefore, try to understand how the people within the target group talk to each other and their expectations while talking to a computer. This speaking style is also reffered to as *register* [7]. To implement this strategy consider the following:

- Understand the registers of your target group regarding
  - sharing assumptions and expectations regarding the topic
  - expectations, how the conversation should develop
  - Quality and quantity of the contributions of the participants
  - Politeness
  - Consistency
  - ...
- In case you are in doubt, talk to a representative of that group to get a better understanding.
- Design the prompts with the following dimensions of register in mind [13]:

  Mode is the channel of communication like writing or speaking.

  Field focuses on the content of discourse and the social setting in which the language is being used.

Tenor focuses on the roles and relationship of the participants such as information seekers and information providers.

The mode takes respect to the differences of written and spoken language and if the communication happens in person or remote. This is invariable in voice-only interfaces on the input side. In telephony environments, mode can also fall back to DTMF input or in multimodal interfaces to the use of other modalities. On the output side mode can also make use non-speech sounds such as auditory icons, earcons or music.

- Design your prompts to implicate the maxims of quantity, quality and relevance [15] to account for the manner of speaking [11] by
  - Avoid obscurity of expression.
  - Avoid ambiguity.
  - Be brief (avoid unnecessary prolixity).
  - Be orderly.

Consequences

☺ The right level of familiarity is used.

☺ The system prompts are designed in a way that the level of meta information that the user receives is close to the designer's intent.

☺ The system makes clear contributions to the conversation.

☺ Has a positive effect on the user experience if the level of formality matches the target group.

☹ Is hard to implement if the target group varies widely

☹ Does not follow the speaker's change of register.

☹ It is nearly impossible to hinder the user from interpreting unintended information from the prompt using her still unknown contextual background, although it can be reduced significantly.

Example

"Voice interface design is impossible without a thorough understanding of the relevant registers" [15]. The following example. taken from [15], shows show the effect of register in VUI design. In case Dred and Barny arrive at the same location in close temporal proximity. While Fred arrives at 7:00, Barny arrives at 7:01, the following statement can be interpreted differently dependent on register.

Computer: "Fred and Barney arrived at the same time"

If the location is a party this might be an appropriate statement. In a sports register where the system reports about the time both crossed the finish line, the statement will not hold any more.

Another example, taken from [7] shows the effect of register in a GREETING:

Computer: "Hello and welcome to the Frequent Buyer Reward. You can now redeem points online at www dot frequentbuyer dot com"

The original intention of the authors was to inform the callers about the new service. But they failed with their message because the user gets the impression that she better uses the web solution rather than continuing with the automated telephone system.

Related Patterns

REGISTER may be confused with JARGON. REGISTER deals with what can be read between the lines while JARGON makes use of a specialized language.

AUDITORY ICON, EARCON and MUSIC can be used apart from speech to implement mode.

Jargon

Intent

Use the specialized language of the target group

Context

Industries in the vertical market, e.g. in healthcare, often have their own vocabulary that is usually very specific and known to the users of this group. Due to the high specialization the words are well suited to come to the point.

Problem

Special target groups use their own language for communication that is highly adapted to differentiate between relevant expressions. On the other hand. users who are not familiar with this special language and are used of the *default* language of the country will not be able to understand what is expected from them and how to judge the information that was given.

Forces

- Users are more willing to accept the application if it is *speaking* their language.
- Specialized vocabularies make it easier to come to the core of the information.
- Audio is one-dimensional: The user has to wait until the information of interest is presented.
- The specific language must be well understood.
- Excessive use can decrease the acceptance.
- The way how the target group communicates must be well understood.
- Users of an application should not be forced to learn an industry jargon [22].
- Communication between groups can be fraught with difficulties as they come from different backgrounds and have different jargon [23].

Solution

The solution adapts the behavior of the Coquillards. A gang of criminals in northern France around 1450. They were using a special language that they called *Jargon* [36]. Therefore, consider the following

- Try to understand how the users within your target group are communicating.
- Collect the information that you want to deliver in your prompts and make use of the specialized language.
- Consider an expert from the target group if you are in doubt.

