Remote Assistive Interpersonal Communication Exploiting Component Based Development

C.Viglas[1], C.Stamatis[1] and G.Kouroupetroglou[1]

Abstract

Assistive technology has been committed to offer a great deal towards facilitating the communication needs of the disabled. However, existing developments tend to focus primarily on face-to-face communication, thus leaving remote communication possibilities unattended or just "put aside". Remote communication for the disabled however, namely the ability to correspond over a public or computer network and especially the Internet, either by transmitting written messages, voice mailing or even participating in a teleconference, is of paramount importance nowadays. In this paper we present the development of a new generation of flexible, open, adaptable, configurable and cost-effective aids supporting both interactive and non interactive remote/ distance interpersonal communication. Details on the requirements, properties, components and services for remote interpersonal communication along with specific component development are given. The implementations presented comform to the recently introduced ATIC (Access to Interpersonal Communication) application development framework which exploits Component Based Software Development technology to maximize modularity and reusability.

1. Introduction

Interpersonal communication, which is taken for granted in our everyday lives, constitutes a crucial issue for many disabled people. Commercially available, computer-based communication aids are usually oriented towards providing partial solutions to specific communication problems for individual users. In most cases they offer from little to none functional flexibility with respect to their adaptation to either different or evolving user needs [2]. Furthermore, current trends demonstrated a prominent preference to matching the disabled user's requirements for face-to-face communication, leaving almost out of reach any consideration for communicating remotely. Namely, the communication aid is considered a computer-based application that solely satisfies the needs of the disabled for communication and interaction with their social or work environment. This in turn, prevents from taking advantage of what contemporary technology offers, namely accessing the Internet, and thus being able to communicate without barriers, with anyone regardless of the communication partners physical location or concurrent on-line presence (that is, harnessing the benefits of electronic mail, messaging or even voice mailing and real-time conversation via tele- or video- conferencing facilities).

It is clear however, that over the past few years the communication requirements of humans have undergone quite a change [8]. Face-to-face communication, although remains highly desirable, is no

[1] University of Athens, department of Informatics, Panepistimioupolis, Ilisia, Athens, GR 15784, Greece,
E-mail: koupe@di.uoa.gr

longer the only predominant need to be matched. People tend to use computer networks more and more utilizing electronic mail and conferencing facilities to *communicate remotely* with friends and associates from around the globe. In addition, the World Wide Web (WWW) is rapidly becoming the universal delivery medium affecting tremendously the domain of interpersonal communication as well [16]. Furthermore, the developments in electronics and mobile telephony makes reaching and utilizing the telephone device possible for a lot of people with motor disabilities. These new developments have not been efficiently tackled yet by the contemporary communication aids' market.

In this paper we present the development of aids supporting both interactive and non interactive remote/distance interpersonal communication. The implementations presented comform to the recently introduced ATIC (Access to Interpersonal Communication) application development framework [6] which exploits Component Based Development technology. We have shown previously that ATIC framework, leads to a considerable change in the way communication aids are looked upon and moreover affects the overall process of developing communicators into more effective products [12].

In the following sections, after the introduction of the requirements for remote communication in the assistive technology domain, a short presentation of the ATIC framework is given, followed by the technical solution we have devised for the implementation of communication aids with remote interpersonal communication support, with emphasis put on describing reusable components.

2. Requirements for Remote Communication for the Disabled

Aiming to develop communicators that would offer additional features than the average contemporary aid, and would be more efficient in dealing with their users' characteristics, a number of critical issues were revealed as a result of thorough studies on technical specification of user needs and requirements on the one hand, and the features offered by commercially available communication aids, on the other [1]. The analysis of these results led to the identification of the properties for distance interpersonal communication for the disabled that can be mapped to appropriate components and services.

