Usability Methodologies for real-life Voice User Interfaces

Georgios Kouroupetroglou and Dimitris Spiliotopoulos
University of Athens, Greece

ABSTRACT

This paper studies the usability methodologies for spoken dialogue web interfaces along with the appropriate designer-needs analysis. The work unfolds a theoretical perspective to the methods that are extensively used and provides a framework description for creating and testing usable content and applications for conversational interfaces. The main concerns include the design issues for usability testing and evaluation during the development lifecycle, the basic customer experience metrics and the problems that arise after the deployment of real-life systems. Through the discussion of the evaluation and testing methods, this paper argues on the importance and the potential of wizard-based functional assessment and usability testing for deployed systems, presenting an appropriate environment as part of an integrated development framework.

KEYWORDS

Spoken Dialogue Technology, Voice Response Systems, Voice User Interface, Speech User Interface, Interactive Speech Systems, Auditory User Interface, VoiceWeb, Spoken Dialogue Web Interface, Spoken Dialogue Usability, Voice Interface Evaluation, Web Accessibility, Design-for-All

INTRODUCTION

Web technology is rapidly reaching maturity making its use practically possible for most applications by the majority of potential users in the recent years. With high speed internet availability providing access to demanding multimodal services to all homes, most people can reap the benefits of real-time services ranging from voice banking to online socialising and beyond. Most high-level services are provided solely through web pages in the traditional point-and-click manner. In an effort to boost *customer experience* most providers deploy spoken dialogue interfaces as a means to increased naturalness of information access.

Due to the complexity of natural language interaction, it is becoming very important to build spoken language interfaces as easily as possible using the enabling technologies. However, not all technologies involved in the process are of the same maturity, let alone standardisation. Furthermore, there are only a handful of platforms available for building such systems. Given the range, variability and complexity of the actual business cases it is obvious that the enabling technologies may produce working systems of variable usefulness due to design and/or implementation limitations.

As with all human-computer interfaces, speech-based interfaces are built with the target user in mind, based on the requirements analysis. However, they differ from the traditional graphical user interfaces and web interfaces. The use of speech as the main input and output mode necessitates the use of *dialogue* for the human-machine communication and information flow. Information is received by the speech interface and presented to the user in chunks, much alike a dialogue between two humans. The input is recognised, interpreted, managed, and the response is constructed and uttered using speech. The naturalness is indeed far more enhanced than using forms and buttons on a traditional web interface. But, is the user satisfaction similarly improved? Does the performance of the resulting application meet the user requirements? How is usability ensured by design and verified by evaluation in a spoken dialogue web interface?

This work discusses the background of speech-based human-computer interaction and elaborates on the spoken dialogue interfaces. It explores what usability is and how it is ensured for natural language interaction interface design and implementation, both from the designer and the application deployment (business use) points of view. Finally, it presents methodologies for usability testing of spoken dialogue web interfaces, especially focusing on the need for an integrated design and implementation approach that includes already deployed interfaces.

BACKGROUND

People use the web and engage in several different activities, information retrieval, problem solving, entertainment, social interaction, personal, work, etc. Human-computer interaction is the study of interactive communication between humans and computers. People acquire communicative skills over time through the experience of using and operating the user interfaces. As the level of user adeptness rises, the speed and accuracy of the operation increases. The user adapts to the system and interacts more efficiently. The level of absolute efficiency corresponds to the actual system design, and can be assessed either as a full system or as a breakdown of its fundamental design modules or processes. In order to evaluate usability of such interfaces it is important to understand their design requirements and their architecture. The architecture of most applications falls into specific interaction frameworks, described below.

Multimodal Interaction

A general framework (Larson et al., 2003) for the description and discussion of multimodal interaction on the web is developed by the World Wide Web Consortium (W3C). It describes the input and output modes that can be used in a relational abstractive architecture that includes all component types required for the interaction.

In such framework, an application may handle several requests through one or more input modes and respond accordingly. The user may use their input options to make a request for an archive retrieval, the system may respond by either requesting an explicit verification or present all options from the retrieval function, the user may specify or select their preference, allowing the application to present the information. Consider the following examples:

Example 1:

User: “I would like to see highlights from the 2008 Olympic Games, please.” [spoken input]

System: “Please specify the sport category.” [spoken output]

User: “Tennis.” [spoken input]
System: (*starts showing highlights*) [screen output]

Example 2:
User: (*selects event: olympic_games, sport: tennis, action: highlights*) [keyboard input]
System: (*shows available video thumbnails for selection sorted by date*) [screen output]
User: (*clicks a thumbnail*) [pointing device input]
System: (*starts showing highlights*) [screen output]

The above short examples illustrate how a web interface handles an interaction. In the former example the interaction is achieved through speech, while in the latter using other input methods. Both examples can be serviced by an application within the multimodal interaction framework. Figure 1 illustrates the basic components:

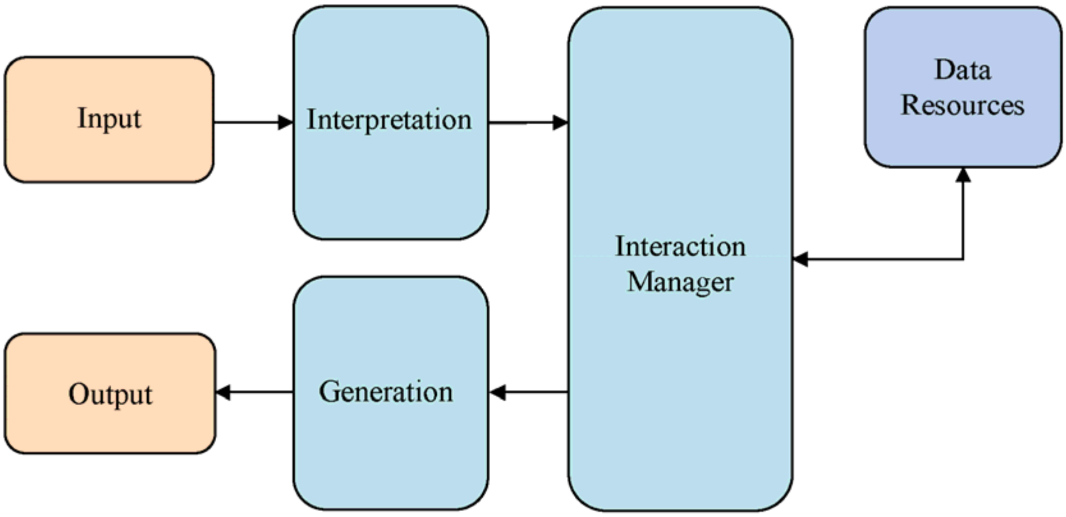

Figure 1: The multimodal interactive framework.

