University of Siena – Faculty of Engineering

Blue Sign Translator

A breakthrough in Personal Communication

The Signs in the picture
are the translation
of the words
"BLUE SIGN"

Team members:
Bennati Paolo
Capasso Tommaso
Giallombardo Francesco
Maggio Emilio

Project mentor:
Giorgi Roberto

Abstract

Blue Sign Translator is an aid instrument for the deaf. It uses the Bluetooth technology to 1) transmit standard messages or messages resulting from human voice and 2) to receive messages and playback them in Sign Language (as a movie) or in another preferred language.

In the whole world, more than seven millions people have hearing problems. Many of them are completely deaf. In the U.S.A. for instance, deaf people are more than 500,000. Most of them, especially those who could not learn a spoken language, will never be able to master perfectly the communication with others. Those people prefer to use the Sign Language for communicating. Sign Language is also an important part of their social integration. The importance of Sign Language interpreters is widely recognized in situations like meetings, seminaries, workshops, but also at school (in particular secondary school and university) and in those common situations where it is necessary to communicate to people at public offices.

Blue Sign Translator – Our project aims to be an answer to Sign Language demand from deaf people, especially in those situations where the human interpreter help is not available. The system is based on the exchange of information in text format (that we called ***"Blue Sign messages"*** format) in order to achieve the maximum in flexibility, both to send and receive information. Our basic system architecture relies on only two Bluetooth devices (a transmitter and a receiver) and could be integrated with existing systems or other Blue Sign systems. We have found many applications that could take advantage from ***Blue Sign*** architecture***. Blue Sign*** messages can come from speech recognition system. They could be embedded in a PDA or in a common cellular phone. On the other side the message can be read as a video in Sign Language (understandable by deaf people), but also as text, eventually translated in a different language, thus facilitating the communication among people who speak different languages.

1. System Overview

Blue Sign Translator is an aid instrument for the deaf. It uses the Bluetooth technology to 1) transmit standard messages or messages resulting from human voice and 2) to receive messages and playback them in Sign Language (as a movie) or in another preferred language. Thus, ***Blue Sign Translator*** can also be used as **an immediate translator to different languages,** becoming a real help for people that are in a foreign country. ***Blue Sign Translator*** operates in real-time.

We imagined the following scenarios for the use of ***Blue Sign Translator:*** 1) the case of a deaf who reads messages in a natural way by using this small and simple device; 2) a single source and several receivers, like in the case of a lesson to a class with hearing people, deaf people, and eventually foreign people; 3) the case of several sources and a single receiver, which selects the interesting messages; 4) the case of a warning message that causes also a receiver vibration.

1.1 Overall system description

Our system is made of two parts, the ***Blue Sign Server*** (**BSS**), which acts mainly as transmitter, and the ***Blue Sign Client*** (**BSC**), which acts mainly as a receiver. The two parts implement the ***Blue Sign Service***, built as an application protocol based on Bluetooth technology.

1.1.1. Blue Sign Server (BSS)

Blue Sign Server translates the information coming from the source and transmits the messages (in ***Blue Sign format***) to a ***Blue Sign Client***. It can work also as a receiver device, if an exchange of information between the two units becomes necessary.

Different types of information sources can exist. We can transmit human voice, captured by a PC or by a cellular telephone (Figure 1). Messages can also be directly typed on a keyboard of a PC or a PDA, or they can come from a custom device with ‘hardwired’ standard messages (e.g. like in the case of an advertisement message or a warning message).

Figure 1: A typical use of Blue Sign Translator.

A, B, C are three subsequent screenshots of the video which is played upon receiving a Blue Sign message. In the picture the Blue Sign message is produced through speech recognition in the cellular phone.

In the case of human voice source, the main reason why it would be ideal to have the information source in a cellular telephone is due to the objective difficulty in recognizing speech from a generic speaker. *The cellular phone can easily learn the voice of its owner, since it is usually a very personal device. In this way, speech recognition would be enormously facilitated. Then, the Bluetooth technology is the missing link to connect the source with the receiver.*

1.1.2.Blue Sign Client (BSC)

The ***Blue Sign Client*** visualises in real time the received information. It can display it in two ways (Video or Text), depending on the service that we would like to have. It can also work as a transmitting device. The client could be implemented either as Laptop, PDA, or videophone.

If users are not deaf, it is also possible to listen at the translation into a certain language by using a cellular phone or to visualise the information on a display as a text string. *Help for translation can be useful for* ***all the people****, not only for the deaf, because* ***a hearing person in a foreign nation can feel like a deaf****.*

1.2 Performance requirements

Our implementation of the system is completely software based. The performance we have obtained results already satisfactory on a 200 MHz processor/32MB RAM notebook.

The heaviest elaboration for BSS regards voice recognition. For BSC, heaviest elaboration is the playback of the video in Sign Language. The receiver would be simplified if the video option were not requested, as in the case of a (simple) translator.

1.3 Design methodology

We started with a preliminary analysis of the problem. We immediately subdivided the design into five (smaller) problems and we tried to implement a software module to solve each problem:

- Speech Recognition
- Bluetooth Link Management
- Translation Engine Development
- Database Management
- Video playback and User Interface Development

We carefully described what each module had to accomplish and how it should interface with the other modules. In this way we could develop and test each part in parallel. **Microsoft .NET** environment was very helpful for the development of our project.

1.4 The Blue Sign Translator contribution

The most important innovation of the Blue Sign Translator is that it allows deaf people to make use of a service that is unique: the availability of a language interpreter always at his hand.

Moreover, Blue Sign Translator is useful to people visiting a foreign country, who have necessity to communicate in a foreign language. Blue Sign Translator allows them to make their own language intelligible to local population, or to read generic messages in their own language.

2. Implementation and Engineering considerations

2.1 Functional Design

The **Blue Sign Translator** structure is based on a logical and physical subdivision of the project into two parts: the ***Blue Sign Server*** (**BSS**), which acts mainly as transmitter, and the ***Blue Sign Client*** (**BSC**), which acts mainly as a receiver. The two parts implement the ***Blue Sign Service***, built as an application protocol based on Bluetooth technology.

2.1.1 Blue Sign Server (BSS)

The server software is organised as in the following logical scheme (Figure 2).