2.1 Identified Properties for Remote Communication

The communication aid should cater for different modes of communication, including:

- Message exchange (be that voice mail, e-mail, written or spoken) between a disabled person using a symbolic or orthographic language and an able-bodied individual and vice-versa.
- Message exchange between disabled people using different symbolic or orthographic languages.
- Interactive remote conversation (in the form of interactive chat or teleconferencing), between two disabled people or a disable and an able-bodied one, using, again, different languages.
- Use of the communication aid by a speech impaired person to communicate through a telephone device with an able bodied individual using the speech synthesizer of the communication aid in order to output voice to the common telephony network.
- Face-to-face communication between disabled persons and between a disabled person and an able bodied individual.

To design a product that targets on the international market, dealing with not converging user needs in various counties, with not the same cultural backgrounds and using different languages, first of all multilinguality support of a number of alternative/symbolic and natural languages is imperative. As far as adaptability is concerned, the selection of the proper language or communication system and associated user specific vocabulary are crucial. At the same time one has to ensure that people using different languages and systems will not be prevented from communicating (locally or remotely). To maintain effective communication, machine translation should be supported, either word by word or based on a common interlingua meaning, using proper techniques to overcome mismatching vocabularies [3,5]. Multiple output modalities, such as speech, audio, visual, printed, should be supported. The aid should also be adaptable and flexibly configurable at a lexical user interface level [6,15,17], to match the user's motor, sensory and mental needs and abilities. Moreover, it should also offer extensive configuration aspects in terms of: input and output language, input and output devices, and user vocabulary selection, along with a high degree of flexibility and adaptability to the -changing with time- user needs. Needless to say that there are even more issues to be considered, like maintainability and ease of operation.

The aforementioned analysis shows a great diversity as far as disabled user communication needs are concerned, making it extremely unlikely to adopt a general-purpose solution towards implementing communicators. To achieve this complicated goal, *component software development* is adopted, as it does offer maybe the only viable solution to cost-effective products. This approach offers adequate interoperability to come up with a sufficient range of items/parts ready to be used off-the-shelf, enabling the desired versatility in a communication aid.

2.2 Identified components and services

The analysis of the previous paragraph leads to the following components and facilities communication aids should support: establish connection over a PTSN network, establish connection over a computer network (namely maintain a TCP Internet connection), handle a communication partner address book, maintain and manipulate a message/mail/voice-mail box, and transmit/receive a message over a network. Such facilities call for either voice output if connected to the common telephony devices, or proper modality (text, voice, graphics, etc.) according to the given circumstances if an Internet-based communication is desired.

On the other hand, a number of general purpose components have also been identified and should be considered as they offer basic functionality one would expect to find in any computer-based communication aid. Naturally, the specific components that would comprise a certain communication aid will depend upon the special features and functional characteristics we would like to be present in the communication aid in question. Typical components would support -at least- the following services [3,10]: vocabulary definition, symbol selection, message reception, sentence/message composition, multilingual support and translation between different user languages in a variety of output modalities (text, sound, voice, graphics), message output in a number of ways (such as transmission over a communication channel, speech production, on screen display, and printer output).

Additionally, there exists a set of equally important components also considered as desirable but not always mandatory, providing input acceleration facilities (usually implemented using word prediction techniques), an editor for the construction of symbols of non orthographic (symbolic) languages, and training on the user's communication system or language.

3. Development Framework for Communication Aids

3.1. Component Based Development

Component based development (CBD) is at present the most widespread technology for application development. It refers to the practice of reaching software solutions by building or buying interoperable components [16]. CBD, is quickly becoming the dominant model for software development, as it has already been ranked most favorable amongst software deve lopers. Supporters exist not only in the generic application domain, but also in the domain of commercial applications and databases. CBD, owes in part it widespread acceptance to the *Internet* which is responsible for almost all upheavals in recent developments. The Internet has influenced commercial and business logic and to support that in a cost-effective manner, component technology seems the ideal solution [16].

CBD, promises and delivers considerable investment return in terms of ease of use and training, to both commercial and corporate developers and vendors such as Microsoft, Netscape Communications and Sun Microsystems who are already actively involved in the CBD domain. On the hand, CBD has broken into rehabilitation technology as well, in a quite remarkable manner, having various developers putting a considerable amount of effort in the interpersonal communication domain in particular [4,10,13].

