10

An Empirical Approach for the Evaluation of Voice User Interfaces

Valéria Farinazzo[1], Martins Salvador[1],
André Luiz S. Kawamoto[2] and João Soares de Oliveira Neto[3]
[1]*Mackenzie Presbyterian University – São Paulo*
[2]*Federal University of Technology - Paraná – Campus Campo Mourão,*
[3]*Mackenzie Presbyterian University – São Paulo*
Brazil

1. Introduction

Nowadays, the convergence of devices, electronic computing, and massive media produces huge volumes of information, which demands the need for faster and more efficient interaction between users and information. How to make information access manageable, efficient, and easy becomes the major challenge for Human-Computer Interaction (HCI) researchers. The different types of computing devices, such as PDAs (personal digital assistants), tablet PCs, desktops, game consoles, and the next generation phones, provide many different modalities for information access. This makes it possible to dynamically adapt application user interfaces to the changing context. However, as applications go more and more pervasive, these devices show theirs limited input/output capacity caused by small visual displays, use of hands to operate buttons and the lack of an alphanumeric keyboard and mouse (Gu & Gilbert, 2004).

Voice User Interface (VUI) systems are capable of, besides recognizing the voice of their users, to understand voice commands, and to provide responses to them, usually, in real time. The state-of-the-art in speech technology already allows the development of automatic systems designed to work in real conditions. VUI is perhaps the most critical factor in the success of any automated speech recognition (ASR) system, determining whether the user experience will be satisfying or frustrating, or even whether the customer will remain one. This chapter describes a practical methodology for creating an effective VUI design. The methodology is scientifically based on principles in linguistics, psychology, and language technology (Cohen et al. 2004; San-Segundo et al., 2005).

Given the limited input/output capabilities of mobile devices, speech presents an excellent way to enter and retrieve information either alone or in combination with other modalities. Furthermore, people with disabilities should be provided with a wide range of alternative interaction modalities other than the traditional screen-mouse based desktop computing devices. Whether the disability is temporary or permanent, people with reading difficulty, visual impairment, and/or any difficulty using a keyboard, or mouse can rely on speech as an alternate approach for information access.

Source: User Interfaces, Book edited by: Rita Mátrai,
ISBN 978-953-307-084-1, pp. 270, May 2010, INTECH, Croatia, downloaded from SCIYO.COM

The current knowledge on VUI comes from small contributions of research projects which propose an assessment for the systems developed in these projects, and attempt to generalize and make recommendations for the evaluation of VUIs, such as PARADISE, EAGLES and DISC (Walker et al., 1997; Gibbon & Moore, 1997; Dybkjaer & Bernsen, 2000). It is important to point out that developing VUI applications is very different from developing GUI applications. The differences include visibility, transience, bandwidth asymmetry, temporality and concurrency (Hunt; Walker, 2000). Hence, it is necessary to review the developing process of VUI applications based on an interface approach, aiming to adapt some peculiar characteristics, starting on non-functional requirements.

2. Requirements of VUI

Graphical User Interfaces (GUI) requirements can be, most of the time, also considered for VUI applications, since usability and feedback must be considered for every human-machine interface. However, there are specific requirements for VUI applications. These requirements come from some basic differences that must be pointed out, especially due to the transient attribute of the voice – while graphical interfaces are persistent. Thus, non-functional requirements were classified as: requirements related to the representation of the information, and requirements related to the data input.

2.1 Non-functional requirements related to the representation of the information

Non-functional requirements of VUI applications related to the representation of the information basically indicate the format that the interaction must assume in order to enable the system to deal with user inputs. These requirements are explained next. (Dybkjaer & Bersen, 2001; Salvador et al., 2008).

Consistency, which is considered one of the most important attributes concerning interface usability (Nielsen, 2000). It controls the unexpected behaviour of the system, reducing the user frustration.

Most of the tasks in VUI systems use only the voice for information input and output. However, the voice is not indicated for all types of application, especially when the user must supply security codes (for example, in a bank system). Thus, sometimes it is convenient to integrate the voice with other interaction modes (Appropriate modes of interaction). The Case Study presented in this chapter integrates two interface modes: Voice for both input and output, and a Graphical interface for output.