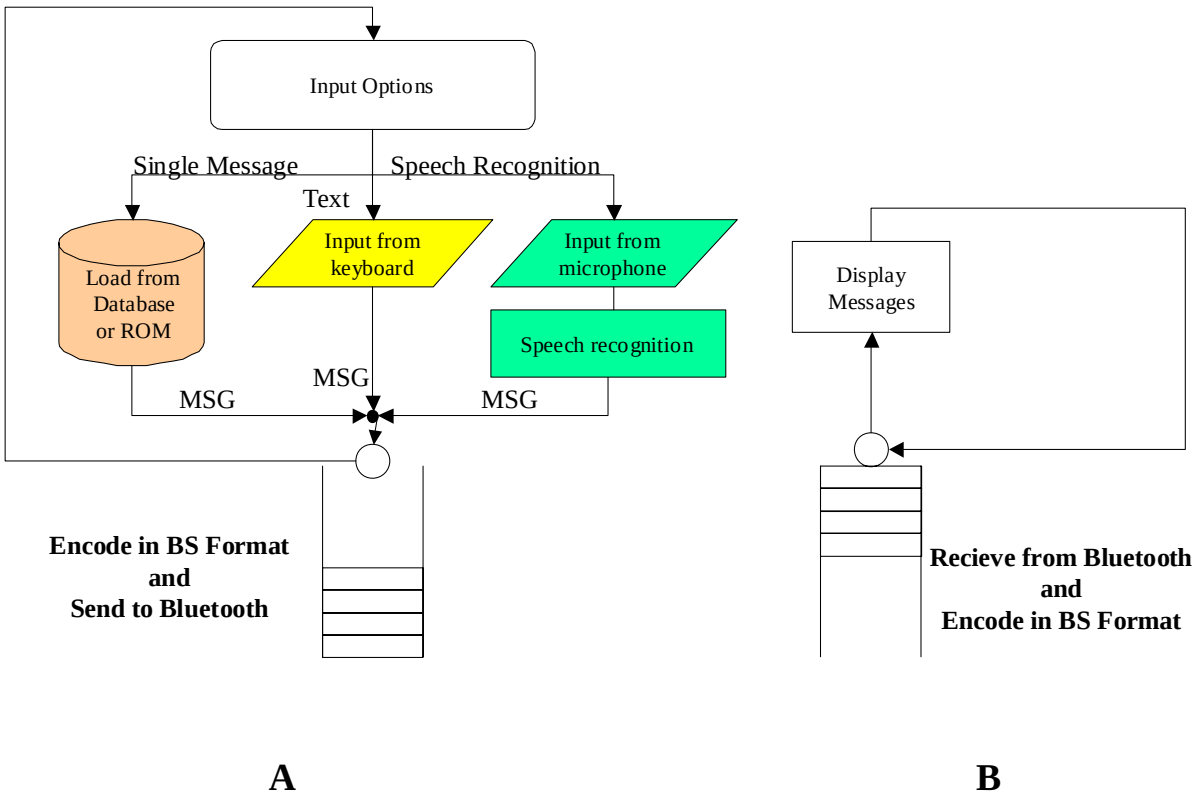

Figure 2: Logical scheme for Blue Sign Server (BSS)
The transmitter portion (A) is the main functionality of the server (BSS). BSS can also receive messages from clients (B).

After starting the server application (BSS), the first step is to setup the Bluetooth card. We can choose the speed of the communication port. Then the BSS initialises the driver (Bluetooth

Stack) that interfaces with the Bluetooth card. All the initialization process is transparent to the user in order to facilitate his operation. At the end of this phase, the server window shows all the functions that were configured (Figure 5).

The BSS allows us to choose default behaviour for a hypothetical incoming dialog request. If the server is enabled to answer to incoming dialog request, then it will automatically open a bi-directional communication session. Otherwise, it will send a message indicating that such option is not enabled on that server. The server itself can also start this kind of session. This could be useful in situations when an urgent message has to be sent to the client, expecting an urgent answer from the client.

Blue Sign Messages can come from three possible sources:

- as a result of speech recognition – in such case the voice is captured by a microphone;
- as a message typed on a keyboard;
- as a predefined message stored in a Data-Base or a ROM.

In the case of predefined message, the server can be setup to periodically send the messages. Database messages can be modified if there are changing needs (e.g. for advertisements).

> *A special use of the system has been considered in situations of danger. The BSS can transmit high priority messages indicating a warning situation*

2.1.2 Blue Sign Client (BSC)

The server software is organised as in the following logical scheme (Figure 3). Once the communication port is setup, the Bluetooth link is managed in a transparent way to the user. The user can configure what is his preferred way to communicate (Video mode or Text Mode). The user can decide the Blue-Sign Piconet (the Blue-Sign Personal Area Network) to be joined.

