A User-Based Method for Speech Interface Development

Yael Dubinsky, Tiziana Catarci, and Stephen Kimani

Department of Computer and Systems Science
University of Rome "La Sapienza"
Via Salaria 113, 00198 Rome, Italy
{dubinsky,catarci,kimani}@dis.uniroma1.it

Abstract. There is a consensus on the significance and high contribution of user involvement in the process of user interfaces development. However, there is no standard way to implement user involvement in software development processes. Dealing with speech-based interfaces that involve vocal interaction of speaking and hearing, the need of user involvement is increased. In this paper, we focus on the characteristics of speech interface development and suggest a user-based method that enables continuous user evaluation. We illustrate the method implementation in two different software projects that contain speech interfaces.

Keywords: user-based development method, speech interface development.

1 Introduction

Speech is the most natural medium to interact with a computer. The main advantage in using a speech interface is that it enables a hands free control. Though there is a significant progress with speech systems, the main disadvantage is its accuracy that suffers from technological issues such as poor signal and background noise as well as language-recognition issues such as dealing with accents and homophones [3; 7; 11].

Speech as an interaction medium is slow for presenting information. It is also transient and thus difficult to review or edit. It also can easily cause cognitive overload and thus interfere with tasks that require significant cognitive resources [10]. Speech input, especially in an open or public setting can exhibit reduced privacy, making users feel less secure in the process [7]. There are also challenges concerning possible distractions and disruptions such as due to noise pollution.

Nonetheless, as indicated by Shneiderman in [10], speech interfaces are sometimes helpful for hands-busy, eyes-busy, mobility-required, or hostile environments. Moreover, speech has been shown to be useful for store-and-forward messages, alerts in busy settings, and input-output for visually-impaired, blind and physically-impaired users. According to Maguire: "speech interfaces can important offer benefits to users with visual impairments, and in a commercial study carried out by the author, a small sample of visually impaired users were very positive about the concept of a speech-based bank machine." [7]. The accuracy of dictation input has been on the increase. However, adoption outside the disabled-user community has been slow compared to visual interfaces [10].

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2007, LNCS 4554, pp. 355–364, 2007.

When developing speech interfaces one should consider the right way to include the user perspective in the process. In traditional software development approaches, when user-centric techniques are used (if at all), the design of the system is refined according to the user evaluation and mainly during the design phase. In contemporary software development methods, the design phase is merged with the coding and testing phases along the entire development process, and therefore an updated approach should be adopted for the user involvement.

In this paper, we suggest a new perspective to evaluating user interfaces and illustrate it in our plan of evaluating speech interfaces. Our main contribution is our approach to the evaluation process that basically says that this process exists as long as the product development process is, and the users as well as experts in human-computer interaction are involved with the evaluation and improved design. In what follows we describe the challenges in evaluating speech interfaces and suggest guidelines for this evaluation based on existing evaluation techniques (Section 2). In Section 3 we present the method to combine the software development approach with the user evaluation one, and illustrate it in two case studies in Section 4. In Section 5 we summarize.

2 Speech Interface Development: The User Perspective

The evaluation of user interfaces aims at improving the interfaces design and to ensure the accessibility and functionality of the system as per users' requirements [4]. Ideally, the evaluation is based on real users' experience as well as human-computer interaction experts, and performed in an iterative manner in order to ensure feedback till the design is finalized. In this part we examine evaluation techniques for speech interfaces.

We review four existing evaluation techniques for user interfaces and suggest how they can be used for speech interfaces design and evaluation. The first two techniques are performed in the absence of the user and they are the cognitive walkthrough and the heuristic analysis techniques. The second two techniques are performed by the users and they are the wizard-of-oz and the think aloud techniques. For each technique we briefly describe its principles and suggest our guidelines with respect to the evaluation of speech interfaces.