It is important, in any type of communication that the feedback provided to be suitable. Computer interaction requires a planned feedback (Foley & Van Dan, 1990). A suitable feedback implies that the user can feel that he is in control of the interaction. The user must feel confident that the system really understood his commands and is working for providing answers to the commands.

There are three levels of feedback: hardware level, which indicates whether the user inputs were successful (for voice inputs, it indicates that the system has actually captured what was said); sequence level, which indicates that a input was accepted (in VUI, it indicates that the system understood that input as an action that has to be performed); and functional level, which indicates that the system is working in order to provide an answer (messages like "please, wait a moment" delivered to the user).

The VUI must support all classes of users, being able to identify each one of them and adapt itself to the user, adapting both content and presentation according to the User Model. A few strategies can be used, for instance, providing barge-in and more detailed information to expert users, whereas providing more concise and superficial information, besides sentences at the end of the dialogue to novice users (Komatani et al., 2003).

The VUI must minimize the cognitive effort the user has to do in order to perform the tasks. Mixed initiative dialogues and sentences at the end of the dialogue may be provided to guide the user towards a suitable utilization of the system.

The content of the system outputs must be correct, relevant and informative enough, without providing and overload of information to the user. The way the system expresses itself must be unambiguous and clear, with suitable language and terminology familiar to the user.

According to the user point of view, the quality of the output voice is related to questions of clarity and intelligibility (proper intonation, emotion, rhythm). There are three types of voice output in a system: entire phrases are recorded and played (used when the information is not dynamic); concatenation of recorded phrases or words; or text-to-speech (TTS), ie. The system synthesizes voice in real-time.

2.2 Requirements related to data input

Dybkjaer & Bersen (2001) and Salvador et al. (2008) defined a set of usability evaluation criteria for VUI systems related to the user access. The criteria are explained next.

According the user point of view, an appropriate recognition means that the system rarely misunderstands the user inputs. However, that depends on several environment factors (whether the environment is noisy or not), on user factors (sex, age, accent, voice tone), and on the quality of the sound received by the system.

It is necessary to manage inputs so that the user feels that the speech is natural. If limitations imposed by the task are satisfied, and the system manages to control the input language, users can feel that the dialogue is natural.

In order to support natural interaction, it is necessary to establish a reasonable dialogue initiative between the system and the user. That depends on the level of knowledge the user has about the system. Dialogues directed by the system may work well for tasks that require that the user provide specific parts of the information, especially when users are new to the system. Aiming to satisfy expert users, who are used to manage large amounts of information, the system must adapt itself and accept dialogues directed by the user.

It is important that the dialogue structure defined by the developer is natural to the user, reflecting his expectations, mostly in dialogues directed by the system, where the user is not able to interfere. When unnatural dialogue structures are used, the users usually try to take the dialogue initiative, and the system sometimes is not prepared to answer such attempts.

It is necessary to provide instructions enough to the user, so he can feel that he controls the interaction. Speech is not suitable for providing complex instructions to novice users. On the other hand, it is necessary to consider expert users and all the issues related to satisfy all the levels of expertise, such as turn taking versus barge-in; help facilities and output for unobvious behaviour of the system.

Covering tasks and domain is also a crucial requirement for the natural interaction. Even if the user is not familiar to a VUI system, usually it is preferable to provide detailed information about the services the system can provide.

Also, when users are aware that they are actually talking to a primitive interlocutor, they tend to assume the system is able to perform small pieces of reasoning that human beings do without even thinking about, and which are intrinsically related to the natural dialogue of the task.

The interface must provide a help mechanism whenever it is required or when the user is in a difficult situation. For VUI, a dialogue must provide a list of possible actions the user can take in the system every time the user does not take the initiative of the dialogue. Strategies of dialogue confirmation may also be used.

A good interface is able to prevent user from committing errors. In VUI, the interface can try to guide the user to quickly reach his goals. For instance, the control of the dialogue can be transferred to the system whenever the user is in difficulty, or the system can provide additional sentences in the end of each dialogue, alerting the user about the next steps that can be taken in the system.