- Input and output – the entry and exit points of the information. In multimodal environments the input may be through various modes such as haptics, keyboard, pointing devices, speech, audio, handwriting, special accessibility devices, Dual-Tone Multi-Frequency (DTMF) signal, etc.
- Interpretation – processes the input using specialized modules for each type of input. In effect, the input is analyzed and its semantic and pragmatic meaning is channeled to the system manager.
- Generation – creates the appropriate output for the system response. It translates from the internal system representation to a usable response for the user. It decides how that information would best be rendered by the most suitable output mode or combination of output modes.

- Interaction manager – it is the most complicated component comprising of several modules that handle the interaction state, the system information, the data resources, the validation and verification of input and response data, the process management, the business model, the user experience, the application functions, the environment variables, and many more.
- Data resources – the data pools, databases, web services and any external information needed or requested by the system in order to fulfill the information requests and data flow.

More information on multimodal dialogue can be found in the latest literature (Kuppevelt et al., 2005; Walster, 2006).

Speech-based Interaction

The use of speech as input/output for interaction requires a spoken language oriented framework that adequately describes the system processes. W3C has defined the Speech Interface Framework to represent the typical components of a speech-enabled web application (Larson, 2000).

Speech-based interaction is context-dependent. The context of the user input is analysed by the system in an attempt to understand the *meaning* and *semantics* within the application domain. The interaction itself is called a *dialogue*. Spoken dialogue interfaces handle human-machine dialogue using natural language as the main input and output. A general depiction of a Spoken Dialogue Interface is shown in Figure 2.

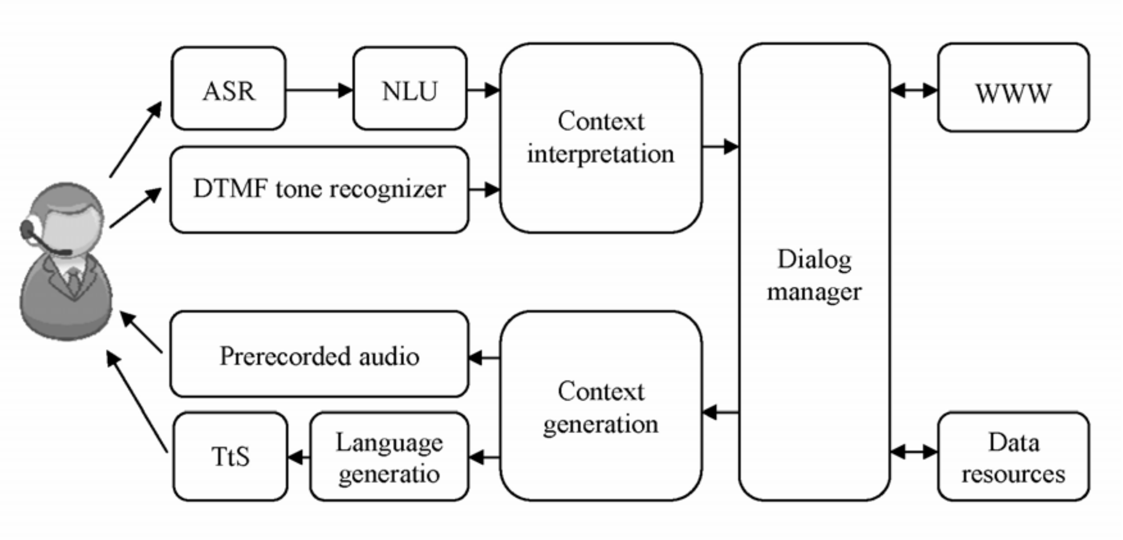

Figure 2: Spoken dialogue interface framework

Broadly speaking, a generic dialogue system comprises of three modules:

- Input – commonly includes automatic speech recognition (ASR) and natural language understanding (NLU). The ASR converts the acoustic user input into text while the NLU parses the text in order to semantically interpret it. Additionally, a DTMF tone recognizer may be included in order to allow for such input.

- Dialogue Management – is the core of the dialogue system. It handles a unique and complete conversation with the user, evaluating the input and creating the output. In order to do that, it activates and coordinates a series of processes that evaluate the user prompt. The dialogue manager (DM) identifies the communicative act, interprets and disambiguates the NLU output and creates a specific dialogue strategy in order to respond. It maintains the state of the dialogue (or belief state), formulates a dialogue plan and employs the necessary dialogue actions in order to fulfil the plan. The DM is also connected to all external resources, back-end database and world knowledge.
- Output – usually includes a natural language generator (NLG) coupled with a text-to-speech synthesizer (TtS). The NLG renders the dialogue manager output from communicative acts to proper written language while the TtS engine converts the text to speech and/or audio. A lot of applications, for the sake of customer satisfaction, use prerecorded audio queues instead of synthetic speech for output. In that case, the dialogue manager forms the output by registering all text prompts and correlating them with prerecorded audio files.

More on the speech-based interaction enabling technologies can be found in respective textbooks (Dybkjær et al., 2007; Dybkjær & Minker, 2008; Tatham & Morton, 2005)

Spoken dialogue web interfaces and voice browsers

Before entering the usability realm, there are several principles and notions governing the spoken dialogue web interfaces and voice browsers. Humans have the ability to communicate with certain complexity. The most natural way of doing so is with the use of natural language. Speech is the direct product of the communication and dialogue denotes the interaction between two or more participants. Non-speech web interfaces use modalities other than speech to communicate. The underlying philosophy of the web interface designer is directly dependent on the mode of communication. In effect, the same service would be designed and implemented in much different way if the hosting platform was a traditional point-and-click web interface than a speech-based one.

For a spoken language dialogue system, the communicative skills of the system are explicitly encoded by a dialogue designer. A good designer defines the core methods that are collectively known as dialogue management according to the given requirements. The design requirements are set by the application functional requirements. All applications have intrinsic *business logic* associated with their basic functions. This means that any application interface should accommodate and handle all functions in a specific way.