The Cognitive Walkthrough technique. Using this technique we explore what the user thinks when first using the interface without any training [9; 4]. The evaluator knows who the users are and based on a paper or working prototype of the interface selects a task to evaluate. The evaluator then should provide a well-explained reliable narrative that audits the actions the user performs in order to complete the task in addition to the motivation for each action. If the narrative is not reasonable in some sense, it indicates that a problem exists with the interface. The advantages of the cognitive walkthrough technique are that it assists in refining the requirements and that there is no need of working software to perform it. The disadvantages are that the evaluator may not represent the real users of the system, sometimes there is no enough time to go over all tasks, and there are cross-tasks situations that are not examined since each task is stand-alone examined.

With respect to speech interface evaluation, it is clear that when there is only paper prototype the cognitive walkthrough technique lacks. For such interfaces the evaluator should experience by hearing in order to provide a narrative for specific tasks. For example, without hearing, problems that relate to voice clarity as input and output for the system can be disregarded. When using the working prototype for the evaluation, the evaluator hearing skills and the noise of the inspection settings are to be considered. It is expected that the narrative that is provided by the evaluator includes utterances that are said or heard during the inspection.

The Heuristic Analysis technique. Using this technique we evaluate the user interface according to nine (that later extended to ten) defined heuristics that guide us with the design decisions [8]. A group of evaluators use the heuristics in order for each of them to identify and rank severity of problems with a paper or working prototype of the interface. Then all problems and ranks are combined by an individual or as preferred by group activity. The advantages of the technique are that by few general guidelines we can reach many interface problems and that it does not involve much effort. The disadvantages are that the evaluator may not represent the real users of the system and that there are tasks that can be disregarded by all evaluators. In what follows we provide the heuristics each followed by our suggestion for speech interface evaluation and design guidelines.

- Simple and natural dialog – natural flow of task activities without irrelevant or rarely used information. Regarding speech interface evaluation this means that the language used as input and output as well as each sequence of utterances is clear, short, simple and concise.
- Speak the user's language - use words and concepts from the user's world as opposed to system-specific engineering terms. This heuristics seems to fit speech interface evaluation as is.
- Minimize user memory load - the user does not need to remember things from one action to the next, and the information should be left on the screen until it's not needed. In speech interface evaluation this means that the interface should provide feedback by repeating the voice that was heard and using recording enables a series of past utterances [1]. Also, the system should provide the possibility to undo utterances, and to enable deletion of old audio items.
- Be consistent - similar action sequence in one part of the system should give the user the same results when applied in other places of the system. Consistency in speech interfaces refers for example to the way the speech recognizer works, which accents are accepted, and the way ambiguity is handled.
- Provide feedback – The interface provides feedback as per users' actions. For speech interface evaluation, it should be noticed that the feedback expected to be heard but if the task itself uses the audio device the feedback should be provided otherwise (by a special distinct tone or if no other choice by screen output).
- Provide clearly marked exits – Exit is correctly enabled in inner parts of the system. For speech interface evaluation, it is recommended to use a short utterance composed of one syllable in order to provide quick navigation and specifically exits.

- Provide shortcuts – an expert way-of-working is provided. For speech interface evaluation, it is expected to have a way to define by record own customized utterances in order to shorten sequences of actions.
- Good error messages – specific and clear messages are provided as error messages. For speech interface evaluation, error messages are actually voices or utterances that differ from each other in order to indicate different error.
- Prevent errors – minimize as much a possible the errors that can occur. For speech interface evaluation, more errors mean more output voices that can be referred to by the user as noise pollution, therefore should be prevented.

The Wizard of Oz technique. Using this technique we fake as if there is a system and simulate an experiment with real users who are not aware of course to the fake. A hidden person provides with the feedback the users should see. The advantages of this technique are that no need for real implementation of the system, and real users are the participants. The disadvantages are that such a setting usually needs a substantial implementation effort and it can be considered controversial because of not exposing the experiment setting to the participants. With respect to speech interface evaluation, this technique was successfully used [3; 1].