A good interface is able to quickly correct inputs, increasing the productivity of the users and stimulating them to explore the system. VUIs can attend this requirement by adopting mixed initiative dialogues, confirmation techniques and, in telephony systems, transferring the call to a human attendant. It is possible to divide the error treatment into four classes:

- Repair the system initiative: necessary when the system is not able to understand or is not sure whether the user input was correctly understood. The system can ask the user to repeat the input, to speak louder, to change the mode the input is being done, or even repeat what was understood and ask the user to correct or confirm the input. If this does not solve the problem, the system can change the interaction to a simpler mode, or even transfer the control to a human operator;
- Repair the user initiative: some systems require the use of specific keywords. This is not natural and sometimes it is hard for the user to remember these keywords. Another possibility is to adopt the eraser principle, where the user simply repeats the inputs until the system accept the message;
- Explication asked by the system: when the user input is inconsistent or ambiguous, the system asks an explication to the user;
- Explication asked by the user: happens when the system produces inconsistent or ambiguous outputs, or when the user is not familiar with the terms used in the communication;

The lack of cooperativity in the system output can be diagnosed from the occurrence of communication problems in real or simulated interactions between user and system. The issue related to capturing and analysing these data is that this activity requires high expenses, especially because a large amount of data is necessary in order to solve most of the communication problems caused in the system. Avoiding such interaction problems more efficiently requires the application of an evaluation methodology already in the project system phase.

A subjective measure of usability derived from personal preferences and contextual factors is the User Satisfaction. This measure can be obtained from quizzes and interviews with users.

2.3 Technical issues

According to Alapetite et al. (2009) and Deng & Huang (2004), when a VUI application is developed, there are a few questions which cannot be underestimated for the application success. Those questions are explained next.

The size of the vocabulary and the domain coverage affects voice recognition. Thus, large vocabularies with good domain coverage are more attractive, due to the fact they are able to recognize more words. However, smaller vocabularies increase the level of correctness in the recognition process. Besides, transcription systems work better when using restricted domains.

Voice recognition is affected by the clarity, consistency and the accent of users. User-dependant systems have a recognition rate higher than systems that do not depend on users. However, user-dependant systems require training sessions – considering that the system adapts its acoustic model to the user – and may be more sensible to noise, microphone and voice variations (for example, if the user has a cold). Besides, non-native speakers in the system language should be trained, as well as recognition rates for children and elder people should be considered.

Noisy environments affect voice recognition in two ways: voice signal distortions imply in higher difficulty to distinguish the spoken words; and when there is noise, the users usually change their voices and, thus, distort the speech signal.

Every VUI system is based on statistical patterns principles. However, despite their similarities, systems differ from each other in the parameterization of their voice signal, acoustic model of each phoneme, and the language model used for choosing the words more appropriately. Thus, systems can generate different error recognition rates, even if their recognition rates are similar.

3. Criteria and guidelines for the evaluation of VUI

Traditional methodologies for evaluating GUI can be used for VUI systems. However, there are substantial differences, since, as mentioned before, the voice is a transient type of information, while the image is persistent. The challenges for evaluating VUI systems are:

- Which interface requirements may be, or may be not considered for VUI;
- What are the general requirements and what are the specific VUI requirements that must be considered;
- Which requirements, among the several discussed are said to be fundamental and, hence, must be considered;
- How to measure each fundamental requirement
- How to evaluate the systems in a viable way, with cost and time acceptable to the application domain;
- Which techniques to use for the evaluation, when evaluate and, moreover, if the final user should be involved.

3.1 How to evaluate voice recognition systems

According to Dybkjaer & Bernsen (2001), in order to evaluate a voice recognition system, it is necessary to adopt templates which contain the following questions:

- What is being evaluated (for example, appropriate feedback);
- Which part of the system is being evaluated, for example, the dialogue management;
- What is the evaluation type , for example, qualitative;
- The evaluation method, for example, user observation;
- Symptoms to be checked, for example, if the system help is consistent;
- The importance of the evaluation, for instance, crucial;
- The level of difficulty of the evaluation, for example, easy;
- The support tools, i.e. the tool used to measure the time a task takes to be accomplished.

The idea is to provide a set of tools enough to the evaluator so that, following this template, the VUI can be evaluated effectively and efficiently. We must also consider that the importance of the criteria for evaluation of a VUI depends on the application and user, or group of users of this system.