Let us consider a request for a typical customer service speech-based application. Such request includes specific directives about handling the various user types, the type of dialogue and attitude for the interaction, the actual tasks that the system should perform (perhaps even comparing to existing services), performance requirements and, most importantly, *acceptance criteria*. The acceptance criteria typically include the performance and reliability factors as well as the user experience evaluation. The latter would most probably count twice as much towards the final product acceptance. It is only fair, after all, for the user experience to be the most valuable factor in a human-machine interface evaluation.

When building a speech-based human-computer interaction system, certain basic modules must be present. The Dialogue Manager is responsible for the system behavior, control and strategy. In general, a dialogue with a machine is a sequential process and contains multiple turns that can be initiated by the machine (system initiative), the user (user initiative), or both (mixed initiative). The ASR and NLU recognize the spoken input and identify semantic values. The language generator and TtS or the prerecorded audio generator provides the system response. The dialogue is usually restricted within the thematic domain of the particular application. The performance of the particular modules is an indication of usability issues. The ASR accuracy and the lack of language understanding due to out-of-grammar utterances or ambiguity hinder the spoken dialogue. Moreover, the lack of pragmatic competence of the dialogue manager (compared to the human brain) and the response generation modules sometimes overcomplicate the dialogue and frustrate the user.

In our analysis, voice browsers can be considered as a subset of the spoken dialogue web interface description. Voice browsers are, by design, system-directed (or even user-directed) dialogue applications with a very limited domain and limited dialogue strategy. They are meant to provide the means to browse information and navigate web documents. In this case, dialogue management complexity is not a demand. In this respect, the usability requirements and evaluation methods for spoken dialogue web interfaces discussed later in this paper also apply to voice browsers.

Interested readers may are refer to spoken dialogue textbooks for further reading (Bernsen et al., 1998; Jurafsky & Martin, 2000; Huang et al., 2001; McTear, 2004).

USABILITY FOR SPEECH-BASED SYSTEMS

The term usability has been used for many years to denote that an application or interface is *user friendly*, *easy-to-use*. These general terms apply to most interfaces, including web interfaces and more importantly speech-based web interfaces. Usability is measured according to the attributes that describe it, as explained below (Rubin & Chisnell, 2008):

- Usefulness – measures the level of *task enablement* of the application. As a side result, it determines the *will* of the user to actually use it for the purpose it was designed for.
- Efficiency – assesses the *speed*, *accuracy* and *completeness* of the tasks or the user goals. This is particularly useful for evaluating an interface sub-system since the tasks may be broken down in order to evaluate each module separately.
- Effectiveness – quantifies the system *behaviour*. It is a user-centric measure that calculates whether the system behaves the way the users expect it to. It also rates the system according to the level of *effort* required by the user to achieve certain goals and respective *difficulty*.
- Learnability – it extends the effectiveness of the system or application by evaluating the user effort required to do specific tasks over several repetitions or time for training and expertise. It is a key measure of user experience since most users expect to be able to use an interface effortlessly after a period of use.
- Satisfaction – it is a subjective set of parameters that the users are asked to estimate and rank. It encompasses the user overall *opinion* about an application based on whether the product meets their *needs* and performs *adequately*.
- Accessibility – in the strict sense, it is not part of the usability description. As a starting point, it is a totally different approach on system design. Accessibility is about access to content, information, and products by everyone, including people with disability. *Design-for-all* is a term that denotes that an application is designed in such way so that everyone can use it to full extent (Stephanidis, 2001). An accessible web site should be

implemented according to specification in order to enable voice browsers to navigate through all available information. An accessible web interface should allow for everyone to use. A blind user, for instance, could use certain modalities for input but the system should never respond by non-accessibly visual content (Freitas & Kouroupetroglou, 2008). Accessibility is a very important and broad discipline with many design and implementation parameters. It can be thought as an extension of the aforementioned usability attributes to the universal user. Universal Accessibility (Stephanidis, 2009) strives to use most modalities in order to make the web content available to everyone. Speech and audio interfaces are used for improved accessibility (Fellbaum & Kouroupetroglou, 2008; Duarte & Carriço, 2008). For example, spoken dialogue systems are considered as key technological factors for the universal accessibility strategies of public terminals, information kiosks and Automated Teller Machines (ATMs) (Kouroupetroglou, 2009). It is mentioned here for completeness; however, it is out of the scope of this work.

Interaction design lifecycle (interfaces) and usability

The basic interaction design process is epitomized by the main activities that are followed for almost every product. There are five activities in the lifecycle of a speech interface (Sharp et al., 2007):

- Requirements specification and initial planning
- Design
- Implementation and testing
- Deployment
- Evaluation

In terms of usability there are three key characteristics pertaining to user involvement in the interaction design process (Sharp et al., 2007):

- User involvement should take place throughout all five stages.
- The usability requirements, goals and evaluation parameters should be set at the start of the development
- Iteration through the five stages is inevitable and, therefore, should be included in the initial planning.

Figure 3 shows how usability generally integrates with the development of a speech-based dialogue interface.

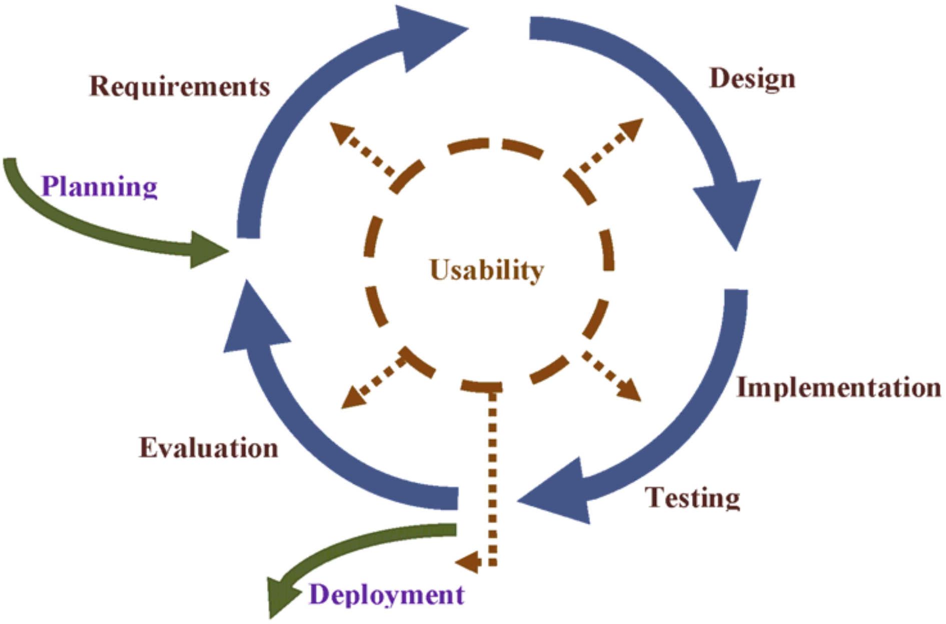

Figure 3: Typical interface lifecycle of a speech-based dialogue system.