The Thinking Aloud Method. Using this technique we ask the user to tell us own thoughts while observing the user's work with the interface so we will "hear the user's thoughts". Analyzing the user actions and thoughts, the evaluator can check whether the sequences fit, and expose interface problems that cannot be revealed by examination of user actions only [4; 6]. The advantages of this technique are that it is simple and can provide new perspectives. The disadvantages are that it is subjective and relies on thoughts description that can mislead. With respect to speech interface evaluation, this technique is not simple but rather complicated. The user is asked to work with the system so speaks and listens as well as to speak about own thoughts, and the observer needs to distinguish between utterances of using the interface and these of the user's thoughts.

3 The Concept of a User-Based Development Method

The main concept that we suggest is that user evaluation should exist during the entire software development process. This concept as well as other suggestions that appear in this section was emerged in light of the principles of the agile approach and its derived methods[1].

The method of work as part of the evaluation process has few principles. The first is *user collaboration*. This principle is not new with respect to evaluation and design *with* users. The difference is that in the last decade when agile methods become more common, the design phase is no longer a phase but rather an on-going activity as part of each short iteration of few weeks of development. Therefore, as customer collaboration is accepted and fostered, the same is for the real users of the system. Given a continuous users feedback provide us with the second principle of *dealing*

[1] See the agile manifesto at http://agilemanifesto.org/

with changes in users' needs, since when design and user evaluation are on-going conducted, we can easily cope with changes in users' needs during the entire development process.

The background of the third principle is the analogy between interface evaluation and software testing. For many years software testing was neglected since it was mainly at the end of the software cycle after detailed design and implementation, when time and cost does not easily permit investing enough effort. This situation influenced the level of quality and the customer satisfaction[2]. The solution of the agile approach is to perform exhaustive automated testing from the beginning of development, and request the customer to define acceptance tests together with the team. Adopting this solution, we suggest the users to be engaged with the development process and assist in *defining usability automated tests* that enable fast feedback to developers also when users absence, e.g., [2].

While the method described can be used in any software process and interface evaluation, we illustrate in one case study (Section 4) the use of automated measurements for speech interface evaluation.

In light of the above, we suggest that the evaluation process is iterative and inductive. Meaning, in order to ensure the evaluation existence, we should iteratively *understand the users' needs, provide evaluation measures* with respect to users' needs, and *improve the interface design and implementation* in a way that each such cycle refines our product. Figure 1 presents a schematic illustration of this concept. We note that challenges in speech interface evaluation can be further detailed in future work with respect to each of the aforementioned challenges, e.g. delving into the details of users' cognition, memory, and attention.