Furthermore, by demonstrating a high degree of reusability (both at the code and application level) and harnessing the advantages of Object-Orientation, CBD has been identified as offering a balanced flexibility and adaptability thus leading to modular and open products that comply to such requirements as: interoperability, modularity, distribution, extensibility, and independence from programming language, as far as building a component application is concerned [9,16].

3.2. Fundamentals of the ATIC approach

The implemented ATIC framework, aims to be used by a number of different vendors, harnesses the benefits offered by the component-based technology, and introduces a new perspective in building better, feature-rich and more manageable communication aids, affecting drastically their life cycle [11,12]. It endorses a modular design for communication aids complying to a general adaptable solution to serve a wide range of users, able to deal adequately with users having motor, perceptual and cognitive problems. A versatile technology is used [11], allowing for rapid changes, supporting multiple features in order to meet the physical abilities, cognitive and language levels and conversational needs of persons using these aids [17]. Any user language, symbolic or natural, is supported via a database of multimedia language elements, to be utilized in all aspects of the disabled user's communication needs, be that face-to-face or remote communication and even group conversations (i.e. support for multiple output modalities). In order to reach the aforementioned goals, the ATIC approach, utilizes, and is based upon the provision of a novel component architecture (defining communication protocols, components connectivity, building rules, and software tools inclusive) [10,12], and the implementation of a number of components (modules) complying with the new architecture that can be interoperably used to built communication aids.

All in all, ATIC constitutes a development framework which promotes reusability in the design, the code generation and binary execution. These features render our framework in an advantageous position as far as communication aids application development is concerned. Taking advantage of its Object-Oriented design [11] and the features of CBD, the ATIC framework is demonstrates a set notable advantages that are often demonstrated by object-oriented frameworks [7] leading to:

- better and more concrete control over the application development process,

- greater reusability and adaptability potential for the developed components,
- applications that can demonstrate either a general purpose or a highly specialized character,
- even more robust and customizable user interfaces,
- ultimately, bigger productivity and variety of implementation, which can lead, in the long run, to a vaster range of choice but also to better quality (due to competition), which is of utmost importance as far as interpersonal communication for the disabled is concerned.

According to ATIC, any identified user needs are mapped to appropriate communication functions and translated into the software-oriented domain. The main functions identified are then broken down into sub-functions leading to *elementary functional elements* (n.b. *services*) that the communicator's components would be expected to offer and serve]. In this context, a *communicator is considered a system providing a number of* ***functions*** *and/or* ***services*** *dependent on the particular user-needs, abilities and cognitive level. Each function may be implemented independent of others, either as a separate entity or as a set of elementary services, offered by a component or set of interoperable components responsible for the implementation of a function or service in a way transparent to the architecture and the communicator itself.*

The ATIC framework defines both the programming model and the required binary standard for creating, managing, and accessing object-based components that provide (and use) services to other objects and applications. As a technology, it allows components to interact across process boundaries as easily as objects interact within the same process. This is enabled by means of a message registration and passing mechanism. ATIC does specify an object model that supports concurrency, re-entrant multi-threading, internal synchronization of processes and inter-process communication of components, that allow for component applications to be built using virtually any programming language [11].

4. Remote Interpersonal Communication Implementation

Remote communication was implemented under the ATIC component software framework and its Object Model (using OMT notation [14]) is presented in Fig. 1. Two different kinds of objects can be distinguished: the *module* object, which represents a component of the interpersonal communication aid responsible for the implementation of a specific function, and the *ATIC* object. A more general object, the *Communicator*, represents the entire communication aid.

The *ATIC* component is responsible for any information exchange, plus error handling, between the communicating components. All information between the components and the *ATIC* is implemented through message exchange: *ATIC* receives messages from the "client" component, decides which component is the proper recipient (more than one modules might be affected by a certain message), and transmit the necessary message to the "server" module(s). This communication is represented through the "message exchange" association [11].