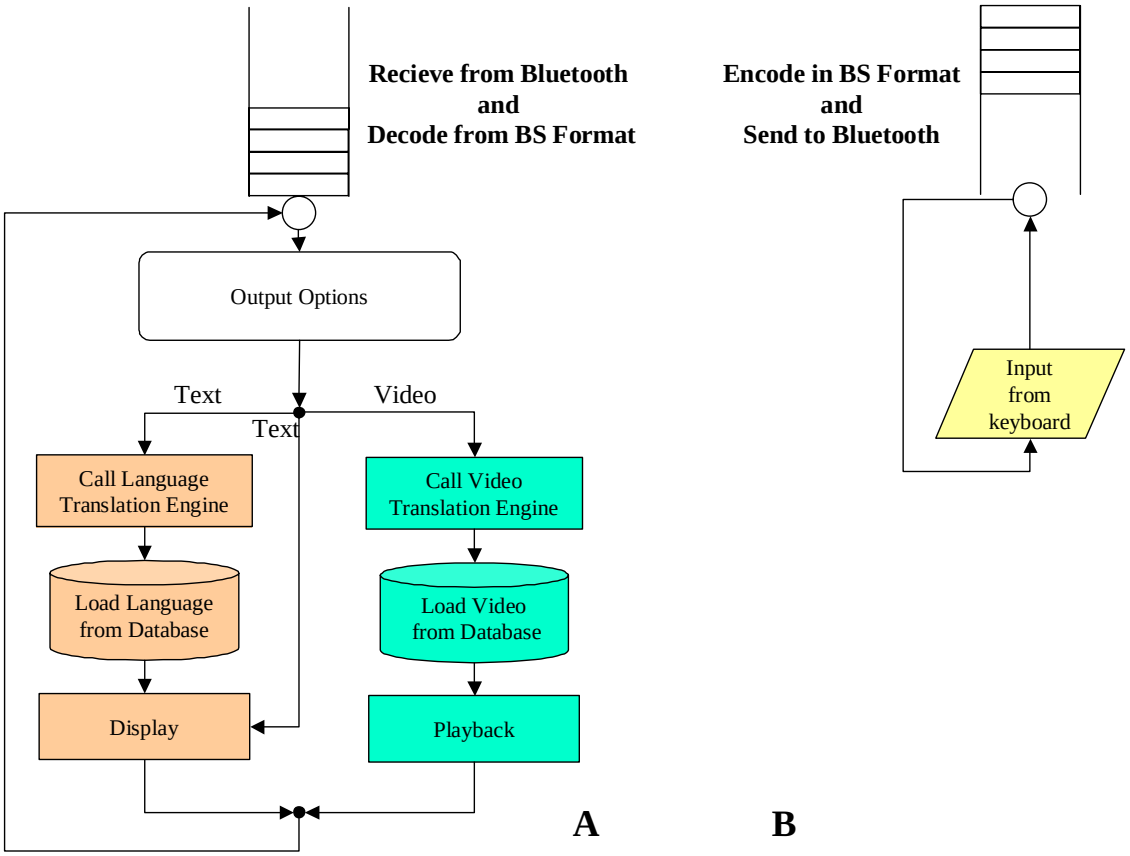

Figure 3: Logical scheme for BSC.
The receiver portion (A) is the main functionality of the client (BSC). BSC can also send messages to a selected server (B).

Video Mode: this is the innovative service for deaf people. The BSC receives and elaborates the Blue Sign messages by translating them into Sign Language and by reproducing the video of the corresponding gestures.

Thus, the deaf person will be able to watch the message in his preferred language: the Sign Language. Movies are loaded from the Video Database (VDB), which contains an archive of

video clips for mostly for words and phrases. If the transmitter portion is enabled, the BSC can also send messages, for example to notify the BSS, about special situations.

Text Mode: in this case the message is displayed as text, without any particular elaboration, if the server and client languages use the same language. If the language is different, then it is used again a translation engine related to the mother tongue of the owner. Again, the client can issue requests for dialogue to the server, which could be enabled to accept them or not. In the negative case, the receiving device will show the “server unavailable” message on the display.

The dialogue occurs in a traditional way in which messages to be transmitted are edited by keyboard and the received ones are shown in the form of strings.

> *If the receiving device detects a high priority message, then it activates a special modality to generate a special alarm like a vibration of the receiver.*

2.2 Blue Sign Service Implementation

Our first step in the implementation of the system has been the definition of an application protocol to exchange what we called a "**Blue Sign Message**". Blue Sign Messages are based on the "**Blue Sign Packet**" (Figure 4). One issue of our project was then how to manage the Bluetooth Communication.

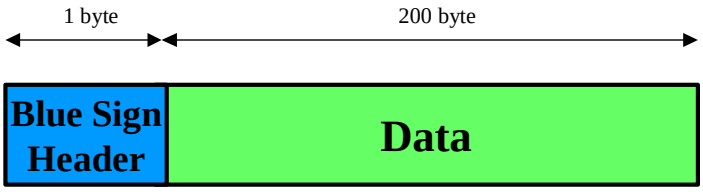

Figure 4: Blue Sign Packet

Since we were using Visual Basic for the User Interface, we used Visual Basic also for the Bluetooth link management. We found a component offered by CSTACK [1], [8], importable in Visual Basic, to interface with the Bluetooth cards. In particular, we used the January 2002 version, which turned out to be compatible both with Ericsson card and with Xircom PCMCIA, provided in the CSIDC Toolkit.

We imported the communication component in our application, exploiting set up procedures, channels, and Link management, without needing to redefine them from the beginning: this simplified and sped up our work.

Regarding the Blue Sign Packet, we added to data only a one-byte code (header). An explanation of the header codes is reported in Table 1.

Table 1. Header codes of the Blue Sign Packets.

Header Codes	
Data Code	Indicates a normal data exchange between Server and Client. It is the default code.
Dialog Request Code	Indicates an incoming Dialog request from Client or Server.
Alarm Code	Indicates a special situation.

2.3 Development kit

The choice of the development environment for the software programming was particularly delicate: all the modules required different programming techniques in order to work in the optimal way. In order to avoid sacrificing the performance of the system, after several attempts we have chosen Visual Basic and Visual C++ [2], [4], [5], [6], [7], [9], [10], [11], [12], [13], [14], [15].

We used the latest version of Visual Studio.NET. This allowed us of integrating the source code of the two languages very easily. We have been able to enjoy the advantages of C++ programming to update the database and to manage the tree searching structures, while we enjoyed the easiness of Visual Basic to design the User Interface, and to manage the Bluetooth Stack Functions, which drive the Bluetooth cards.

2.4 Blue Sign Server (BSS) development

The Blue Sign Server (BSS) is the software that manages the operations on the server side. It has been developed in Visual Basic. In our prototype, the user is presented a main dialog window, which contains the three main functionalities of the server to send a message coming from: i) a Speaker (*Speech Recognition Mode*); ii) a Keyboard (*Dialog Mode*) ii) a Database or ROM (*Single Message Mode*) (Figure 5).

The Single Message option calls a module written in Visual C++, which manage the Message Database. In the "*Bluetooth Device List*" and "*Event List*" the user can verify that the connection management proceeds as desired.

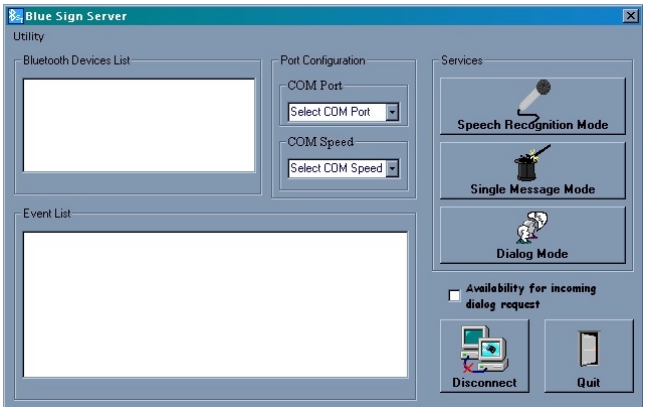

Figure 5: Blue Sign Server Main Window

2.4.1 Initial Setup and configuration

Initially the user of BSS can choose the communication port and its speed (*Port Configuration*). By selecting the check box (deactivated by default) "*Availability for incoming dialog request*" in the main menu (Figure 5 and in detail Figure 6), the BSS can become available to satisfy possible dialog requests coming from a BSC.