For instance a dialog system with a more technical background can come up with the follwoing message after an internal error has occurred within the application:
Computer: “An error has occurred. Return to main”

The same message could be synthesized as follows for a broader target group:
Computer: “Sorry, there was a technical problem, so we’ll have to go back to the main menu.”

Consequences

☺ Has a positive effect on the user experience if the level of formality matches the target group.

☺ Information can be delivered in a way that it is well understood by the target group.

☺ The delivered information is precise and short.

☺ Can also be applied to groups who are not part of the vertical market like people involved in sports or casual groups.

- ☹ Can only be applied if a closed target group exists and only for that special target group.
- ☹ May result in users rejecting the application if the language is not correctly used.
- ☹ Excessive use may result in Bullshit Bingo.

Related Patterns

Register may be confused with Jargon. Register deals with what can be read between the lines while Jargon makes use of a specialized language.

Reminder Card

Intent

Provide the user with a list of available commands

Context

It is known that a person is going to use a certain application that she did not use before.

Problem

The user does not know how the designer structured the application. However, a basic understanding of the underlying structure helps the user to operate the application more efficiently. How to inform users about the available commands in advance?

Forces

- Too complex structure hinders users from using the system at all.
- Not all users are known in advance.
- Users want to use the service right away.
- The behavior of a system may be a subject of change in a maintenance release after the first start.
- Speech is invisible: Users do not know how the developers structured the system.
- First-time or infrequent users are likely to require instructions and/or guidance through a system [17].
- Experienced users who interact frequently with the system want ways to bypass the instructions and move through. the interaction more efficiently [17].
- Humans are visual oriented [30].

Solution

The solution leaves the field of voice-only solutions as presented in this pattern language and uses a hardcopy to visualize the structure of the system and how the user may navigate the application. An example is shown in figure 4 This hardcopy is shipped to the users as an additional information that they can have at hand while using the system. To implement this strategy consider the following:

1. Prepare a sheet or a flyer with instructions on how to use the application.
2. Ensure that the call flow can be easily grabbed e.g. by using state. diagrams
    - Use descriptive names for the states.
    - Mark the transitions with the commands initiating the transitions.
3. Send the instruction list to the user through mail or e-mail.

Consequences

☺ Solves the *What can I say?* problem and enables users to use the system right-away from the start without the need to give lengthy explanations even for complex structures.

☺ Can save time for power users since lengthy explanations are not needed.

☺ The solution visualizes the structure of the system so that user are able to see how it is structured

☹ Not very effective since users tend to start using the application before reading the instructions.

☹ User must be known or registered.

☹ Changes are not easily doable.