Aggregation is identified between the *Communicator* object the *ATIC* and the component objects. More specifically, the *Communicator* is an abstract object, which consists of one ATIC instance and a collection of instances of the Module class. For this reason, the various instances of the Module object, which are most crucial in the model, are discussed in the following sections, using top-down generalization. Subclasses of the Module class can be derived by identifying the desired functionalities of a most generic (all-encompassing) communicator, and then decomposing these functionalities to elementary functions. Each function is independent of the others and unaware of their presence or absence in the communicator system (if a service that is not provided is required, the

ATIC component is responsible for sending the appropriate error messages). This is the reason why no association or link whatsoever exists between *Module* subclasses. Communication between them is established through message exchange via the *ATIC component*.

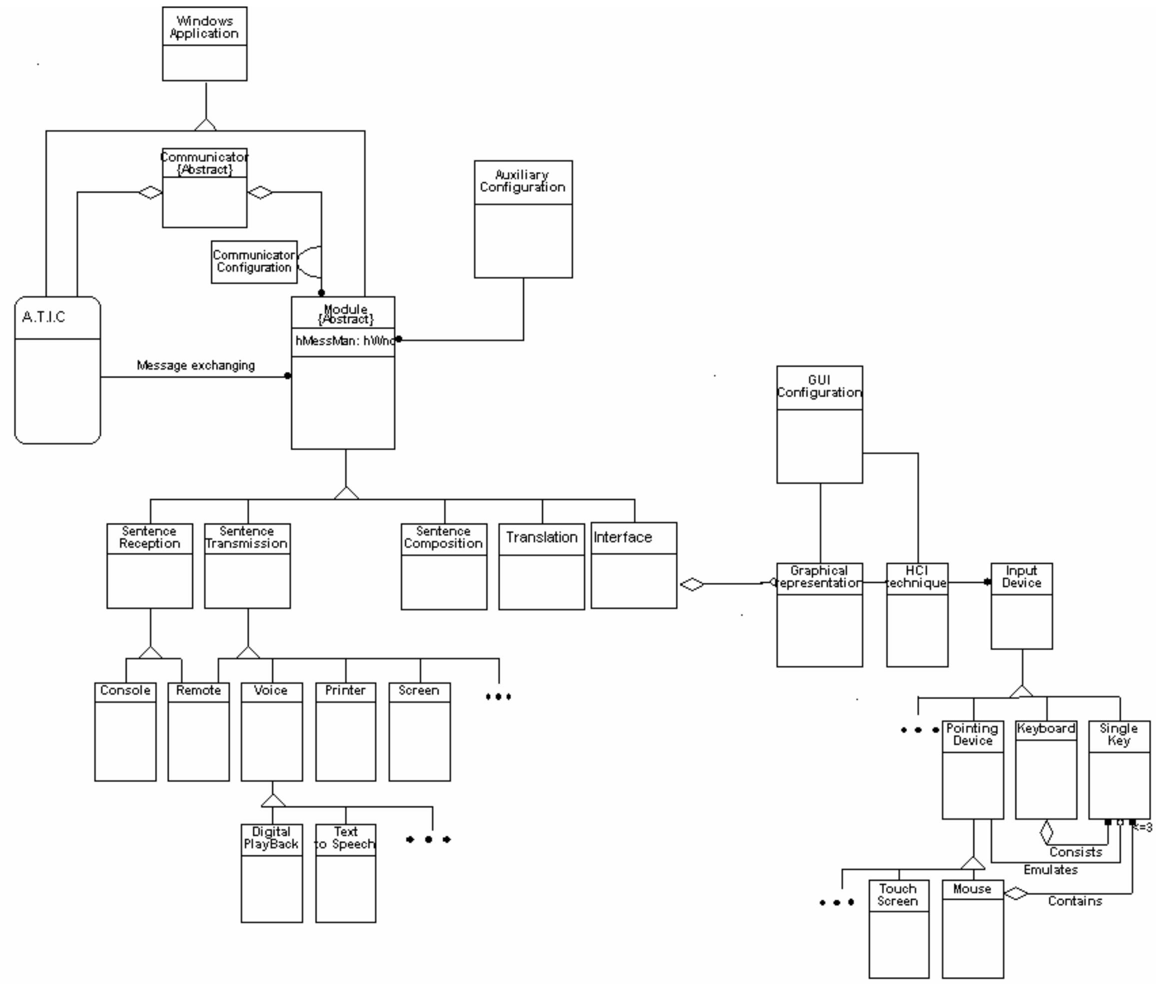

Figure 1: The Object Model for remote communication

4.1 The Module object

The Module object is generally concerned with the implementation of certain services that the communicator is expected to provide. Its functionalities and specifications depend on the particular component that is being developed.

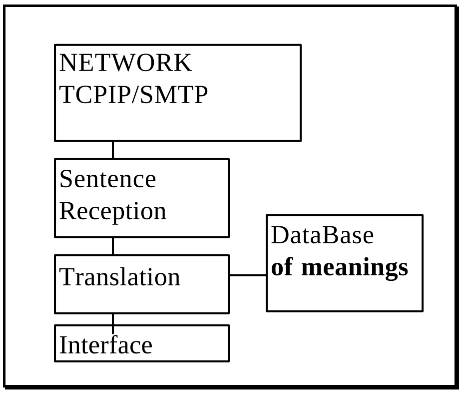

Figure 2: Message reception

In the case of distance communication, two different functional modes can be identified: the sentence reception mode and the sentence transmission mode. The first is accomplished through the network by using the SMTP protocol in the case of E-mail (Fig. 2). The message is received from the SMTP port by the Sentence reception component. This component firstly removes the header of the message and then it breaks the message in a set of discrete meanings associated with the original message composed in the user's own language. The meanings are passed through the ATIC component to the translation component by the use of a message called TRANSLATE. The translation component, incorporating a database of meanings, translates the meaning to the proper user language and passes it to the interface where the message appears