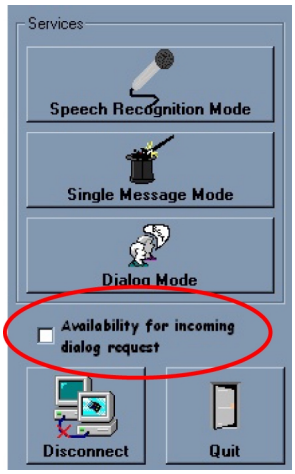

Figure 6: Availability for incoming dialog request check box

2.4.2 Speech Recognition Mode

In this modality, the speech recognition function is used to transmit messages in Blue Sign format to remote users. In our prototype, to activate/deactivate the acquisition, we used the two buttons "*Speech On*" and "*Speech Off*" (Figure 7). The text that has been recognized is displayed in the central pane.

Figure 7: Speech Recognition Window

Our speech recognition engine is based on the Microsoft libraries SAPI 5.1 [2]. The libraries can be used easily through Visual Basic. After the installation of the SAPI a new 'control' is

available at application level, through which we implemented the training of the speech recognition engine for a certain speaker.

2.4.3 Dialog Mode

In this modality, besides the speech recognition function, we can also type in directly the text that we wish to send. In this case, when the user has finished, he has to press the "*Send*" button (Figure 8).

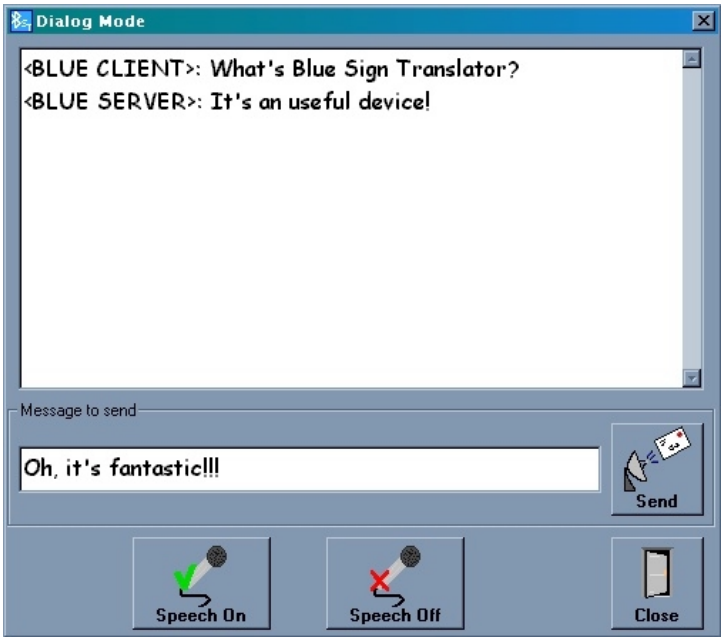

Figure 8: Dialog Window

The request to start a dialog session can be initiated either by the Server or by the Client. In this case, the window is opened automatically only if the checkbox "*Availability for incoming dialog request"* is selected (Figure 6). In such situations, the "Dialog Request Code" is specified in the header (Table 1).

2.4.4 Single Message Mode

BSS can be used in particular environments such as stations, airports, postal offices, banks, etc. In all these places, the types of messages that are sent through the loudspeakers are usually

standardized. For example, a station may want to announce that the departure of a train has been delayed, or a store may want to communicate that is going to close in a few minutes. In this case, the Blue Sign Service can be added to the exiting system as a plug-in that routes the original messages to the BSS.

The messages can also be set independently from other equipment through the "*Message Manager*", one of the tools available in the "*Single Message Mode*" dialog window (Figure 9).

Furthermore there are two options: it's possible to choose how many times we want to re-sent the same message and periodicity of this event.

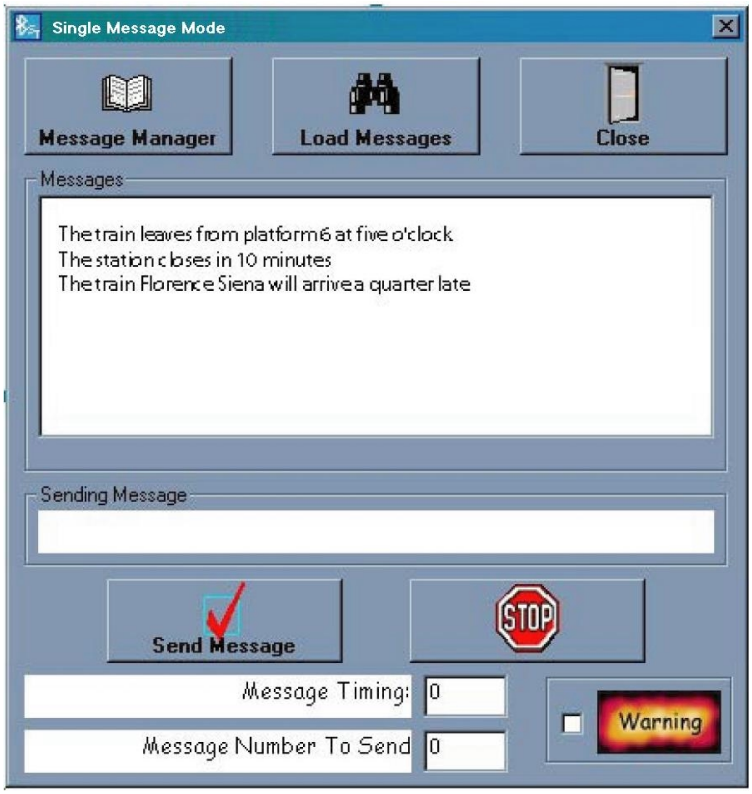

Figure 9: Single Message Window

Message Manager

The management of the message database has been implemented in a separate Visual C++ module. Classical Database management functions are available to the user, like inserting a new message or visualizing the database content, through a simplified User Interface (Figure 10).