Spoken dialogue interfaces may be of three types depending on their design:

a. DTMF replacement
b. Simple system or user-directed question-answering
c. Open-ended natural language mixed-initiative conversational system.

Type (a) systems are the very basic menu-driven interfaces where a static tree-based layout is presented to the user. The user may respond with yes/no and navigate through the menu via options. Such systems are not user-friendly, typically used for very limited domain services, and require patience and time from the user in order to complete a task. The main advantage is that they are very robust, since the user is presented with only a few options at any time, and can only go forward or backwards in the tree-structured menu.

Type (b) systems use more advanced techniques in order to accommodate a more natural interaction with the user. The menus may be dynamic, have confirmation and disambiguation prompts as well as more elaborate vocabulary. Still, the system or the users have to use voice responses within the grammar. Such systems have reusable dialogue scripts for dialogue repair. The small grammars keep the system relatively robust. Such systems are used for most applications at the moment, providing a trade-off between efficiency and robustness.

Type (c) systems are used for large scale applications. These systems are targeted for user satisfaction and naturalness. The users may respond to natural “how may I help you” system prompts with equally natural replies. The utterances may be long, complex and exhibit great variety. The dialogue is dynamic and the demand for successful ASR is high, as is the use of statistical or machine learning methods for interpretation. The dialogue management is task-based, the system creating tasks and plans of actions to fulfil. The users expect high-level natural interaction, a very important element to factorise in usability parameterisation.

It is obvious, now, that each type of design entails particular usability expectations. Each type is expected to excel in certain aspects.

Table 1. Usability impact on spoken dialogue interface development lifecycle

Type	**Requirements**	**Design**	**Implementation**	**Deployment**	**Evaluation**
a	low	medium	low	Low	low
b	medium	medium	low	Low	medium
c	high	high	medium	high	high

Based on the analysis of our-own involvement through the development and testing of a number of nationwide-size spoken dialogue business applications, we present in Table 1 how usability is taken into account in each stage of the product lifecycle. The development of such applications is an iterative process, as mentioned before. Reliant on own experience as described above, we can declare that practitioners in industrial settings agree that usability parameters, as well as testing, are also part of the iterative process. Type (c) systems possess the highest potential for usability integration. In that respect, the remainder of this paper refers mostly to type (c) systems and less to the other two. These days, such systems are the centre of the attention by researchers, developers and customers alike, focusing on advanced voice interaction and high user satisfaction. The use of natural voice response (both acoustic and syntactic) and the natural dialogue flow constitute the state-of-the-art in spoken dialogue interfaces. The web provides the means for the application deployment and the system-world communication, aiming to provide stability and vast amount of available information.

Typical requirements for real-life spoken dialogue interfaces

In systems engineering the term *non-functional requirements* is used to denote the requirements that specify the criteria for assessing the operation of a system, while *functional requirements* define the behaviour. In this context, usability is one of the major functional requirements. Non-functional requirements do not encompass usability per se, however they are effective constraints on the design of the system and may indirectly affect the user experience.

Before the start of the design phase, there are certain accustomed typical requirements pertaining to the areas that the design should focus, i.e. the actual issues that the spoken dialogue system is asked to realize or abide with. Some of them are specifically usability-oriented while others are domain-dependent or generic system-oriented. These typical requirements for Voiceweb interfaces are:

- User satisfaction – Users should be satisfied or very satisfied either as standalone users or comparing their input from using an earlier interface.
- Quality of service offering – Improvement on the quality of the way the requested services/tasks are presented. For example, a large DTMF tree-based dialogue may require the users to navigate through several menu layers to achieve their goal, while a natural language dialogue may identify the initial user request and retrieve the requested service right at the start.
- State-of-the art solution – The system should deploy cutting edge technology.

- Ability to provide customised behavioural or personalised interaction for specific user groups. A common example is the use of a preferred type of interaction (formal, casual, friendly, entertaining, etc.) set specifically for the application domain.
- Complete access to all services or business units that are supported. By design, the system should be able to provide the users the same high quality interaction for all services that the interface is used for.
- Reliability – Extends to the system providing the intended functions continuously without failing.
- Continuity of processing – Also includes problem recovery. In this case a natural language interface should cater for the interaction when a system problem occurs.
- Auditability – Ensures the transparency of the system providing supporting evidence to trace processing of data.
- Performance requirements that describe the capacity of the system to perform certain functions or process certain volume of transactions within a prescribed time.
- Usability-related factors that the operator of a spoken dialogue interface may find prudent to stress upon to the designer.

These requirements are usually followed by a list of mandatory *acceptance tests* that the final system should pass before it is deployed to the web. The format imposed for the acceptance tests is generally comprised of *key performance indicators* (KPIs) for ASR and TtS success. These should be developed by the designer and be available on production to use also for tuning purposes. Furthermore, acceptance tests include task completion evaluation for all requested tasks that are to be tested.

For average size/complexity spoken dialogue interfaces, a magnitude of 10-15 trialists should be sufficient for the acceptance tests. There are two main areas that the tests are carried out in:

1. *Functional assessment* of the system respective modules and functions such as accuracy of information relayed to the user, start/end of dialogue or sub-dialogue flow, service/information provision accuracy, and so on.
2. *User Experience assessment* in terms of:
    a. quality issues
        i. speech or dialogue pause length between activities such as voice request, system search, information retrieval, information relay/output, and prompt delays between responses
        ii. output voice (natural or synthetic) consistency and naturalness for all stages of dialogue as well as in special cases where critical information or explicit help is required
        iii. choice of presenting output voice, clear and non-breaking, during loudspeaker mode or in noisy environments
        iv. correct pronunciation and focus placement in sentences
    b. user interaction
        i. ease of use, navigation through the interface
        ii. instructions and help prompt quality
        iii. smart recovery from misinterpretations or misrecognitions
        iv. disambiguation and confirmation function performance
        v. dialogue flow cohesion
        vi. overall satisfaction.

