

# Software Tutors for Dialogue Systems

Jaakko Hakulinen, Markku Turunen, and Esa-Pekka Salonen

Department of Computer Sciences  
University of Tampere  
Finland

{jaakko.hakulinen, markku.turunen, esa-pekka.salonen}@cs.uta.fi

**Abstract.** We have used text, graphics and non-speech audio to tutor new users in a spoken dialogue system. The guidance is given by a software tutor, a software component that interactively tutors the user. Four different variations of tutoring were implemented and experiences were collected from user tests in order to gain insights into these tutoring concepts. Real-time visualization of speech interaction with comic book style balloons and structured guidance were received well while various other methods received mixed acceptance.

## 1 Introduction

While speech is a natural way of human to human interaction, speech interfaces may become inherently easy to use only when computers understand speech like a human being. Such ultimate speech systems are not possible with today's technology and therefore guidance is needed [3]. Currently speech interfaces lack the fluency and unlimited language of human interaction. Only when users know what each system understands speech applications can be successful services.

Providing on-line help is challenging in speech interfaces. Embedded assistance on the other hand is common and is realized as hint type guidance, expanding system prompts and system initiated dialogue [9]. Many speech application use traditional manuals as well, commonly in the form of web pages.

One option for initial user guidance is a software tutor. A tutor provides guidance within the actual software application, monitors user actions and is capable of providing help in appropriate context and according to users' needs [1]. There are some studies of tutorials in the context of speech interfaces. Kamm et al. [4] studied a tutorial that was not connected to an application but showed in a web page how an example task can be carried out. Consistently higher user satisfaction ratings were found in the tutored group compared to embedded assistance only. We have found an actual software tutor in speech interfaces to be effective as well. Tutoring was delivered in between system prompts and enabled users to learn a speech interface with significantly fewer problems than with a web manual [2].

Incorporating text and graphics has potential benefits of tutoring in a speech interface. Separate tutoring modality widens communication channel and tutoring can happen simultaneously and in synchrony with the system actions [6].

Graphics can also stay on screen as long as needed. Graphical presentations can be very powerful, for example, Terken and te Riele [8] conclude that in their study the graphical part of a multimodal interface gave users a mental model of the interface. While graphics are not usually available when speech interface is used, the initial learning often happens in a situation where this is not the case, for example, web pages and Java applets can be available.

In order to effectively guide users to speech interfaces one must understand what the users need to learn. We have identified the following topics: 1) Interaction style; when and how to speak and turn taking. 2) The error prone nature of speech recognition. 3) System functionality. 4) Language models i.e. what kind of inputs the system understands.

In particular, tutoring must consider recognition errors. If a tutor can detect an error, it should provide guidance that helps users identify the error situation and correct it. One option is to visualize the recognition results and thus help the users to detect errors. A tutor can also give explicit instructions for the user to say something and monitor speech recognition results to make sure that the user can give inputs successfully.

In this paper we introduce a set of graphical software tutors that provide guidance to the users of a speech-based timetable system. The tutors use graphics and audio notifications. They are connected to the telephone-based timetable system to visualize the system and monitor users' actions. The next chapter describes the tutors and our experiences with them. The experiences and findings are discussed in the end.

## 2 Four multimodal tutoring implementations

The tutors teach how to use Bussimies, a telephone service for bus timetables. Bussimies has grammars of around 1500 words with word spotting. Users can express themselves reasonably freely, but they need to know what questions Bussimies can answer, and what concepts, like bus lines and destinations, it knows. Interaction style is mixed-initiative; users can ask questions freely, but when errors occur the system can take initiative.

The four tutor versions share the same four part structure: an initial instruction set, a guided hands-on exercise, a second set of instructions and a free experimentation. Initial instructions introduce Bussimies, explain speech recognition and show some example inputs. During the hands-on exercise a user gives a call to Bussimies and gives it some inputs by following tutor's instructions. The second set of instructions contains more guidance on valid inputs and a summary. Free experimentation with Bussimies is possible in the end.

During the hands-on exercise and free experimentation the speech recognition results and systems outputs are visualized. Context sensitive help is given in error situations. Users control the tutor with *Continue* and *Back* buttons.

During the iterative development of the tutors 19 people participated in user tests. All users tested each of the four versions. Observation, questionnaires and discussions were used to collect opinions and findings to guide the development.

The differences between the four tutors are mostly in the visual representations of the Bussimies system. Next, the four variations are discussed with findings from the user tests.

## 2.1 Balloon tutor

The balloon tutor, as seen in Figure 1, visualizes the spoken interaction with balloons similar to those in comic books. New balloons scroll into screen from right and when more space is needed, the leftmost balloon is removed. The user and system balloons have slightly different shapes. The primary motivation for using balloons is to show the users the results of speech recognition and thus help them to notice speech recognition errors. The visualization of system output should help the users with speech synthesis. Comprehension of speech synthesis often improves after listening to it for a few minutes. During this period, seeing the same information as text can help. Balloons also leave a short term dialogue history visible on screen.