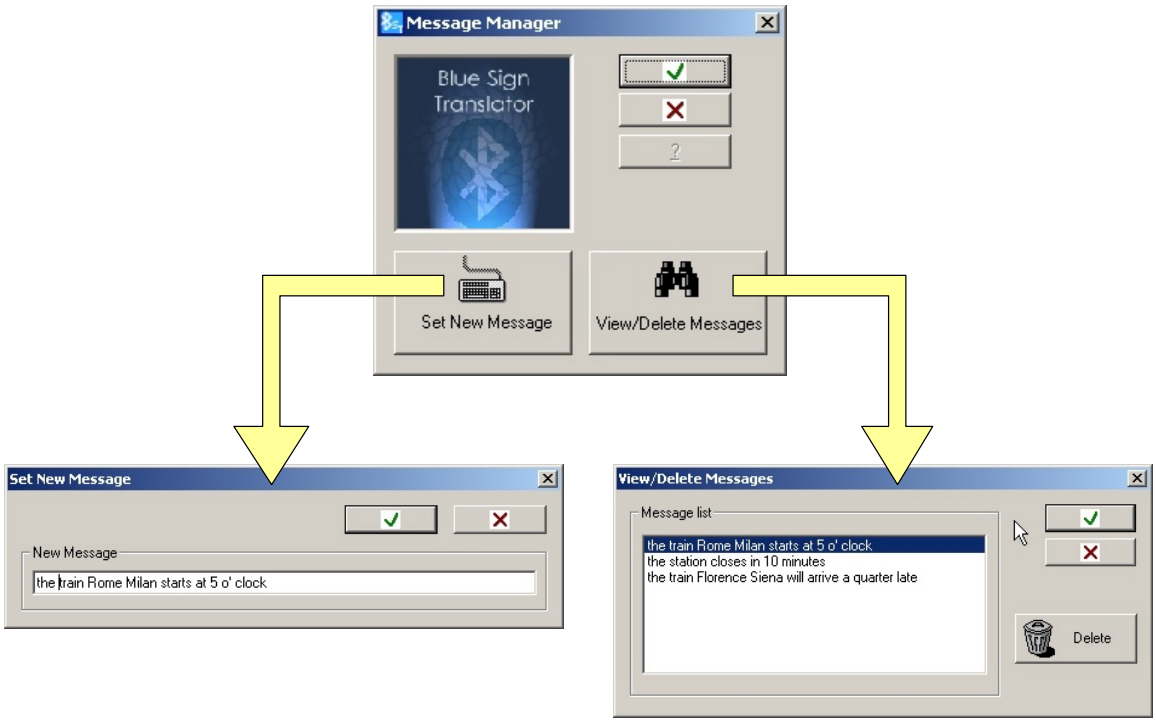

Figure 10: Message Manager

Messages are showed in alphabetic order and it is possible to select and eliminate them by the "*View/Delete Messages*" dialog window.

2.4.5 Warnings

A special function permits the transmission of high-priority messages, as warning messages. Selecting the "*Send Warning*" option present in both dialogs "*Single Message Mode*" and "*Speech Recognition Mode*" activates this function. Warning messages are characterized by a special header code (Table 1).

2.4.6 Trade-Offs

We have improved the speech recognition process and adapted it the requirements of the **Translation Engine** (§2.5.1) running on the Client, as explained below.

By default, SAPI libraries wait a fixed time before starting sentence acquisition. We have implemented an optimization of the process, since the Translation Engine is able to translate longer sentences in a single video. We limited the length of a translation sequence at 200 characters or whenever the user stop talking for more than 5 seconds. In this way, we achieved a higher fluidity of the video playback.

2.5. Blue Sign Client (BSC) Development

The Blue Sign Client (BSC) is the software that manages the operations on the client side. It has been developed in Visual Basic and Visual C++ modules. In particular, User Interface, Bluetooth connection management, and video playback were developed in Visual Basic. The **Translation Engine** and the database management programs were developed in Visual C++. We also made use of the ***Dynamic Link Library (DLL)*** to interface some of the application modules.

2.5.1 Translation Engine

The goal of *Translation Engine* is to perform the translation of English sentences into Sign Language sentences. The input for the engine is a text string and the output is the list of video clips, which the BSC has to playback in sequence.

Translation engine works by using three different dictionaries: sentences dictionary, word dictionary, and verb dictionary. Each element in all dictionaries contains an item and its translation in Sign Language. When engine initialization runs, items are dynamically loaded in three b-trees. We have chosen the b-tree data structure, since it makes search operations very fast. When the translation of a certain string is required, for example “hello world”, the Translation Engine checks whether the string contains a sentence belonging to the “sentence

dictionary". If the search result is negative, a new research will be started in the "word dictionary", and then in the "verb dictionary". If the result is negative again, the string will be decomposed in single characters and they will be translated respecting into the corresponding Sign Language letter. This way the engine is always able to generate a video sequence that can be understood by the user. At the end of this process, the video queue contains the entire movie sequence corresponding to the input string (Figure 11). Finally, we send the video sequence to Windows Media Player to obtain its visualization.