Since all this information is available to the designer beforehand, it can be put to good use especially during the design. Most of these requirements are the constraints set by the operator so that the design should be built around them. A good design should take those into account in order for the final system to pass the acceptance test assessments.

USABILITY EVALUATION FOR SPEECH-BASED SYSTEMS

Usability evaluation can be formative or summative and thus it can be performed either during or at the end (or near the end) of the development cycle. The methodologies that can be used for that differ in their scope, their main difference being that, when a product is finished (or nearly finished), *usability testing* serves for fine-tuning certain parameters and adjusting others to fit the target user better. During the design phase, usability evaluation methods can be used to probe the basic design choices, the general scope and respective task analysis of a web interface. Some of the most common factors to think about when designing a usability study are:

- Simulate environment conditions closely similar to the real world application use.
- Make sure the usability evaluation participants belong to the target user group
- Make sure the user testers test all parameters you want to measure
- Consider onsite or remote evaluation.

These factors are referenced later in this section.

Methodologies

Usability evaluation for speech-based web interfaces is carried upon certain evaluation methods and approaches on the specific modules and processes that comprise each application. Each approach measures different parameters and goals. They all have the same goal, to evaluate usability for a system, sub-system or module. However, each approach targets specific parameters for evaluation. The main two usability evaluation classes for spoken dialogue systems include the Wizard-of-Oz (WOZ) formative testing (Harris, 2005) and the summative usability testing.

The Wizard-of-Oz formative evaluation

It is a common formative approach that can be used not only for speech-based dialogue systems but for most web applications. It enables usability testing during the early stages by using a human to simulate a fully working system. In the case of speech-based dialogue systems, the human “wizard” performs the speech recognition, natural language understanding, dialogue management and context generation. Cohen et al. (2004) list the main advantages of the WOZ approach:

- Early testing – it can be performed in the early stages in order to test and formulate the design parameters as early in the product lifecycle as possible.
- Use of prototype or early design – eliminates problems arising later in the development such as integration.
- Language resources – Grammar coverage for the speech recognition (ASR) and respective machine learning approaches for interpretation (NLU) are always low when testing a non-finalised product. Low scoring for ASR-NLU may hinder the usability evaluation, however, the use of the human usability expert eliminates such handicap.

- System updates – the system, being a mock-up, can be updated effortlessly to accommodate for changes imposed from the input from the test subjects, making it easier to re-test the updated system in the next usability evaluation session.

The WOZ approach is primarily used during the initial design phase to test the proposed dialogue flow design and the user response to information presentation parameterisation. Since errors from speech recognition and language interpretation are not taken into account, the resulting evaluation lacks the realistic aspect. Expert developers usually know what to expect from the speech recognition and interpretation accuracy because these are domain dependent.

There are two requirements for successful usability testing, the design of the tasks and the selection and training of participants. The participants must be representative of the end-user population, taking into account age, demographics, education. Other criteria may be set depending on the actual application domain, for example users of a specific web site. Moreover, novice and expert users can be recruited in order to provide the means of applying the system design to the worst-case (low experience level) and best-case (high experience level) population.

The participants are required to complete a number of tasks that are carefully selected to test the system performance. In a dialogue system the primary concern to evaluate is the dialogue flow. Two sets of scenarios should be designed, one asking the participants to perform specific actions or pursue predetermined goals and another asking for uncontrolled access of the system pursuing goals of their own choice. The controlled predetermined scenarios are used to evaluate the behaviour of the participants against the behaviour expected by the designer, exposing possible flaws of the design. The uncontrolled interaction is used to evaluate the generic performance of the participants revealing basic design faults, such as non-obvious availability of *help* function or *ambiguous* interaction responses from the system.

The results of the WOZ tests are both from the user subjective feedback and the examination of the objective performance measures. The performance measures include:

- task completion – whether the participants completed the specified tasks that were set within the scenarios successfully,
- efficiency – whether the participants chose the most direct route to the goal, using the predetermined scenario feedback to compare against the optimal path for the same scenario that was expected by the designer,
- dialogue flow – how the participants chose to interact with the system, the number of times the help was requested and how informative it was, as well as the number of times disambiguation, confirmation and error recovery sub-dialogues were enabled.

The subjective input of the participants is recorded through questionnaires that the participants fill in after each task completion as well as at the end of the evaluation. The questions are used to assess the user experience asking about complexity, effort required, efficiency, linguistic clarity, simplicity, predictability, accuracy, suitable tempo, consistency, precision, forgiveness, responsiveness (see Ward & Tsukahara, 2003), appropriateness, overall impression and acceptance of the system, either regarding particular tasks or the full system (Weinschenk & Barker, 2000). The participants are usually asked to mark the level of their agreement to the questions through a 1-to-5 or 1-to-7 scales (for example, 1 being “totally disagree” and 7 “totally agree” and the rest in between), commonly known as Likert scales.

The data are analysed and problems are prioritized in terms of type, severity, and frequency. The subjective feedback also indicates behavioural flaws in the design. Both results enable the designer to take certain action to fix or eliminate those flaws from the design and proceed to implementation.

The summative usability testing of voiceweb systems

At the end of the implementation, pre-final versions of the system should be tested by potential users in order to evaluate the usability. Usability testing at this stage is not much different to WOZ in terms of planning. But now there is no human actor (wizard) but the full system interaction with the user. This means that the ASR, Context interpretation and generation, and TtS are now part of the usability metrics.

At this stage, the usability tests play a much more pivotal role since the development of the system is near completion. There are three distinct purposes for usability testing of a working system: the *development*, *testing* and *tuning*. During the development the users test a nearly finished product, during testing a finished product, and during tuning a finished and already deployed product. Regardless of purpose, the tests focus on all aspects that the WOZ handled as well as several aspects that the WOZ ignored:

- Grammar testing
- Interpretation testing
- Dialogue management/flow
- System response adequacy
- Output speech quality.