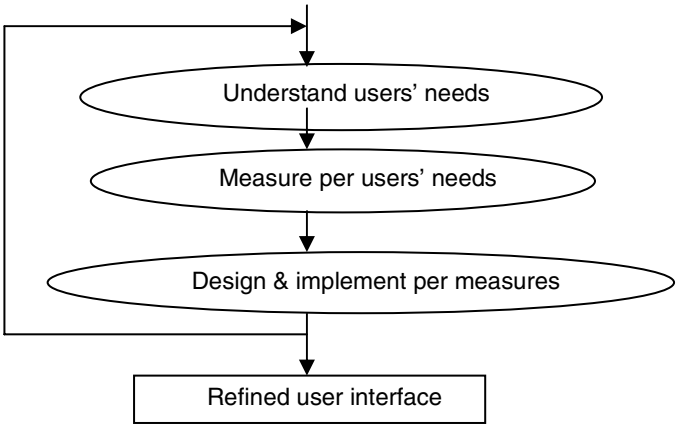


Fig. 1. A schematic illustration of the evaluation process

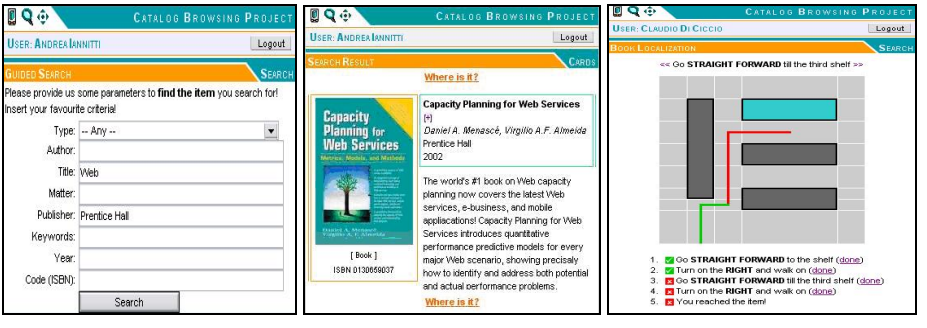
[2] See for example, Mullet, D. (July, 1999). The Software Crisis, Benchmarks Online - a monthly publication of Academic Computing Services **2**(7) – that 75% of enterprise software products are not in use or do not fit customer's requirements.

4 Case Studies

In this part we illustrate the method that is presented in Section 3 using evaluation data that was gathered in two case studies. The first data set is from the catalogue browsing project in which a speech-based mobile interface to a digital library was developed [5]. The second data set is from an entrance control project in which a speech and video interface is developed to control the entrance to a specific laboratory[3]. In this section we focus only on the speech interface development and evaluation of these projects.

4.1 Data Set 1

The interface that was developed in this project is a speech-based mobile interface to a digital library. The interface enables vocal commands for artefact searching and its localization. Speech input is enabled for navigating the application, and speech output is enabled for positioning instructions for artefact localization in a physical library. Figure 2 presents screenshots of the *Guided Search* interface (2a), *Search Result* interface (2b) and the *Book Localization* interface (2c).



(a) **(b)** **(c)**

Fig. 2. Screenshots from the catalogue browsing project

Following the user-based concept that is presented in Section 3, the evaluation process of this interface is composed of evaluation iterations that each examines the artefacts of the previous development iteration and results in design changes for the current or next development iteration. The 1st development iteration provides its artefacts. During the 2nd development iteration, the 1st evaluation iteration took place to evaluate and reflect on the artefacts produced in the 1st development iteration and further to decide upon changes that should be introduced. During the 3rd development iteration, the 2nd evaluation iteration took place to evaluate and reflect on the artefacts produced in the 2nd development iteration, and so on. Each iteration is of 3-5 weeks and as aforementioned the first release was composed of 4 such iterations.

[3] The first project was developed in the University of Rome "La Sapienza". Teh second project was developed in the Technion – Israel Institute of Technology.

We focus on a specific experiment to evaluate the speech aspect of the interface. We conducted a *within experiment* with six participants who are computer science students in different levels, 3 male and 3 female. The experiment task includes login to the system, search activities and book localization activity. The task can be performed using speech (S) or without speech (non-S). Each of the participants performed the task in both modes S and non-S, while 3 participants follow S and then non-S and 3 follow non-S and then S. Further, before starting the experiment, each participant filled an attitude questionnaire and received ten-minute training on how to use the interface. After the experiment each of the participants filled a questionnaire to reflect on his/her activities.

In what follows we present the experiment qualitative and quantitative data. Table 1 presents the answers of the participants to some initial attitude questions with respect to speech aspect, where SD means that the participant Strongly Disagrees with the statement, D means disagree, A means agree, and SA means strongly agree.