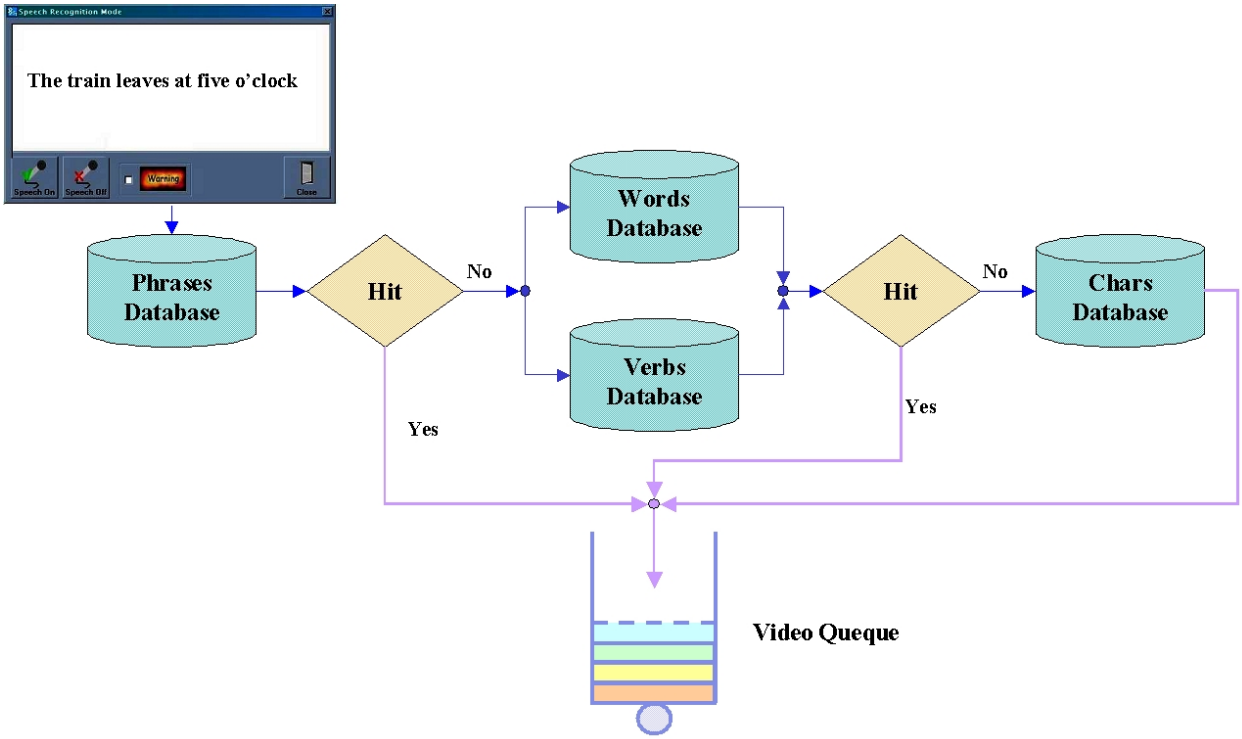

Figure 11: Translation Engine

The type of search performed in the three trees is more complex than a simple alphabetic search. For example, the engine is able to notice that a sentence matches in the sentence dictionary, even if it has a different number of spaces, a different punctuation, or a different number of small-capital letters. The same applies to word and verb dictionaries. Moreover, verb dictionary is able to recognize different tenses of a verb. For example, if the dictionary contains the verb "to attend", the engine is able to recognize that "I'm attending" is present continuous

tense, “I’ve attended” is simple present perfect tense and so on. Moreover, the engine is able to recognize irregular verbs. As a consequence, a verb is translated using two signs: the first one corresponds to the verb, and the second one to the tense.

2.5.2 Trade-Offs

Our implementation of the Translation Engine does not pretend to be exhaustive, since the goal of our project is to show how we can implement our basic idea.

We have also tried to solve the important problem of ambiguities. In fact a word that is separated from the phrase context can be analyzed in different ways. For example the word “fly” is both verb and noun. Another difficulty is caused by semantic ambiguities. For example “get” is always used as a verb, but it can assume different meaning as a phrasal verb. Thus, analyzing single words cannot solve many of these ambiguities. A correct translation requires that the ambiguous word would be compared with the nearest words in the same sentence. This permits the translation of the most common sentences. A perfect correspondence between spoken language and Sign Language does not exist, which means that the same word can be translated with different signs, even if it is not affected by grammar and semantic ambiguities. For example, in translation of the expressions “ten years” and “count up to ten”, the same word “ten” is usually translated with different signs, even if it has the same meaning in both expressions. It happens because “ten” comes from two different contexts.

Another tradeoff regards the implementation of the translation of only the Sign Language translation. We did not implement Translation Engines for other Languages.

2.5.3 DLL Development

We implemented some module as Visual C++ Dynamic Link Library (DLL) [3], [5]. The interfacing with the other modules is achieved through the use of a public function that is then called from Visual Basic modules. The Visual Basic functions call the DLL public function with

the message to be translated as input parameter. This function returns a list containing the names of the video clips.

2.5.4 Initial Set-up

For implementing our prototype, we have used a notebook as receiving device. We developed a Graphic User Interface that reproduces the interaction with a PDA.

The services we proposed do not require any cryptographic transmission system, because in all scenarios that we thought, the use of a system BSS-BSC would just replace a public communication.

The user configures the BSC in order to receive the Blue Sign message as Video ("*Video Mode*") or as Text ("*Text Mode*"). Also, choosing either "*Video Dialog Mode*" or "*Text Dialog Mode*" enables interactive modes.

When activated the BSC software seeks the available services in the area. At this point, the user can selecting the desired Blue Sign Piconet and eventually receive messages (Figure 12).

Figure 12: Client operation. A: configuring the preferred output mode. B: selection of the Blue Sign Piconet. C: playback of the video in Sign Language

2.5.5 Video Mode

This modality has been designed as a translation service into Sign Language for deaf people. Received messages are not displayed as text, but they are further processed. Strings are transferred to the Translation Engine, which generates the video-filename list as output. The video is opened by an ACTIVE-X Windows Media Player control (Figure 13).

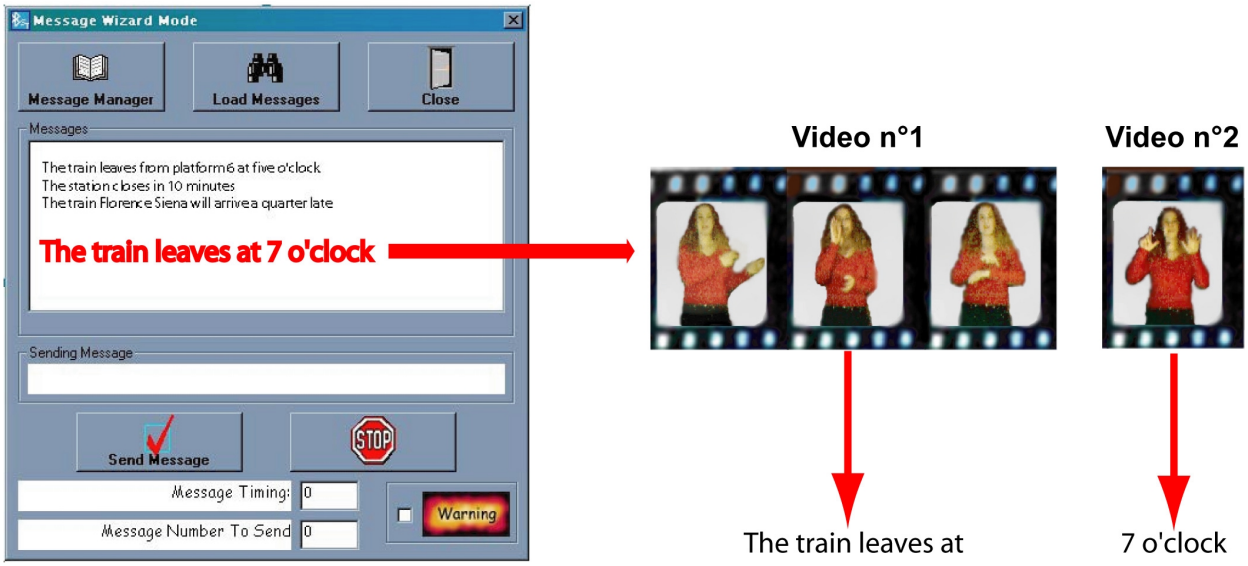

Figure 13: Client Video Mode.