4. Case study

4.1 System for improving pronounce skills

The case study was designed for improving the pronounce skills of non-native English speakers. This application works as follows: random words are shown in the screen to the user, who needs to pronounce them as accurately as possible (Figure 1). Each voice input is analyzed by the application, which verifies which level of correctness (recognition) the engine supplies for that input. If the result coming from the engine and the word displayed in the screen do not match, or if the level of recognition is defined as "low", the user is requested to repeat the word. When the number of attempts reaches three, the word is synthesized to the user (so he or she can hear the correct pronounce), and the word is marked as "not recognized".

In the case of the word is correctly spoken and, therefore, recognized, the application randomly picks another word for the pronounce evaluation. When ten words are spoken, whether recognized or not, a report is generated and presented to the user.

The main features of this application are the recognition of words that are spoken by the user, and the text to speech conversion. The application was developed using the Microsoft Speech Recognition Sample Engine for English (Microsoft SAPI, 2009). This engine uses the Hidden-Markov models (Gales, 2008), which are statistical models based on probability for the speech recognition, and the Text-to-Speech Concatenative Syntehisys technique (Braga, 2008).

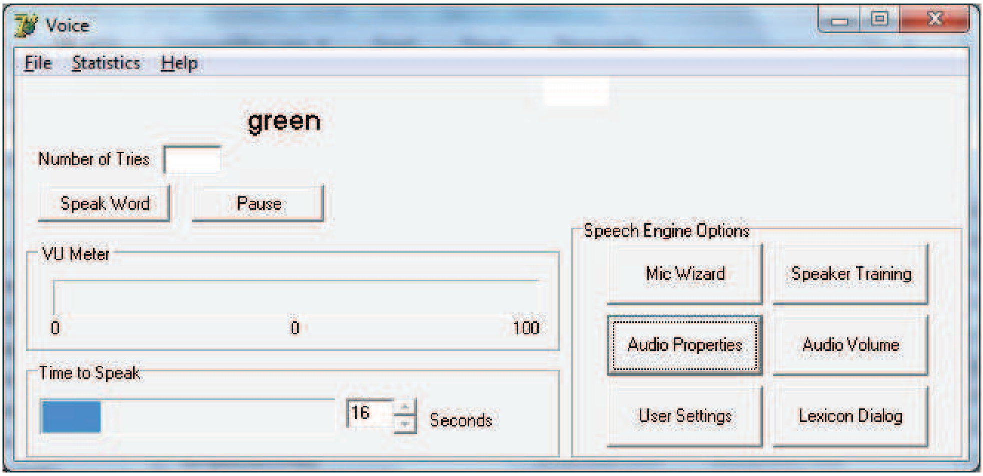

Fig. 1. GUI Interface of the VUI Application

4.2 Implementation issues

The application was implemented using Borland Delphi IDE (Borland Delphi, 2009) and the Microsoft Speech API (SAPI) Version 5.1 (Microsoft SAPI, 2009).

SAPI is middleware that provides an API and a device driver interface (DDI) for speech engines to implement. The speech engines are either speech recognizers or synthesizers. Each speech engine is language specific. The SAPI Architecture is presented in the Figure 2.

A few issues were reported during the implementation. First, it was necessary to stablish a way to suspend and resume the recognition engine. The engine attempts to recognize every