Table 1. Participants attitude to speech interfaces

Statement	*SD*	*D*	*A*	*SA*
I like interfaces with speech features	1	1	3	1
I have experience with speech interfaces	1		5	
I use speech interface when I can	1	3	2	
People whom I know do not like speech interfaces		4	1	
Speech interfaces are slow	1	2	1	1
I feel uncomfortable with speech interfaces		2	4	
Speech interfaces are fun	1		4	1
Speech interfaces are annoying		4	2	
I expect to use more speech interfaces in the future			3	3
I prefer interfaces that do not include speech		3	3	

We note that when for a specific statement the sum of answers is less than 6, it means that some participants did not answer on this one. As can be observed, the attitudes with respect to speech interfaces are mixed and do not follow a consistent approach. Though speech interfaces are fun they are also annoying, and though participants like them, they do not always prefer them.

The same questionnaire had some open questions asking the participants to provide features that they consider important to be included in speech interfaces, advantages and disadvantages of speech interfaces, and a personal scenario that happened to them when using such interfaces. Following are some of the expressions of participants answering those questions:

- "[consider important] using realistic voices"
- "…I had to provide some information to an automatic operator – it was boring waiting for its answers"
- "[disadvantage] it can take several minutes to interact with speech interfaces"
- "[advantage] they can become friendly"

Table 2. Averaged search time (in minutes)

Group	*Averaged search duration*	*Averaged Non-S search duration*	*Averaged S search duration*
Non-S → S	54.66	28	81.33
S → Non-S	26.58	14	39.16

Examining the answers, two main categories are observed which are user interaction and user friendly. One phenomenon that was found is that participants see speech interfaces as both friendly and not friendly, or as both fast and slow. For example one participant answers the following in two consecutive rows, "[advantage] faster than normal interfaces"; "[disadvantage] a user may wait too long before achieving [his/her] purpose".

After filling the questionnaire, the participants receive one-page users` guide and when completed to read with no more questions, they received the task page according to their appropriate experiment order of S and non-S. An automatic time measure, which was developed as part of the system, provides us with the time stamps of the login / logout and with the time stamps of each search start /end. Table 2 presents the averaged time in minutes that was invested on the two search activities by both experiment groups together with its division per mode.

As can be observed, the S→Non-S group performed the entire task almost twice faster than the Non-S→S group. When looking into the data of speech and non-speech per each group, we see that the participants in both groups performed the speech task slower then the non-speech task. This implies that although the speech task required more time from the participants, they learned better the system when first using it with the speech option.

After completing the task, participants were asked to fill a questionnaire to reflect on their own activities. Table 3 presents their level of agreement to some statements. As can be observed, most participants find it hard to use the interface in its current stage, though it was fun and they expect to such interfaces in the future.

Table 3. Participants reflect after using the interface

Statement	*SD*	*D*	*A*	*SA*
I like searching using speech commands		4	2	
I have experience with vocal GPS	1	2	3	
I prefer to work with the silence mode		2	4	
People will feel that the speech mode is too slow		2	4	
I feel uncomfortable with the system I use	1	1	3	1
It was fun			4	2
It was annoying	1	4	1	
I expect I will see such systems in the future			6	

The same questionnaire had some open questions asking the participants to describe what they liked with the interface, what are the problems they have encountered, their severity ranking between 1– not so important and 5– very important, and to recommend on how to deal with the specific problem. Following are some of the expressions of participants answering those questions:

- "I expect the system to vocally recognize also the value I want to search"
- "It was easy to use; Funny to use"
- "[rank 4] Instructions too fast"
- "It's been a new experience to me"
- "[rank 3] too sensitive to pronunciation"
- "[rank 5] unstable"
- "[I like] the GPS system"
- "I like activating commands by voice"
- "[rank 5] sometimes it doesn't understand what I say"
- "[rank 5] I have to repeat"
- "[recommendation] try to translate to Italian; it should be more flexible with pronunciation"
- "[I like] moving the cursor by speaking"

Examining the answers we learned that we should focus on some improvements that concerns with implementing speech for all interface features and improving the on-line usage information. This is based on our observation that when users are introduced to a speech-based interface they expect it to be fully speech-based meaning no using of keyboard at all. Further, they expect to receive vocal on-line help to assist them in the process of using the application.