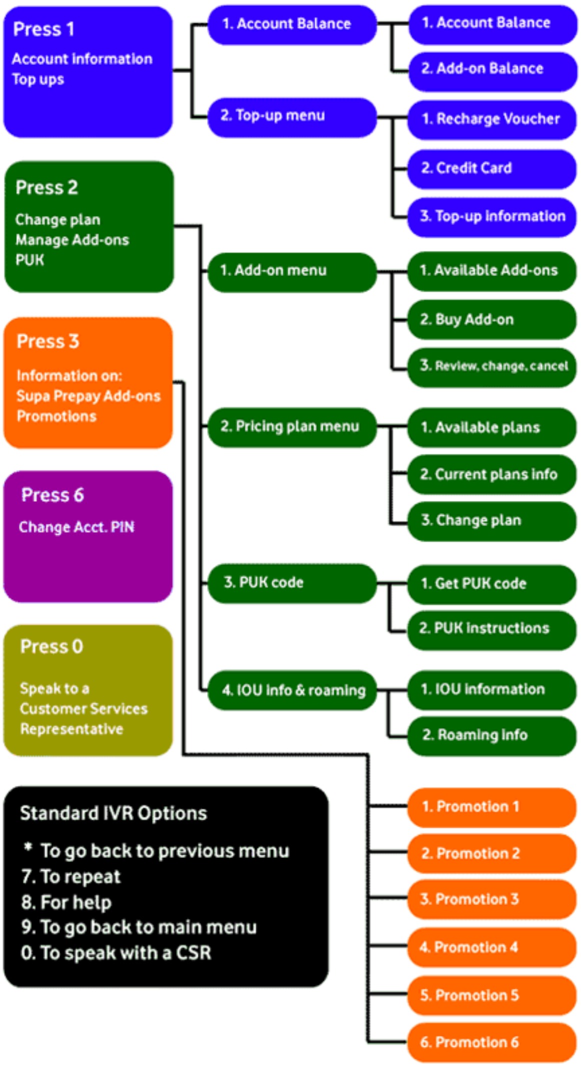

Figure 3: Example of a reminder card. Source http://www.vodafone.co.nz

Related Patterns

TUTORIAL offers a way to guide the user without the need of additional items or in a more flexible setup.

GREETING may be shorter since some of the information is already there.

Tutorial

Intent

Provide users with a step-by-step instruction for using certain features of the application to new or infrequent users.

Context

Although Greeting gives a broad overview about the goals of the system and its services, novice and infrequent users may still need additional information to operate the system. This can be either in the beginning or while they are actually using a special service.

Problem

Novice users need more detailed instruction to learn how to operate the application. The instructions tend to become lengthy for more complex applications. The transient nature of audio lets the user forget most of what she heard when it is about to apply the knowledge [21]. How to instruct users on their first time use of a service or after they forgot the usage instructions how to accomplish their goal?

Forces

- Users have difficulties in following a lengthy description of diverse functionality.
- Novice users need more guidance to know what they might say.
- Experienced users know what to say and need a fast way to enter the data.
- If a feature is not exercised immediately, it is likely to be forgotten.
- “Research shows that callers lose attention after 10-15 seconds of listening to a prompt” [31], especially if it is not helping meet their immediate needs.
- Not all information can be given in the Greeting.

Solution

The solution tries to identify the user and uses this information as a clue if detailed information should be presented. In case of such a detailed guidance, the user will be taught a small number of concepts to interact with the system and also let him know that she can ask for more help if needed. To implement this strategy consider the following:

1. If necessary, identify the user using PIN, Phone as a Token or Speaker Identification or similar

2. If the user already accessed the service continue with the service

3. If the user accesses this service for the first time provide her with step-by-step instructions on how to accomplish their goal

4. Let the user know that help is available and how to get help. This is usually done by expecting that the user may say *help* at any time.

5. Teach only a very small number of concepts

6. Use examples for help

7. Make it interactive. Have the user actually perform the action

8. The instructions can also be given by means of a demo. The demo is a recorded interaction between a simulated caller and the system. The system voice speaks the prompts that are played during real system use.

Also consider the following when writing prompts for the instructions: Use declarative sentences for instructions.

Avoid:

Computer: “Read one of the fund names from your reference card.”
Computer: “Choose from the following list.”
Computer: “Check the form: how many digits are in the PIN? The answer is seven.”

Use:

Computer: “The fund names are printed on your reference card.”
Computer: “You may choose one of these.”
Computer: “A PIN must be at least seven digits long.”

Make declarative sentences user centered.

Avoid:

Computer: “The menu consists of five items.”

Use:

Computer: “You may choose from one of the following.”

Consider replacing ambiguous instructions with questions.

Avoid:

Computer: “Please have your credit card ready.”

Use:

Computer: “Do you have your credit card ready?”