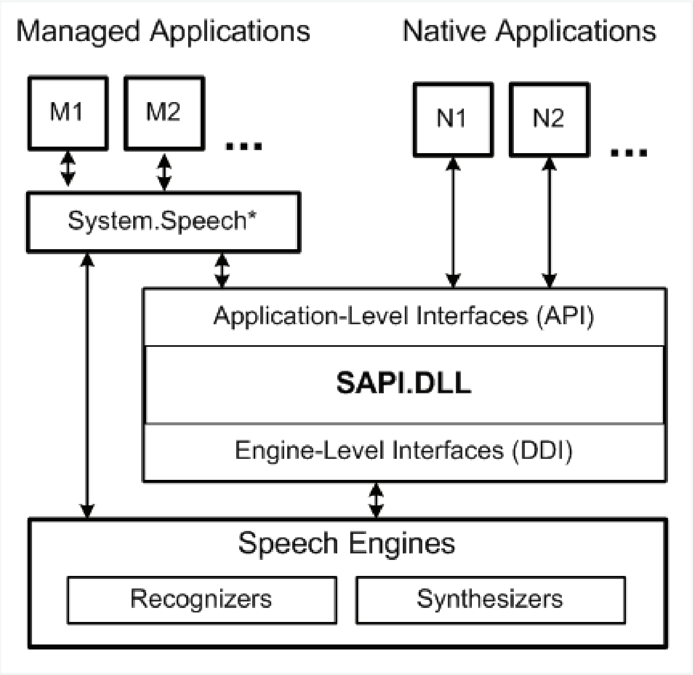

Fig. 2. Speech API Engine (font: http://msdn.microsoft.com/en-us/library/bb756992.aspx)

input that is recorded. This was done by inserting a flag indicating whether the system might accept or not the engine results.

Enabling the application to correctly work on different Operational Systems was another issue, because Speech Recognition is a built-in feature in Microsoft Vista for English Language, but in other OS it must be installed and properly configured. This issue still causes a little concern when the system needs to be installed for a different range of users.

Programming issues were not reported, due to very comprehensive guide available in the Internet for the SAPI, and due to the large number of similar applications available in the Internet. The authors must point out that the system is relatively simple, because it was developed only to support the evaluation of usability proposed.

4.3 Methodology

In order to evaluate the usability of the developed application, we have employed the heuristic evaluation, a type of usability inspection method. We used a checklist based on the heuristics presented in Table 1. These heuristics are based on re-interpretations (Nielsen, 1993), on the study of non-functional requirements for VUIs, and on the good practices of development pointed out by (Dybkjaer & Bersen, 2001), (Salvador et al., 2008) and (Komatani et al., 2003). In order to perform the evaluation satisfactorily, three evaluators were invited to participate. These specialists that participated in this application evaluation are experienced HCI researchers, as well as experts on the VUI applications development process. These specialists are also skilled on heuristic evaluation. They used the checklist presented in section 4.3.1.

For this evaluation task, two scenarios were generated:

- user reads the words, but maybe (s)he know or not the right pronunciation. (S)he is in a quiet environment;
- user is in a noisy environment (probably at work/school), and (s)he, probably, knows or not the right pronunciation of each word.

So, the application evaluation was composed by the following steps:

- *Elaborating the evaluation form that should be fulfilled by specialists*. The design of this form was based on requirements presented in section 2. The final version of the form has three fields: Requirement; Classification (whom eligible values are "Yes", "No", or "Not applicable"); and, Remarks. Fig. 3 shows a template of this form with just one heuristics category – i.e. Appropriate modality. The complete list with all heuristics and their categories are presented in section 4.3.1;

APPROPRIATE MODALITY			
	YES	NO	NOT APPLICABLE
In addiction to using voice, user can use other modalities to interact with the application?			
The use of keyboard or mouse is appropriate to the application?			
REMARKS			

Fig. 3. Form template that should be fulfilled by specialists – Heuristics category "Appropriate modality"

- *Specialists perform evaluation*. Each specialist evaluates the applications verifying whether the principles of our approach were observed, reporting faults and the fault level, concerning the usability principle commitment, found in the application;
- *Results compilation*. An evaluation summary is created based on collected the results collected by specialist.

4.3.1 Heuristics-based usability checklist

The heuristics-based usability checklist built by the authors is listed below:

- Suitable Feedback
    - Does the application provide feedback to every user's action?
    - If the application takes a long processing time, becoming not available, due to user's data input, does the system inform the user about its current status and also for how long the user must wait?
    - Does the system inform the user about successful, or not, word recognition?
- User diversity and user perception
    - In the case of a system designed for wide range of users, does the application provide suitable messages that match the level of each user?
    - Are the dialog styles appropriate to users capabilities, allowing step-by-step actions for novices and more complex inputs to advanced users?
    - Does the application provide shortcuts?
- Minimizing memorization efforts
    - Does the system force the use of key-words?
- Appropriate output sentences