in an editor. Almost the same scenario takes place in the case of interactive chat. The difference is that the received message is broken down in meanings, each individually carried in TCP/IP Internet packets. Then, each meaning is passed to the translation component and presented in the proper user language. This process takes place in real time as the meanings composing the user message arrive.

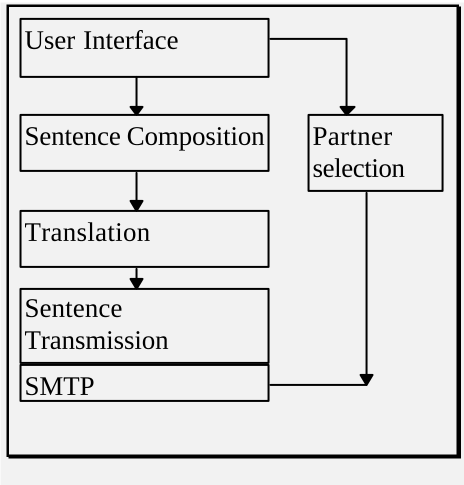

Figure 3: Message transmission for email

Sentence transmission mode is more complex. It begins with the user taking advantage of the services of the message composition component through the user interface component. The message is broken in different words. The discrete words are passed through the ATIC component to the translation component by use of the TRANSLATE message. The translation component using the database of meanings translates the message to a set of meanings which are passed to the proper network protocol, such as the Simple Mail Transfer Protocol (SMTP) in the case of e-mail and TCP/IP in the case of interactive chat. Note that the sentence transmission component is different in each of those cases. In the first case (Fig. 3), the sentence transmission component passes the whole message (as one chunk of text) to the mail protocol. The selection of the user's communication partner is handled by the user interface module, where the user chooses a graphical representation of his/her partner. This is then mapped to the corresponding e-mail address by the Partner Selection component and then passed to the mail protocol. In the second case (Fig. 4), the message is passed word-by-word to the TCP/IP layer which in turn acknowledges reception to the Sentence Transmission component. In the case of a communication error the, User Interface component gets a notification of the error that occurred. Selecting a communication partner in this case is also handled by the User Interface where the user chooses a graphical representation of his/her partner. This is then mapped to an IP address by the Partner Selection component which is responsible for passing it to the Sentence Transmission component. Then this IP address is used by Sentence Transmission component to establish connection, send, listen, check and retransmit