4.2 Data Set 2

The following data is smaller in scope than the first data set and relates to a single first iteration of user evaluation. It is presented here in order to demonstrate *thinking aloud* evaluation data of a speech interface. The project goal is to develop an entrance control system that is based on the identification of faces from a pictures pool. The user who wants to enter activates the system by visual indication when he/she stands in front of a scanner, and by giving a vocal command that starts the process. The system warns the user not to move before taking a series of pictures, and announces the result whether the entrance is granted or denied after comparing with the pictures in its pool.

A thinking aloud experiment was conducted in the first evaluation iteration. 11 participants were asked to use the system, i.e., each of them asked the system to identify his/her face in order to grant the entrance. The participants were asked to report what they think and feel during this experience. Following is the evaluation data. 5 of the participants had the following problem: they gave the vocal command but thought that it was not accepted by the system. They start to give the vocal command again, but meanwhile the scanner took the series of pictures, the process fails, and the scanner starts again due to the second command. In 2 of these 5 trials, the process ends in success, meaning the system could announce the result. 5

participants reported that using the microphone to give the vocal command enabled them with good control over the process. One participant said that he did not know if to speak normally or in a louder tone. The conclusions of the evaluator were as follows: a) The users need a vocal or visual warning about the start and end of the scanner operation. b) The response of repeated scanning in case of a repeated vocal command should be changed. c) The time which permitted users wait should be shortened.

5 Summary

In this paper we present and illustrate the concept of a user-based method for speech interface development. We suggest an evaluation process that foster user collaboration at all stages of development and the definition of automatic tests for evaluation and research of speech interfaces.

We found that combining a user centric approach with the software development method can enhance user evaluation and we illustrate it for the case of speech interface development.

Acknowledgments. This research is supported by the DELOS Network of Excellence on Digital Libraries http://www.delos.info/.

References

1. Brennan, S.E., Hulteen, E.: Interaction and feedback in a spoken language system: A theoretical framework, vol. 8, pp. 143–151. Elsevier, Amsterdam (1995)
2. Byrne, M.D., Wood, S.D, Sukaviriya, P., Foley, J.D, Kieras, D.E.: Automating interface evaluation. In: Proceedings of CHI'94, pp. 232–237. ACM Press, New York (1994)
3. Detmer, W.M., Shiffman, S., Wyatt, J.C., Friedman, C.P., Lane, C.D., Fagan, L.M.: A Continuous-Speech Interface to a Decision-Support System. Journal of the American Medical Informatics Association 2(1), 46–57 (1995)
4. Dix, A., Finlay, J., Abowd, G., Beale, R.: Human-Computer Interaction. Prentice Hall International (1993)
5. Dubinsky, Y., Catarci, T., Kimani, S.: Using Catalogue Browsing for Speech-Based Interface to a Digital Library. In: IASTED Int. Conf. on HCI (in press) (2007)
6. Lewis, C., Reiman, J.: Task centered user interface design: A practical introduction. University of Colorado, Boulder, CO: (1993)
7. Maguire, M.C.: A review of user-interface design guidelines for public information kiosk systems. International Journal of Human-Computer Studies 50, 263–286 (1999)
8. Nielsen, J., Molich, R.: Heuristic evaluation of user interfaces. In: Proceedings of CHI'90 Conference on Human Factors in Computer Systems, pp. 249–256. ACM, New York (1990)
9. Polson, P.G., Lewis, C., Rieman, J., Wharton, C.: Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. International Journal of Man.-Machine Studies 36, 741–773 (1992)
10. Shneiderman, B.: The limits of speech recognition, Communications of the ACM, vol. 43(9), pp. 63–65. ACM Press, New York (2000)
11. Yankelovich, N., Levow, G.A., Marx, M.: Designing SpeechActs: issues in speech user interfaces. In: SIGCHI on Human factors in computing systems, pp. 369–376 Denver, CO (1995)