  - Does the system have outputs with adequacy information?
  - Are the system outputs correct?
  - Are the system outputs relevant?
  - Are the systems outputs really instructive?
  - Does the system outputs cause information overload to the user?
  - Is the output terminology well-know and easily recognized by user?
- Output Voice Quality
  - Is the system output clear?
  - Has the system a right intonation?
  - Has the system an appropriate rhythm?
  - Does the system make the user feel good concerning to listening?
- Proper entry recognition
  - Does the system rarely misunderstand the user input?
- Natural user speech
  - Does the system provide an easy (and natural) interaction human-computer by voice?
- Appropriate dialog start out and adequate instruction about how to interact with the application
  - In the point of view of novice users, does the system conduce, in a well-done way, the dialog?
  - In the point of view of advance users, does the system allow a big amount of input data at a once?
- Natural dialog structure
  - Concerning to the dialog, is it natural to user, accomplishing the user’s expectations, specially in the cases when the dialog is conducted by the system, and user is not allowing to interfere on the dialog structure
- Sufficiency of interface guidance
  - Does the user feel himself as the controller of the interaction?
- Help tool
  - Does the application provide a complete and extensive help to aid the users?
  - Are there different help levels suitable to the complexity of the demanded information?
  - Does the system use dialog strategies based on confirmation?
- Error prevention
  - Does the system emit appropriate sounds when input data problems occur?
  - Does the system provide a feedback to the user when the input information has not been understood?
  - Does the system force the use of key-words?
  - When the user input is inconsistent or ambiguous, does the system request more information?
- Handling errors
  - Error messages help to solve the problem, giving precisely the right location, the specific or general reason, as well as the right actions that user should perform to solve the problem
  - Are the error messages neutral and polite?

- Are the error messages short and elaborated with few words and well-known?
- Are the error messages free of abbreviations or specific codes generated by the operational system?
- Are the message contents updated when users produce the same error consecutively?

4.4 Results

Based on the checklist proposed by authors, the usability evaluation was performed by three VUI experts. The main results are listed below:

1. Appropriate modality: three different modalities are employed for user interaction: keyboard, mouse and voice, which seems to be very enriching for VUI systems.;
2. Suitable Feedback: the application does not point out clearly when the recognition task fails, even for the third attempt of recognition. The system just repeats the word with the correct pronunciation;
3. User diversity and user perception: the application, even in its initial prototype, does not consider the variety of user types (beginners, intermediates and experts) that interacts with the system;
4. Appropriate phrases out: although the content of the output is correct and relevant, and the used terminology is appropriate, there is a lack of information that should be provided to the user about the pronunciation approval or disapproval;
5. Output Voice Quality: as the system pronounce just one word-a-time, some features such as intonation, rhythm and pleasure of hearing can not be evaluated;
6. Proper entry recognition: if the user previously does not perform the voice training task, the system hardly will recognize the user's inputs. As the application can be run without this training phase, user should be informed about the consequence of not performing the voice training task;
7. Appropriate dialog start out and adequate instruction about how to interact with the application: the system could present more introductory information for novices about what would happen as result of user's action. On the very first time interacting with the application, user could face some misunderstandings, since the system is starting to count the time waiting that the user pronounces the word that is highlighted on the screen.
8. Help tool: due the evaluated application is in its prototype phase, the system does not provide a complete help system, nor different levels of help;
9. Error prevention: the feedback provided when the system does not understand what the user has pronounced could be better explained. The feedback current can induce user to error;
10. Handling errors: the error messages are free of abbreviations and/or codes generated by the operating system, which often cause confusion for the user. However, they could be clearer, saying how many times the user has tried to pronounce the proposed word. In the third attempt, for example, the system should announce the correct pronunciation and inform that this word would be considered as a bad pronunciation;

Concerning to the two proposed scenarios considered in our evaluation process, when the evaluation was applied for the scenario 2 (noisy environment), the recognition rate decreased (it is less than 20%), then, the system becomes inappropriate to use.

5. Conclusions

This chapter aimed to present the evaluation of VUIs applications. A specific evaluation plan was proposed and used to test the application by three experts. This plan included inspection tests (checklist method), based on heuristic evaluation.

Our premise is that voice recognition applied to language teaching may improve the users' pronunciation. This will be verified when this application be applied for final users. Then, the prototype will be improved and other teaching levels will be included, enabling the application to be able to be used and tested by final users. One issue to be worked on is related to the low recognition rate. It is necessary to investigate why this is happening.