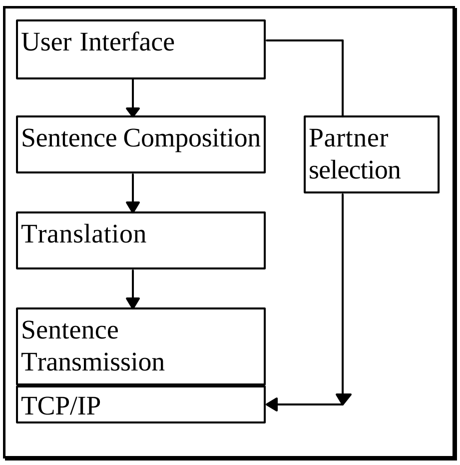

Figure 4: Message transmission for chat

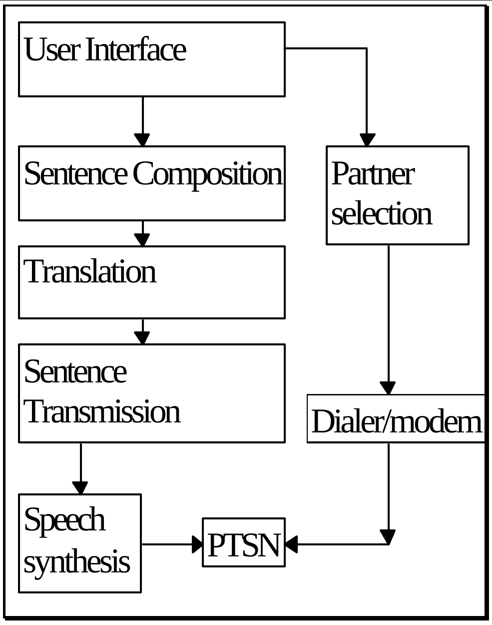

Figure 5: Message transmission for PTSN connection

packets utilizing the TCP/IP layer services. In the third case scenario (Fig. 5), the Sentence Transmission component passes the composed message to a speech synthesizer which in turn produces output to a telephone device. The communication partner selection is once more achieved in the same manner as in the previous cases. The Partner Selection component utilized in this case invokes a modem device to dial directly the desired partner. As soon as the connection is established the User Interface component is notified so that the user can start composing his/her message.

4.2 The User Interface Component

One of the most important components of the communication aid is the User Interface component. This component is compliant with the ATIC component software architecture. This means that it communicates with the other components using ATIC messages. The User Interface component registers the INTERFACE message which carries a structure containing the specific command issued by a functional component to be handled and intended for the Interface object. One of the biggest obstacles that have to be overcome in order to provide a generic solution to the user interface design in a component based environment, is managing the GUI of the application. This becomes more evident in a communication aid where the user interface has to have special behavior supporting many different scanning techniques and interaction modes. Taking into account that the user interface objects must respond to special devices and that different components will have to interact as far as user interface is concerned, having each functional component developer to embody the user interface into each component would make building a communicator extremely complex. The solution to that is to provide a separate User Interface component with a high level GUI definition language that would be interpreted in real time by a user interface server in order to provide and manage the GUI.

```
Screen 1{
backcolor R,G,B
group1{
button1,x,y,z,w,
R,G,B,bitmap.bmp,ATICMESSAGE,wparam,Lparam
.
.
buttonn,x,y,z,w,
R,G,B,bitmap.bmp,ATICMESSAGE,wparam,Lparam
List1, x,y,z,w
.
.
Edit1x,y,z,w

Group1.1 {
Group 1.1.1
Group {1.1….n
}
}
{
group2{
}
.
.
groupn{
}
}
Scrren2{
}
.
.
Screen n{
}
```