This control plays the first video of the queue. When its execution stops, the video is removed from the queue and the next one is played. For a fluid reproduction, we developed an algorithm, which utilises *Media Player Control* events.

Video Casting

Part of our job regarded video casting. We produced linkable video clips with the help of Sign Language experts. We developed a significant set of videos for our prototype (about 100 clips). A good quality/compression ratio has been achieved through the use of AVI format. This is compatible with many compression codec, and allows for the production of small and good

quality clips. A standard message of 10 words has an average playback time of 10 seconds. The size of the clip is typically smaller than 300 KB.

2.5.6 Video Dialog Mode

Like in the *Video Mode*, this modality relies on the *Translation Engine*. It enables the deaf to playback Blue Sign videos and also allows to type in short messages to interact with the server (Figure 14).

Figure 14: Video Dialog Mode

2.5.7 Text Mode and Text Dialog Mode

These modalities are useful for simple devices where the video option is not requested or implemented. The Text Mode has been implemented mainly for the use of the BST as generic language translator. The client should take care of the translation into desired language.

In the case that the user is a hearing person the device could be embedded in a cellular phone and the playback of the message could be performed as streaming audio.

The Dialog mode allows us to type in messages, as seen in the previous paragraph, for interaction with the server.

2.5.8 Warnings

When the received message contains a warning code (Table 1), then the BSC generates a special signal (Figure 15), like a special video clip and possibly a vibration.

Incoming warning messages interrupt all client activities since this is a message of high priority for the user.

Figure 15: Incoming warning message.

2.5.9 Optimizations

- In the future, video clips could be standardized in each country by deaf associations.
- Users could filter out the types of messages according to his preferences.

2.6 Performance Considerations

The proposed architecture is very flexible and can be changed based on the service level.

2.6.1 Blue Sign Server

We implemented a sample application that uses a PC as Server. A minimal configuration required for sending messages, which are edited by a keyboard or selected in a database is proposed, and finally a configuration necessary for voice recognition is suggested.

Minimal Configuration

- Operating System: Microsoft® Windows® 98, Me, 2000, or XP
- 32 MB of RAM
- 200 MHz processor.

Recommended Configuration

- Operating System: Microsoft® Windows® 98, Me, 2000, or XP
- 256 MB of RAM
- 350 MHz processor.

The most troublesome aspect in implementing our system was voice recognition. Obtaining good results was possible only after personal training. We think that a cellular phone would be ideal for implementing a client since “it listens at our voice” for hours and hours! This could be a good opportunity for a commercial implementation. At public offices, where an employee almost always works on the same machine, we can just install a software version for PCs of our system.

2.6.2 Blue Sign Client

The features of the minimal system for the client should be the following.

Minimal Configuration

- Operating System: Microsoft® Windows® 98, Me, 2000, or XP
- 64 MB of RAM
- 200 MHz processor.
- 32 MB space available for storage

2.7 Tools developed during the project

2.7.1 Client Message Manager

Client Message Manager is a utility to manage the Translation Engine Database. Translation Engine chooses part of the message to be associated with a video clip. The main purpose of this tool is the management of the association of the chosen video clips with their filenames (Figure 16).

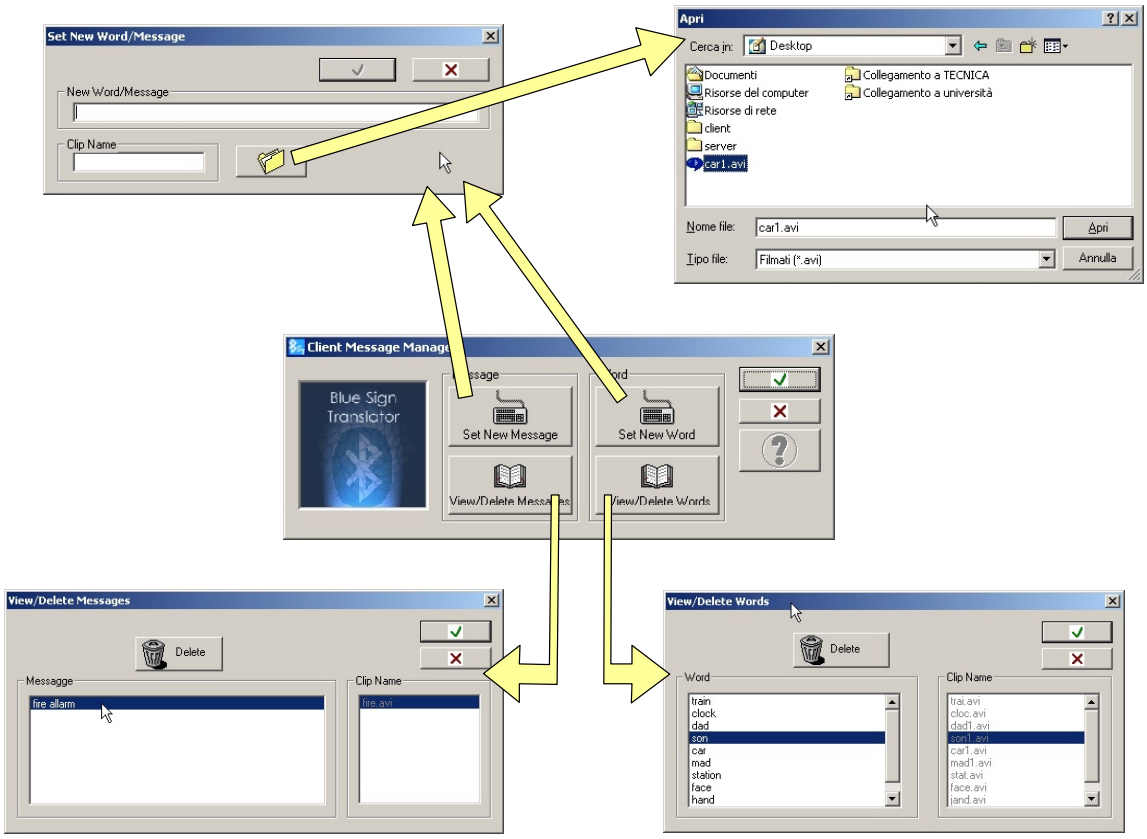

Figure 16: Client Message Manager

This utility is used only in cases when user has to add specific terms to the standard dictionary. The program can be used in two different modes. If the Client is installed on a PC (like a notebook), the Client Message Manager is integrated in the BSC. If a PDA is used

instead, the Client Message Manager runs on a PC, which connects to the PDA when we need to update the video dictionaries. Video can be also downloaded from an Internet support site.