These heuristic rules were adapted to cover the case study. It is important to verify if these rules are sufficient for other case studies.

Future work will involve the development of others VUI tools to improve the user's listening and grammar for foreign students. Besides, a study about improving the recognition level when the application is executed in noisy environments should be delivered.

6. References

Alapetite, A.; Boje, A. H. & Morten, H. (2009). Acceptance of speech recognition by physicians: A survey of expectations, experiences, and social influence, *International journal of human-computer studies*, vol. 67, n. 1, pp. 36-49.

Borland Delphi IDE http://www.borland.com/br/products/delphi/index.html

Braga, D. Algoritmos de Processamento da Linguagem Natural para Sistemas de Conversão Texto Fala em Português. PhD Thesis. University of A Coruña, A Coruña, Spain. (2008)

Cohen M; Giangola J. & Balogh J. (2004). *Voice User Interface Design*. New York: Addison Wesley.

Deng, L. & Huang, X. 2004. Challenges in adopting speech recognition. Commun. ACM 47, 1, pp 69--75, (2004)

Dybkjaer, L. & Bernsen N.O.: Usability Issues in Spoken Language Dialogue Systems. Natural Language Engineering, Special Issue on Best Practice in Spoken Language Dialogue System Engineering, Vol. 6 Parts 3 & 4, 243-272, (2000).

Dybkjaer, L. & Bernsen, N. O. (2001). Usability Evaluation in Spoken Language Dialogue Systems, *Proceedings of the ACL 2001 Workshop on Evaluation Methodologies fro Language and Dialogue Systems*.

Foley, J. D. & Van Dan, A. (1990). *Computer Graphics: Principle and Practice*. Reading: Addison Wesley.

Gales, M. The Application of Hidden Markov Models in Speech Recognition, ISBN 9781601981202, Now Pub, 2008.

Gibbon, D. & Moore, R., WINSKI, R. (Eds.): Handbook of Standards and Resources for Spoken Language Systems. Mouton de Gruyter, Berlin, New York, (1997)

Gu, Y. & Gilbert, J. E. (2004). The development of a Voice System that interacts with a student information database. In *Proceedings of the 42nd Annual Southeast Regional Conference* (Huntsville, Alabama, April 02 - 03, 2004). ACM-SE 42. ACM, New York, NY, 248-252.

Hunt, A. & Walker, W.: A fine Grained Component Architecture for Speech Application Development, SUN Research, Project: SMLI TR-. 2000-86, June (2000).

Komatani, K.; Ueno S.; Kawahara, T. & Okuno, H. G. (2003). Flexible Guidance Generation using User Model in Spoken Dialogue Systems, *Proceedings of the 4lst Annual Meeting of the Association for Computational Linguistics*, pp. 256-263.

Microsoft Speech API (SAPI) Version 5.1. http://msdn.microsoft.com/en-us/speech /dd380587.aspx

Nielsen, J. (1993). *Usability Enginnering.* Academic Press, Cambridge, MA.

Nielsen, J. (2000). *Designing web usability*. Indianapolis: News Riders Publishing.

Rubin, J. (1994) *Handbook of Usability Testing - How to plan, design and conduct effective tests.* New York: Wiley.

Salvador, V. F. M.; Oliveira Neto, J. S. & Kawamoto, A. L. (2008). Requirement Engineering Contributions to Voice User Interface. *Proceedings of the First International Conference on Advances in Computer-Human Interaction, 2008, Sainte Luce. First International Conference on Advances in Computer-Human Interaction*, p. 309-314.

San-Segundo, R.; Montero, J. M.; Macías-Guarasa, J.; Ferreiros, J. & Pardo, J. M.: Knowledge-Combining Methodology for Dialogue Design in Spoken Language Systems, International Journal of Speech Technology 8, 45-66, Springer Science + Business Media (2005)

Walker, M.; Litman, D.; Kamm, C. & Abella, A.: PARADISE: A Framework for Evaluating Spoken Dialogue Agents. Proc. of the Association of Computational Linguistics (ACL), pp 271—280, (1997)

ISBN 978-953-307-084-1
Hard cover, 270 pages
Publisher InTech
Published online 01, May, 2010
Published in print edition May, 2010

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821