For spoken dialogue interfaces, the following 15 objective (both quantitative and qualitative) and subjective usability evaluation criteria have been proposed (Dybkjær & Bernsen 2000):

1. Modality appropriateness.
2. Input recognition adequacy.
3. Naturalness of user speech relative to the task(s) including coverage of user vocabulary and grammar.
4. Output voice quality.
5. Output phrasing adequacy.
6. Feedback adequacy.
7. Adequacy of dialogue initiative relative to the task(s).
8. Naturalness of the dialogue structure relative to the task(s).
9. Sufficiency of task and domain coverage.
10. Sufficiency of the system's reasoning capabilities.
11. Sufficiency of interaction guidance (information about system capabilities, limitations and operations).
12. Error handling adequacy.
13. Sufficiency of adaptation to user differences.
14. Number of interaction problems (Bernsen et al. 1998).
15. User satisfaction.

Bernsen & Dybkjær (2000) have proposed the use of the *evaluation templates*, i.e. “models of what the developer needs to know in order to apply an evaluation criterion to a particular property of a Spoken Language Dialogue System or component”, in their methodology as best practice guides. Later, they formed a set of guidelines for up-to-date spoken dialogue design, implementation and testing, covering seven major aspects: informativeness, truth and evidence, relevance, manner, partner asymmetry, background knowledge, repair and clarification (Bernsen & Dybkjær, 2004). These aspects can be used as the basis for usability testing strategies and for evaluation frameworks (Dybkjær & Bernsen, 2001; Dybkjær et al., 2004). One of them is the PARADISE evaluation framework (Walker et al., 1998; Hajdinjak & Mihelic, 2006) with general models developed for it (Walker et al., 2000).

As with WOZ, usability testing needs participants. The recruitment procedure is pretty much the same as described earlier in WOZ, with a few additional parameters. The participants use the real system, which means that, at this stage, functional parameters in speech recognition and speech synthesis should be tested, measured and decided upon. There is extensive work on the comparison of usability evaluation feedback between in-house recruited participants versus real users. The differences are mainly on the use of barge-in, explicit requests, the use of help and dialogue acts preference/selection (Ai et al., 2007). Moreover, parameter measurements in speech recognition rejection, choice of interaction ending, help and repeat requests, user interruptions and silence timeouts, show that there users behave differently in the first month of their interaction. After that, the users become accustomed to the system, experienced and their behaviour becomes more or less stabilised (Turunen et al., 2006).

Kamm et al. (1998) stress the importance of a successful quick tutorial on the users before using a speech-based application. They show that the initial user experience can be ensured when the first-time users are trained on the use of the system. The user satisfaction and the system performance were significantly improved in this case. Also, there is significant differentiation between onsite and remote evaluation. Participants recruited for onsite evaluation know that they are required to evaluate the system and may behave unexpectedly or even use extreme caution when using the system, a behavior much dissimilar to that of real users.

Apart from task completion and dialogue flow, depending on the domain, as a general rule, functional measurements should be recorded for at least the following indicative parameters:

- Average call duration
- Peaks and valleys of usage per hour per day
- Successful speech recognitions
- Misrecognitions
- No-inputs
- Timeouts
- Rejections
- Early hang-ups
- Successful interpretations
- Failed interpretations (no-matches)
- Successful repairs
- Failed repairs.

Performance metrics may be derived by calculating parameters such as the number of user and system turns, elapsed time, number of help requests, number of timeout prompts, mean ASR accuracy and number of speech recognition rejections (Kamm & Walker, 1997; Kamm et al. 1999). Generally, the above parameters can indicate functional problems with the application and the degree that each of those affects the user experience (Walker et al., 1999). Furthermore, the data can be automatically processed using appropriate methods (Hartikainen et al., 2004), used to train models for evaluation-based problem prediction that leads to an adaptive spoken dialogue interface (Litman et al. 1998; Litman & Pan, 2002).

The user subjective feedback is also very important at this stage. It illustrates the user experience as perceived by the user (to be compared with automatically-derived user experience level from the performance metrics analysis) and stresses the points where the users were not satisfied. By analysing the questionnaires down to the usability factors (Larsen, 2003) the designer can even, to an extent, predict the quality and usability of spoken dialogue services (Moller et al. 2006; 2008).

Functional assessment and usability testing for deployed systems

In the recent years, there has been a need for designers to examine how a deployed system behaves and be able to alter the system output in real time, for testing, updating (observing user behavioural transformation after major updates) and fine-tuning. Expensive, large-scale systems enter the market by incremental deployment to groups of real users. Between the increments, the designing team evaluates the functional aspects of the system as well as the user experience of first-time or advanced users. That is also the case for when major updates or supplemental features are implemented and released to target users for the first time. To do that, the application framework should be able to support a WOZ-like mode that allows the designer to intervene on the dialogue process during an interaction. This is especially suitable for spoken dialogue web interfaces where the implementation of such feature would enable the web interface designer to try out alternative routes to the interaction tasks and goals. Figure 4 shows the layout of a real-time wizard-based testing environment for already deployed systems.