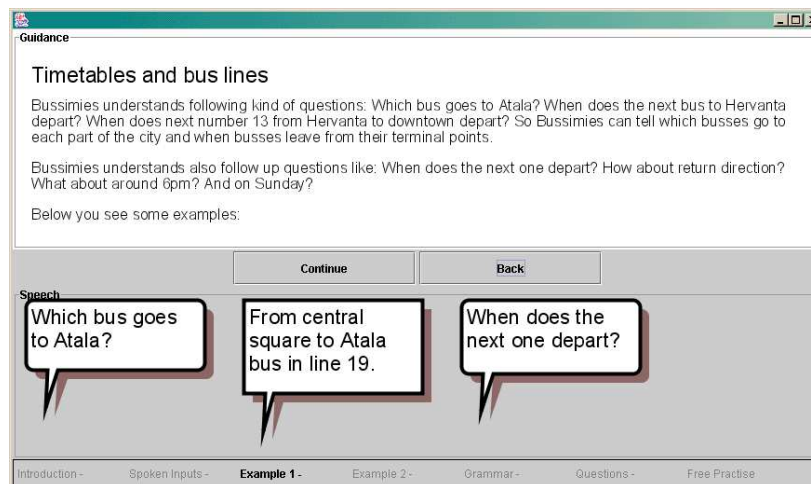


Fig. 1. Balloon tutor

Balloons are an efficient visualization of a dialogue since people are familiar with the concept from comics and associate balloons with speech. Dialogue turns are also naturally visualized.

In the user tests, the balloon tutor was received very well. Mostly people liked the tutor because of its simplicity; the balloons were considered easy to understand and follow. The idea was familiar and the movement from right to left provided the feeling of an advancing dialog. The timing of the balloons that appeared slightly before Bussimies started speaking also seemed effective. The visualization matched the flow of the spoken dialogue very well.

The users were able to follow the performance of speech recognition using the balloons. In error situations they knew what had happened and some of those who had problems with recognition, reported that they used this information to adjust their way of speaking to get better results. The fact that they could see how Bussimies had recognized their speech was often reported to be the best feature of the tutoring.

The only negative comments on balloons concerned the animation; some participants found the movement a bit confusing.

## 2.2 Form tutor

The form tutor, as seen in Figure 2, visualizes a form that Bussimies uses in dialogue management. In addition, another form shows users' input as form items. Speech recognition results and system outputs are visualized as separate balloons. The idea is that revealing the systems internal representation of the queries provides the users with an accurate mental model of the system. This model should tell what kind of questions the system understands.

The screenshot shows a window titled "Guidance" with the following content:

**Timetables and bus lines**

Bussimies understands following kind of questions: Which bus goes to Atala? When does the next bus to Hervanta depart? When does next number 13 from Hervanta to downtown depart? So Bussimies can tell which busses go to each part of the city and when busses leave from their terminal points.

Bussimies understands also follow up questions like: When does the next one depart? How about return direction? What about around 6pm? And on Sunday?

Below you see a description of Bussimies functioning. On left the user input is viewed the way Bussimies heard it, next is the same input in a form style representation that Bussimies uses. Third column contains the internal form of Bussimies. Bussimies fills the form according to your speech, and looking at timetables and deducting e.g. from prior discussion. In the rightmost column you see speech of Bussimies. You can view this visualization when we call Bussimies next.

Buttons: **Continue** and **Back**

User Input	Continue	Back	System Output
Which bus goes to Atala?	question type <b>which line</b>  destination <b>Atala</b>	question type <b>which line</b>  destination <b>Atala</b>  departure location <b>Central Square</b>  bus line <b>18</b>  bus line <b>19</b>  bus line <b>27</b>	From central square to Atala bus in line 18, bus in line 19 and bus in line 27.

Navigation: Introduction - Speech Recognit. - **Comprehension** - Practise 1 - Practise 2 - More info - Dictionary - Question types - Free practise

Fig. 2. Form tutor

The form tutor was favored by some participants because it provided a way to see how Bussimies understands the user inputs. However, many of the participants found the forms useless and irrelevant. They were also considered complicated and technical. One participant wrote: "[The form] Clarifies the understanding [...], but on the other hand requires a terrible amount of concentration."

### 2.3 Interactive GUI tutor

The interactive GUI tutor has a graphical user interface (GUI) that users can construct queries with. As seen in Figure 3, the resulting natural language queries are shown to the users in balloons. The tutor formulates the queries into a phrasing that Bussimies accepts. The balloons also provide the visualization of dialogue. One of the strengths of GUI is that the possibilities and limitations of the system can be seen from the interface. In the case of Bussimies, the users can see from the GUI what kind of queries can be made with speech.