2.7.2 Other tools that we were developed during the project

The following tools have been used to test each function and procedure before their integration in the main project. This procedure substantially decreases the development time.

Animation Tool: This tool has been used for adjusting the streaming of video files.

Text To Sign Tool: This tool has been used for checking the Translation Engine. It provides a simple text-box for the input data. The entered phrases and words are put into a text file, which corresponds to a video filename. If the filename matches, the right video name is displayed; otherwise, the spelling of the input messages is generated.

Speech Tool: This tool has been used to test the Speech Library and Speech Recognition functionalities. Microphone receives voice and sends the recognized text to the Translation Engine.

2.8 Testing

For the purpose of testing the whole system, we tested its efficiency in a "real world" situation: we visited the primary school with deaf students in Siena. ***Deaf students personally tried the BST system and they were satisfied.*** Teachers were really interested to find out more about BST, and believed its usefulness in education. In schools with deaf students, BST system could be indispensable support in translation. The use of Blue Sign system would only require supplying each teacher with a microphone, and each student with a Blue Sign Client device. The Blue Sign Server could be installed in a existing PC.

We would also like to remark the fact that ***BST does not require new cables*** to be mounted in the school building. **The BST system could be used in many types of lessons**, as it offers the opportunity to manage personal profiles for each teacher. Also, it is worth notifying that the system offers the possibility of avoiding the presence of an interpreter, who often disturbs students during a lesson.

2.9 Marketing considerations

In order to obtain more reliable marketing information, we contacted national associations and agencies specialized in teaching to deaf people. Since we wanted to obtain deeper understanding of the international approach to the problem and technology support for deaf, we took into consideration also the opinion of students who had the possibility to study abroad (e.g. University of Gallaudet, DC, U.S.A, a specialized University for the deaf).

In order to analyze the opinion of the students and the University staff, we sent to a local Institute a brochure with synthetic presentation of Blue Sign Translator followed by a questionnaire.

The feedback showed that the possibility of using the BST system certainly aroused remarkable interest. Many interviewed people emphasized the importance of a future

development, and the most common request was that receiver should be made simple and manageable (Table 2).

Table 2. Answers received from interviewed people.

Answers received from interviewed people		
Impression on BST	Very good	87%
	Interesting	13%
Proposed places to use	School	22%
	Conferences or public debates	39%
	Additional educational purposes	39%
Willing to know more	Yes	92%
	No	8%

2.9 Evaluation of design costs

The development of our prototype system did not require expenses neither for hardware nor for software. We used two personal computers in our possession, and Bluetooth hardware that was shipped with the Kit; as for the software, it was developed using Visual Studio .NET package. Although the implementation of the BSC using cellular videophones or PDA devices is suggested, our software also runs on regular PCs. As for the BSS, the integration with a cellular phone would be ideal in order to maximize the resolution of the voice recognition. The software could be used in office PCs or public agencies. In this case, the only expense would be installation of the Bluetooth hardware and of course of the BST software itself.

3. Summary

We decided to implement the **Blue Sign Translator** mainly as an aid for the deaf, because we believe that the technological progress must be at the service of the mankind. In fact, we believe that a society can claim that it is effectively evolved especially when it succeeds to guarantee the same rights to all people. This includes to take care of the less fortunate people by using all the possibilities offered by the technology.

The Blue Sign Translator was invented after a long search and documentation on the deafness: this has made us to inquiry the opinion of experts who understand what are the major problems in this field. We have come to know that great part of the deaf people will never master the use of spoken language perfectly and will use the Sign Language in order to communicate with the other people and to participate to the social life.

The name "**Blue Sign Translator**" synthesizes the idea for which the system has been invented: the Bluetooth technology and the translation into Sign Language or in any other language. The Bluetooth technology allows for a universal use in various situations and a greater mobility due to the wireless connection. The Blue Sign Service implement upon it allows for a flexible way of communication, in a number of different scenarios. We considered and implemented a Translator into Sign Language, which results as a very important tool for the deaf, in order to feel independent and integrated in his own society.

4. References

[1] http://www.cstack.com

[2] http://msdn.microsoft.com

[3] http://www.allapi.net

[4] http://vbsimple.virtualave.net

[5] http://www.vbapi.com

[6] Advanced Microsoft Visual Basic 6.0 seconda edizione – Microsoft Press

[7] C terza edizione – Schildt S. – McGrawHill

[8] Documentazione PDF "Programmers references for the handybluestack Com object"

[9] Linguaggio C++ - Schildt H. – McGrawHill

[10] Microsoft Visual C++ 6.0 Programmer's guide – Microsoft Press

[11] MSDN per Visual Studio 6.0

[12] Visual Basic 6 – Cornell G. – McGrawHill

[13] Visual Basic 6 Tutto & Oltre – Thayer R. – Edizioni Apogeo

[14] Visual C++ 6 Guida completa – Chapman D. – Edizioni Apogeo

[15] Visual C++ 6 - Pappas C.H., Murray W.H. – McGrawHill

Special Thanks

- Prof. Roberto Giorgi – for his availability and his useful suggestions.
- Dr. Irina Branovic – for the her help with translations
- Ing. Alberto Raggioli – for his useful lessons on the Visual C++ programming.
- Prof. Marino Bennati – President of A.I.E.S. (Italian Association of deaf teachers) for the useful suggestions on the education of the deaf.
- Prof. Anna Chiantini for her pedagogic and didactic contribution.
- Dr. Chiara Bennati and Dr. Sandro Sicilia for the help in making the video clips
- Institute for the Deaf 'A. Provolo' (Chievo – Verona) for testing the final product
- Institute for the Deaf 'Figli della Provvidenza' (Modena) for testing the final product