Consequences

☺ Provides an effective instructional tool for first time or infrequent users.
☺ Everybody can get a feeling for the application.
☺ Users are taught by the application itself.
☺ No previous knowledge is needed.
☺ Experienced users can skip the tutorial and can use the system tight-away.
☺ The help is given in time and can thus be short.
☺ Reduces the amount of information that has to be given in Greeting
☹ Not suitable for a large amount of information.
☺ Users can exercise the lessons learned and can remember the information more easily.
☹ Can be boring if the user already knows comparable systems.
☹ Users may loose attention if the examples chosen are too long.

Sample Code

Here is an example of tutorial using different voices:

<interrupting help chime>
Help Coach: “State the date like this…”
Prompt Voice: “Date of birth?”
Male Voice: “Six-eighteen-forty-nine”
Help Coach: “Now you try it.”
<closing help chime>
Prompt Voice: “Date of birth?”

Related Patterns

The availability of a *help* command is usually implemented as a UNIVERSAL (ref. to [29]). REGISTER accounts for how and why a user reads between the lines.

Takes some information to convey from GREETING.

5 Conclusion

The patterns that we presented here integrates into the pattern language for voice user interface design that was presented started in [28] and continued in [27, 26, 29]. This paper introduced a more detailed concept of system output patterns.

In this paper we introduced first steps towards a pattern language to help designers to cope with the invisible and transient nature to achieve a good user experience with a focus on system output in spoken dialog systems. An overview of the language with its relations is shown in figure 4.

In this language we differentiated between different types of system outputs.

For greetings we introduced the pattern GREETING.

Prompts that are used to indicate turn-taking are handled by the patterns EXPLICIT PROMPT and IMPLICIT PROMPT

Feedback and application data can be handled by PRONOUNS AND ADVERBS, DIALOG PROGRESS INDICATOR, END FOCUS PRINCIPLE, REGISTER and JARGON.

Instructions can be implemented as IMPLICIT PROMPTS, TUTORIAL or REMINDER CARD.

In the future we will extend this language, e.g. to also cover aspects of synthesized speech and prerecorded speech output.

6 Acknowledgments

Thanks a lot to Bettina Biel who who was the EuroPLoP shepherd for this paper and gave some excellent and constructive advice on how it might be improved.

References

[1] R. C. Atkinson and R. M. Shiffrin. Human memory: A proposed system and its control processes. In K.W. Spence & J.T. Spence, editor, *The Psychology of Learning and Motivation.* New York: Acad. Press, 1968.

[2] C. Baber. *Interactive speech technology: Human factors issues in the application of speech input/output to computers*, chapter Developing interactive speech technology, pages 1–18. Taylor & Francis, 1993.

[3] Bruce Ballentine. *It's better to be a good machine than a bad person.* ICMI Press, 2007.

[4] S. J. Boyce. *Human factors and voice interactive systems*, chapter Spoken natural dialog systems: User interface issues for the future, pages 37–61. Kluwer Academic Publishers, 1999.

[5] J. Brown. Short-term memory. *British Medical Bulletin*, 20(1):8, 1964.

[6] H.H. Clark and S.E. Brennan. Grounding in communication. *Perspectives on socially shared cognition*, 13(1991):127–149, 1991.

[7] Michael H. Cohen, James P. Giangola, and Jennifer Balogh. *Voice User Interface Design.* Addison-Wesley, Boston, January 2004.

[8] A. Dix, J. Finlay, and G.D. Abowd. *Human-computer interaction.* Prentice hall, 2004.

[9] Klaus-Rüdiger Fellbaum. Speech input and output technology - state of the art and selected applications. In Antje Düsterhöft and Bernhard Thalheim, editors, *8th International Conference on Applications of Natural Language to Information Systems*, pages 7–13, Burg (Spreewald), June 2003. GI, Gesellschaft für Informatik.

[10] A. L. Gorin, G. Riccardi, and J.H. Wright. How may i help you? *Speech Communication*, 23:113–127, 1997.

[11] H.P. Grice. Logic and conversation. *New York*, pages 41–58, 1975.

[12] Barbara Jean Grosz. *The representation and use of focus in dialogue understanding.* PhD thesis, 1977. AAI7731381.