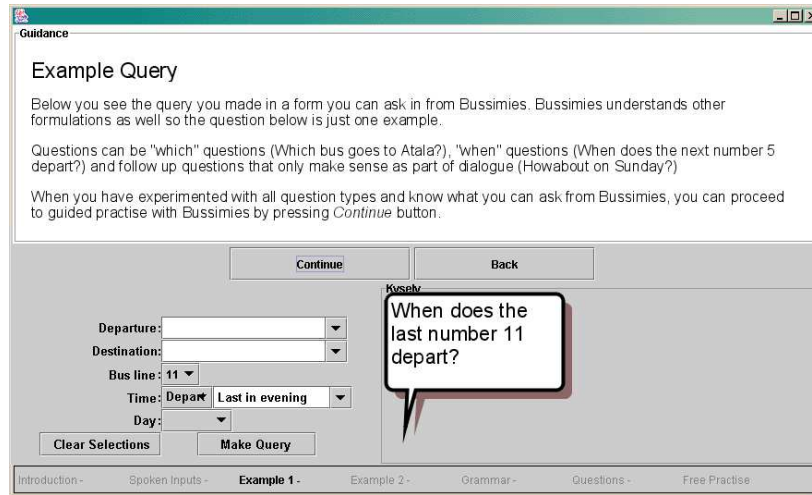


Fig. 3. GUI tutor

The participants liked the interactive GUI tutor because it allowed them to do things themselves. The interactive GUI tutor also best communicated the features of Bussimies to some participants. Many commented that after using the GUI they found Bussimies to have more features than they previously thought.

However, not all participants found the GUI useful. Partly this could be because the users tended to fill in the whole GUI form once and proceed. Therefore they did only see one type of question. Only a small proportion of the participants experimented with the GUI to find out the possibilities and features of Bussimies. The lack of interest in the GUI may be because the GUI was not available when the users freely experimented with Bussimies. The design of the GUI did not clearly suggest all the possibilities either.

### 2.4 Animated tutor

The animated tutor, as seen in Figure 4, has two animated features: on the left side the system components of Bussimies are visualized and on the right a

human like character does the tutoring. There are visualization icons for speech input, speech output, database and a form to visualize dialogue management. The icons are animated when the corresponding system component is active. The visualization was supposed to show the users what the system is doing at any given time.

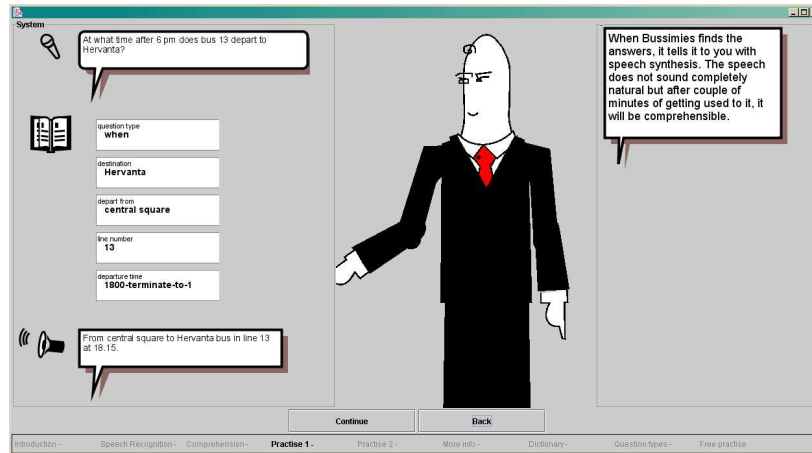


Fig. 4. Animated tutor

The motivation behind using a human like character is "persona effect"; users find interface with an anthropomorphic character more helpful, credible and entertaining [5]. The character in the tutor was animated so that it pointed and looked at the icons. Additionally, the character randomly rotated its head and torso a little bit all the time. These idle-time acts [7] are supposed to tell the users that the character is "alive" and the system is responsive.

The participants who preferred the animated tutor referred to the character in a way compatible with the persona effect, e.g.: "[It] felt like there was somebody helping." However, the character received negative comments from the majority of the participants; it was considered foolish and annoying. Opinions on the visual design seemed important to the acceptance of this tutor. A feature like this should be optional and easy to get rid of if one finds it useless or annoying.

The visualizations icons were in general considered helpful. However, the initial versions that consisted of abstract visualizations of speech recognition and synthesis were considered unnecessarily complex and even scary. Simplified versions seen in figure 4 were accepted by the participants. The complex layout of the animated tutor was also found problematic as there was information on both sides of the screen. The users did not always know where to look at.