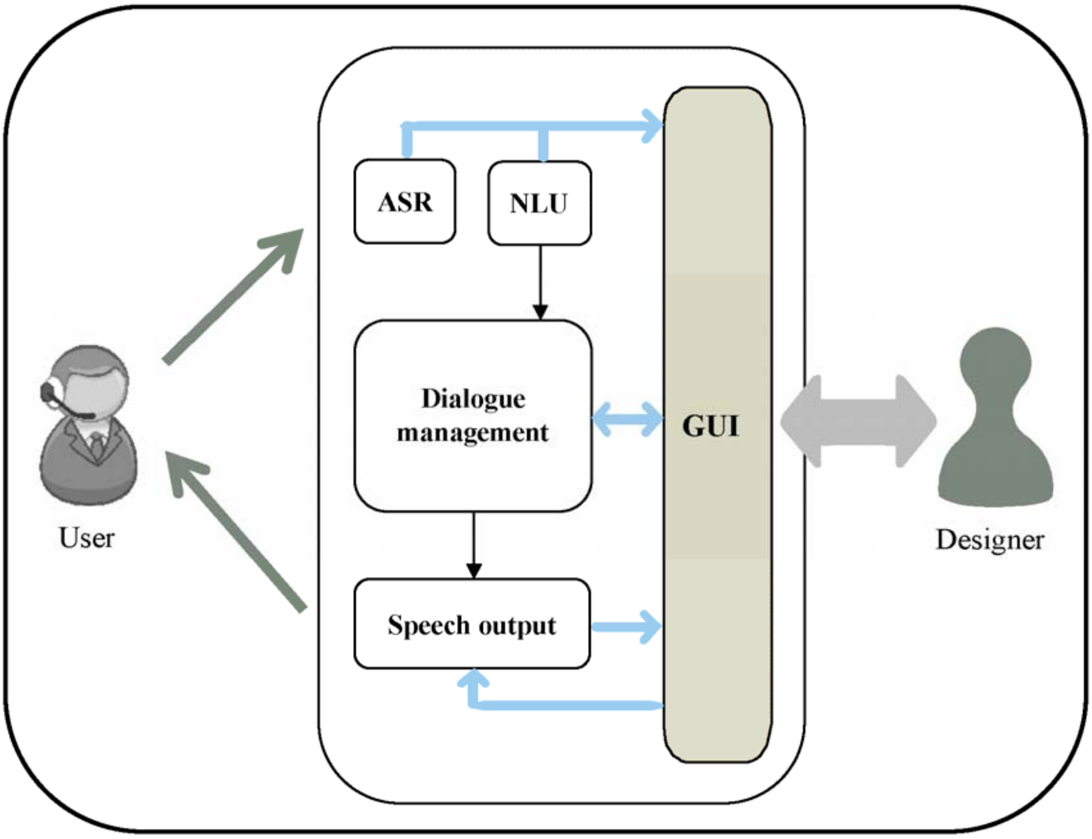

Figure 4: Wizard-based usability testing for a deployed Spoken Dialogue Interface.

Using this framework any member of the design and implementation team can perform real time inspection of the interaction, watching for speech recognition, interpretation and dialogue management parameters, observing the dialogue flow, history, user input and system responses. They can intervene at any time by involving themselves in the decision process of the dialogue manager, overriding the system response to change an interpreted request or redirect the dialogue flow. Even more, the wizard may validate the user input or trigger a disambiguation sub-dialogue to test the user and system responses. This approach has been successfully applied in the case of a real-life Voice User Interface application of a call-center (Spiliotopoulos et al, 2009).

CONCLUSIONS

Natural language dialogue web interfaces interact with the users in a natural and convenient way. Usability integration and evaluation is a fundamental requirement for such delicate interaction. This paper presented the guidelines and methodologies for designing, developing, testing and deploying usable spoken dialogue interfaces. As technology advances, the use of natural language interfaces requires more sophisticated approaches to enhance the user experience with high-level linguistic input-output and advanced dialogue management. Such endeavour necessitates the use of equally advanced usability methodologies during all stages of the system development.

This work discussed the theory behind usability evaluation methods and approaches as well as the frameworks that incorporate usability evaluation and testing for speech-based web interfaces. Moreover the major methods used for formative and summative evaluation have been examined and analyzed in the context of voiceweb systems.

Usability evaluation is used during the requirements analysis, design, implementation, testing deployment and evaluation of speech-based dialogue interfaces. The requirements/design stages of development are benefited by the input of potential user groups. The designer can use the feedback to formulate an interface that provides all the requested services in a suitable user-approved interaction design. The finished or nearly finished systems are put to test in order to assure the quality of interaction, as well as performance, completeness and naturalness. At this stage the functional tests are also performed and included in the usability evaluation.

Finally, the already deployed systems need to be re-evaluated at any time during their life on the market for either testing or updating purposes. At this point the existing functional and non-functional parameters are taken for granted. The new or updated technologies or dialogue flow can be evaluated by the designer or tester using a graphical user interface, a front-end for dialogue overview and control that enables the designer to monitor all dialogue processes and override, manipulate, confirm or dismiss any input or output of the system. The importance of such application framework as part of the integrated design and implementation approach facilitates a professional after-release support and development for the ever growing requirements of spoken dialogue web interfaces.

ACKNOWLEDGEMENTS

The work described in this paper has been funded by the Special Account for Research Grants of the National and Kapodistrian University of Athens under the KAPODISTRIAS program.

REFERENCES

Ai, H., Raux, A., Bohus, D., Eskenazi, M., & Litman, D. (2007). Comparing spoken dialog corpora collected with recruited subjects versus real users. In *Proc. of the 8th SIGdial workshop on Discourse and Dialogue* (pp. 124–131).

Bernsen, N. O., Dybkjaer, H., & Dybkjaer, L. (1998). *Designing Interactive Speech Systems: From First Ideas to User Testing*. New York: Springer-Verlag.

Bernsen, N. O., & Dybkjær, L. (2000). A Methodology for Evaluating Spoken Language Dialogue Systems and Their Components. In *Proc. 2nd International Conference on Language Resources & Evaluation - LREC 2000* (pp.183-188).

Bernsen, N. O., & Dybkjær, L. (2004). Building Usable Spoken Dialogue Systems: Some Approaches. *Sprache und Datenverarbeitung 28*(2), 111-131.

Cohen, M., Giancola, J. P., & Balogh, J. (2004*). Voice User Interface Design*. Boston: Addison-Wesley Professional.

Duarte, C., & Carriço, L., (2008). Audio Interfaces for Improved Accessibility. In S.Pinder (Ed.), *Advances in Human Computer Interaction* (pp. 121-142). Vienna, Austria: I-Tech Education and Publishing.

Dybkjær, L., & Bernsen, N. O. (2000). Usability Issues in Spoken Language Dialogue Systems. *Natural Language Engineering*, *6*(3-4), 243-272.

Dybkjær, L., & Bernsen, N. O. (2001). Usability Evaluation in Spoken Language Dialogue Systems. In. Proc. *ACL Workshop on Evaluation Methodologies for Language and Dialogue Systems*, (pp. 9-18).

Dybkjær, L., Bernsen, N. O., & Minker, W. (2004). Evaluation and Usability of Multimodal Spoken Language Dialogue Systems. *Speech Communication*, *43*(1-2), 33-54.

Dybkjær, L., Hemsen, H., & Minker, W. (Eds.) (2007). *Evaluation of Text and Speech Systems*. Berlin: Springer-Verlag.