[13] M.A.K. Halliday. *Language As Social Semiotic: Social Interpretation of Language and Meaning.* Hodder Arnold, 1978.

[14] M.A.K. Halliday and Rugaiya Hasan. *Cohesion in English.* Longman Pub Group, July 1976.

[15] Randy Allen Harris. *Voice Interaction Design: Crafting the New Conversational Speech Systems.* Morgan Kaufmann, 2004.

[16] J.A. Jacko and A. Sears. *The human-computer interaction handbook: fundamentals, evolving technologies, and emerging applications.* CRC Press, 2003.

[17] C. Kamm. User interfaces for voice applications. *Proceedings of the National Academy of Sciences of the United States of America*, 92(22):10031, 1995.

[18] Candace Kamm. User interfaces for voice applications. In Lawrence R. Rabiner, editor, *Proceedings of the National Academic Science, Human-Machine Communication by Voice*, volume 92, pages 10031–10037, October 1995.

[19] Siegfried Kunzmann. Applied speech processing technologies - our journey. *ELRA Newsletter*, January-March:6–8, 2000.

[20] C. Linde. Information structures in discourse. *Studies in language variation: semantics, syntax, phonology, pragmatics, social situations, ethnographic approaches*, page 226, 1977.

[21] George A. Miller. The magical number seven, plus or minus two: Some limtis on our capacity for processing information. *Psycological Review*, 63:81–97, 1956.

[22] Mike Moore and Jim Milroy. *Speech in the User Interface: Lessons from Experience*, chapter Applying best practices to VUI design using a vertical market approach. Trafford Publishing, February 2010.

[23] A.F. Newell and P. Gregor. User sensitive inclusive design-in search of a new paradigm. In *Proceedings on the 2000 conference on Universal Usability*, pages 39–44. ACM, 2000.

[24] Jakob Nielsen. Inverted pyramids in cyberspace. `http://www.useit.com/alertbox/9606.html`, June 1996. accessed on 04/04/2011.

[25] Florian Schiel, Christoph Draxler, and Marion Libossek. Lingua machinae - an unorthodox proposal. In *INTERSPEECH*, 2006.

[26] Dirk Schnelle. *Context Aware Voice User Interfaces for Workflow Support.* PhD thesis, Technische Universität Darmstadt, 2008.

[27] Dirk Schnelle and Fernando Lyardet. Voice User Interface Designt Patterns. In *EuroPLoP 2006 Conference Proceedings*, 2006.

[28] Dirk Schnelle, Fernando Lyardet, and Tao Wei. Audio navigation patterns. In Uwe Zdun Andy Longshaw, editor, *Proceedings of 10th European Conference on Pattern Languages of Programs (EuroPlop 2005)*, pages 237–260. UVK Universitätsverlag Konstanz, 2005.

[29] Dirk Schnelle-Walka. A Pattern Language for Error Management in Voice User Interfaces. In *EuroPLoP 2010 Conference Proceedings*, 2010.

[30] Josef W. Seifert. *Visualisieren. Präsentieren. Moderieren.* Gabal Verlag GmbH, 2009.

[31] Bernhard Suhm. *Human factors and voice interactive systems*, chapter IVR Usability engineering using guidelines and analysis of end-to-end calls, pages 1–43. Springer, 2007.

[32] Alksandra Tešanović. What is a pattern. In *Dr.ing. course DT8100 (prev. 78901 / 45942 / DIF8901) Object-oriented Systems.* IDA Department of Computer and Information Science, Linköping, Sweden, 2005.

[33] M. Turunen et al. Speech Application Design and Development. 2004.

[34] R. Woofit, N.M. Fraser, N. Gilbert, and S. McGlashan. *Humans, Computers and Wizards: Conversation Analysis and Human Computer Interaction.* 1997.

[35] Nicole Yankelovich. How do users know what to say? *interactions*, 3(6):32–43, 1996.

[36] ZDF. Metropolis: Mord in Paris, 2007.