### 3 Discussion

The concept of graphical tutoring was received well in the user tests. Most users reported that they learned to use Bussimies with the tutors. Each participant selected a favorite tutor; balloon tutor was selected 5 times, form tutor 6, the interactive GUI tutor 6 and the animated tutor 2 times.

The interaction style tends to be the most challenging topic to teach using traditional manuals. The tutors taught this to the participants successfully. All users learnt when and how to speak. If they had problems during the tutoring, the tutors detected it and appropriate guidance was given. Additionally, the balloons enabled the users to notice problems themselves.

The real-time visualization of interaction was valued by the participants and it was often selected as the best feature of the tutoring. The participants understood well the error-prone nature of speech recognition and words like "mishear" were used to describe this. The visualization of recognition results gave the users a possibility to learn what kind of mistakes the system makes, how it behaves when errors happen, and how one should speak to avoid errors.

To some participants the guidance failed to explain the rather open nature of grammars and the word spotting. They thought that the given examples were the only valid phrasings. The form and GUI tutors that were supposed to best provide this information had the problems discussed in the previous chapters that turned many users away from them. In the discussions after the tests the term keyword was often used by the participants to refer to the parts of the input that Bussimies uses. It was suggested that this point of view could be used in the tutors. The tutoring texts have now been modified to use the keyword concept and to better convey the nature of open grammars.

After the tests the keyword concept was further developed: the keywords in the speech recognition result balloons are bolded. The enhanced visualization provides much of the same information as the form tutor but does not look technical, add the complexity or break the timely correspondence between the spoken interaction and the visualization.

One feature of the multimodal interaction with the tutor is focus shift between the speech interface and the graphical tutor. Our initial design, a notification sound from the tutor, worked well. Every time users heard the sound from computer, they knew that there was something to look at on screen.

A theory that rose from the experience is the required correspondence between the spoken interaction and the visualization in the tutor. The moving balloons as well as the animated icons successfully match the dialogue flow. The form did not have much correspondence to the dialogue flow and the GUI breaks down the speech interaction even more.

### 4 Conclusions

We have built four versions of graphical tutoring for a speech interface. From users tests we gained insight on this tutoring concept. In general the multimodal

tutoring worked well. Several improvements for the current tutoring concepts were found but the foundation is firm.

The big challenge with tutoring in speech interfaces is speech recognition errors. The users must not get stuck in the tutoring process due to errors and appropriate guidance must be given to sort out the problems. The visualization of recognition results and context sensitive guidance help users to understand what is going on and to effectively adjust to the speech interface when necessary.

The best feature in the tutors was the real-time visualization of the dialogue with balloons. It helped users to notice speech recognition errors and provided an effective view on how the system sees the dialogue. The timely correspondence between the visualization and the dialogue was received favorably.

Already in the tested form, the multimodal software tutoring of speech applications was found a viable concept. It can teach novice users the interaction style and functionality of a speech interface in matter of minutes.

## References

1. García, F.: CACTUS: Automated Tutorial Course Generation for Software Applications. Proceedings of Intelligent User Interfaces 2000 (IUI'2000). (2000) 113–120
2. Hakulinen, J., Turunen, M., Salonen, E-P. and Rähkä, K-J.: Tutor Design for Speech-Based Interfaces. Proceedings of DIS2004. (2004) 155–164
3. Heisterkamp, P.: "Do not attempt to light with match!": Some thoughts on progress and research goals in Spoken Dialog Systems. Proceedings of Eurospeech 2003. (2003) 2897–2900
4. Kamm, C., Litman, D. and Walker, M.A.: From Novice to Expert: The Effect of Tutorials on User Expertise with Spoken Dialogue Systems. Proceedings of the International Conference on Spoken Language Processing, (ICSLP98). (1998) 1211-1214
5. Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A. and Bhogal, R. S.: The Persona Effect: Affective Impact of Animated Pedagogical Agents. Proceedings of CHI 97. (1997) 359–366
6. Nakatani, L.H., Egan, D.E., Ruedisueli, L.W., Hawley, P.M. and Lewart, D.K.: TNT: A Talking Tutor 'N' Trainer for Teaching the Use of Interactive Computer Systems. Proceedings of CHI'86 (1986) 29–34
7. Rist, T., André, E. and Müller, J.: Adding Animated Presentation Agents to the Interface. Proceedings of the 2nd international conference on Intelligent User Interfaces. (1997) 79–86
8. Terken, J. and te Riele, S.: Supporting the Construction of a User Model in Speech-only Interfaces by Adding Multimodality. Proceedings Eurospeech 2001 Scandinavia (2001) 2177–2180
9. Yankelevich, N.: How Do Users Know What to Say? Interactions, 3, 6 (1996) 32–43