Dybkjær, L., & Minker, W. (Eds.) (2008). *Recent Trends in Discourse and Dialogue*. Berlin: Springer-Verlag.

Fellbaum, K., & Kouroupetroglou, G. (2008). Principles of Electronic Speech Processing with Applications for People with Disabilities. *Technology and Disability*, *20*(2), 55-85.

Freitas, D., & Kouroupetroglou, G. (2008). Speech Technologies for Blind and Low Vision Persons. *Technology and Disability*, *20*(2), 135-156.

Hajdinjak, M., & Mihelic, F. (2006). The PARADISE evaluation framework: Issues and findings. *Computational Linguistics*, *32*(2), 263–272.

Harris, R. A. (2005). *Voice Interaction Design: Crafting the New Conversational Speech Systems*. San Francisco: Morgan Kaufmann.

Hartikainen, M., Salonen, E.-P., & Turunen, M. (2004). Subjective Evaluation of Spoken Dialogue Systems Using SERVQUAL Method. In *Proc. 8th International Conference on Spoken Language Processing - ICSLP*, (pp. 2273-2276).

Huang, X., Acero, A., & Hon, H.-W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Upper Saddle River, NJ: Prentice Hall PTR.

Jurafsky, D., & Martin, J. H. (2008). *Speech and Language Processing. An Introduction to Natrural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ: Prentice-Hall.

Kamm, C. A., & Walker, M.A. (1997). Design and Evaluation of Spoken Dialogue Systems. In Proc. *IEEE Workshop on Automatic Speech Recognition and Understanding*, (pp. 14–17).

Kamm, C. A., Litman, D. J., & Walker, M. A. (1998). From novice to expert: The effect of tutorials on user expertise with spoken dialogue systems. In *Proc. 5th International Conference on Spoken Language Processing - ICSLP*.

Kamm, C. A., Walker, M. A., & Litman, D. J. (1999). Evaluating spoken language systems. In *Proc. Applied Voice Input/Output Society Conference - AVIOS*, (pp. 187–197).

Kouroupetroglou, G. (2009). Universal Access in Public Terminals: Information Kiosks and Automated Teller Machines (ATMs). In C. Stephanidis (Ed.), *The Universal Access Handbook*, (pp. 761-780). Boca Raton, FL: CRC Press.

Larsen, L. B. (2003). Issues in the Evaluation of Spoken Dialogue Systems using Objective and Subjective Measures. In *Proc. 8th IEEE Workshop on Automatic Speech Recognition and Understanding -ASRU*, (pp. 209-214).

Larson, J. A., Raman, T. V., & Raggett, D. (2003). *W3C Multimodal Interaction Framework.* Retrieved August 2, 2009, from http://www.w3.org/TR/mmi-framework/

Larson, J. A. (2000). *Introduction and Overview of W3C Speech Interface Framework*. Retrieved August 2, 2009, from http://www.w3.org/TR/voice-intro/

Litman, D. J., Pan, S., & Walker, M. A. (1998). Evaluating Response Strategies in a Web-Based Spoken Dialogue Agent. In *Proc. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conf. on Computational Linguistics (ACL/COLING)*, (pp. 780–786).

Litman, D. J., & Pan, S. (2002). Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, *12*(2-3), 111–137.

McTear, M. F. (2004). *Spoken Dialogue Technology: Towards the Conversational User Interface*. London: Springer-Verlag.

Moller, S., Englert, R., Engelbrecht, K., Hafner, V., Jameson, A., Oulasvirta, A., Raake, A., & Reithinger, N. (2006). MeMo: Towards Automatic Usability Evaluation of Spoken Dialogue Services by User Error Simulations. In *Proc. 9th International Conference on Spoken Language Processing - ICSLP*, (pp. 1786-1789).

Moller, S., Engelbrecht, K., & Schleicher, R. (2008). Predicting the quality and usability of spoken dialogue services. *Speech Communication, 50*(8-9), 730-744.

Rubin, J., & Chisnell, D., (2008). *Handbook of Usability Testing: Howto Plan, Design, and Conduct Effective Tests*. Indianapolis, IN: Wiley Publishing, Inc.

Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction Design: Beyond Human-Computer Interaction*. West Sussex, UK: John Wiley & Sons, Inc.

Spiliotopoulos, D., Stavropoulou, P. & Kouroupetroglou, G. (2009). Spoken Dialogue Interfaces: Integrating Usability. *Lecture Notes in Computer Science*, 5889, 484-499.

Stephanidis, C. (Ed.). (2001). *User Interfaces for All: Concepts, Methods and Tools*. Mahwah, NJ: Lawrence Erlbaum Associates.

Stephanidis, C. (Ed.). (2009). *The Universal Access Handbook*. Boca Raton, FL: CRC Press.

Tatham, M., & Morton, K. (2005). *Developments in Speech Synthesis*. West Sussex, UK: John Wiley & Sons, Inc.

Turunen, M., Hakulinen, J., & Kainulainen, A. (2006). Evaluation of a Spoken Dialogue System with Usability Tests and Long-term Pilot Studies: Similarities and Differences. In *Proc. 9th International Conference on Spoken Language Processing - INTERSPEECH* (pp. 1057—1060).

van Kuppevelt, J., Dybkjær, L., & Bernsen, N. O., (Eds.). (2005). *Advances in natural multimodal dialogue*. Dordrecht, The Netherlands: Springer.

Wahlster, W. (Ed.). (2006). *SmartKom: Foundations of Multimodal Dialogue Systems.* Berlin: Springer-Verlag.

Walker, M. A., Litman, D. J., Kamm, C. A., & Abella, A. (1998). Evaluating spoken dialogue agents with PARADISE: Two case studies. *Computer Speech and Language, 12*(3), 317-347.

Walker, M. A., Borland, J., & Kamm C. A. (1999). The utility of elapsed time as a usability metric for spoken dialogue systems. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop - ASRU*, (pp. 317–320).

Walker, M. A., Kamm, C. A., & Litman, D. J. (2000). Towards developing general models of usability with PARADISE. *Natural Language Engineering*, *6*(3-4), 363-377.

Ward, N., & TsukaharaW. (2003). A Study in Responsiveness in Spoken Dialog. *International Journal of Human-Computer Studies, 59*, 603–630.

Weinschenk, S., & Barker, D. T. (2000). *Designing effective speech interfaces*. New York: John Wiley & Sons, Inc.