Figure 6: Example of the user interface definition language

The User Interface component of a

communication aid is structured, namely, there is a hierarchy amongst the interface objects that comprise it. Thus, the interface definition language should be a structured one supporting multilevel encapsulation of objects. The hierarchy defined in this encapsulation determines the order in which the interface objects are scanned. The first level of this encapsulation is the "screen" object. The Screen is the container which contains all the other objects. Within a screen, scanning groups are defined. The scanning groups are objects which can contain other groups or childless objects. A Childless object is a button a listbox or an editor. An example of the GUI definition language is presented in Figure 6.

5. Discussion

The aids for remote interpersonal communication described above have been implemented under the MS-Windows 95 environment for an IBM-PC compatible platform. They offer enhanced functionality by fulfilling all the requirements identified in section **2** of this paper.

The communication aids described in this paper were evaluated by speech motor and language cognitive impaired users in Great Britain and Finland within the ACCESS project as well as in Greece under the HORIZON project HESTIA.

Figure 7 depicts a typical remote communication scenario for two disabled people using ATIC compliant communication aids. More specifically, they reside different countries (Greece and Finland in the example), one is using BLISS as a communication system while the other is using Pictogram, and they establish a connection over a telephone or computer network.

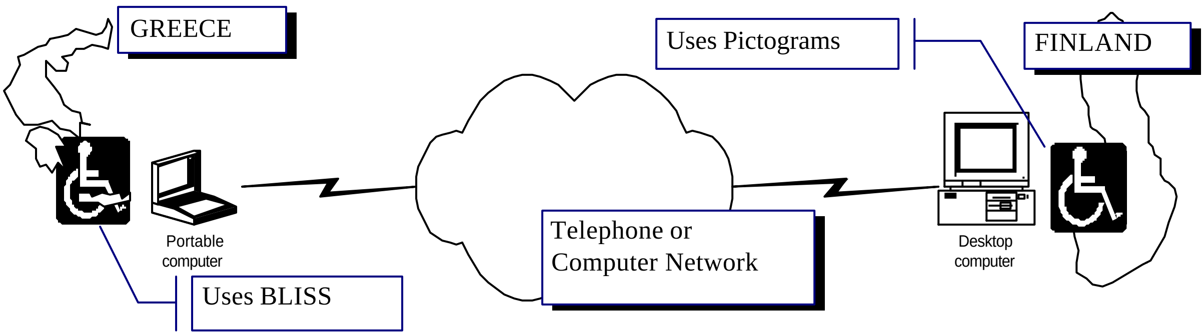

Figure 7: Typical ATIC enabled, remote communication

Acknowledgments

The work reported in this paper was carried out within the framework of the ACCESS project, partially funded by the TELEMATICS for the Integration of Disabled and Elderly people (TIDE) Programme of the Commission of the European Union, DG XIII. Partners in this consortium were: CNR (Italy), FORTH (Greece), University of Athens (Greece), RNIB (U.K.), Seleco S.P.A. (Italy), MA Systems and Control (U.K.), Hereward College (U.K.), NAWH (Finland), VTT (Finland), Pikomed (Finland) and University of Hertfordshire (U.K.).

References

[1] Ahonen, J., A.Kalliala, E.Kaasinen, G.Kouroupetroglou, R.Lauder, K.Nyholm, T.Pulli, V-P.Raty, A-L.Salminen, K.Salomaa, C.Stephanidis: "*Final Report on User Needs*", TIDE project 1001 ACCESS, Report D.4.1.2, Volume II, February 1995.

[2] Alm, N., A.F. Newell, J.L. Arnott, An Integrated Research Strategy for Improving Communication Systems for Severely Physically Impaired non-speaking People, Proceedings of the 1st TIDE Congress, pp. 249-253, 1993.

[3] Antona, M., C. Stephanidis, and G. Kouroupetroglou, Vocabulary Management in Modular Interpersonal Communication Aids, in: Advancement of Assistive Technology (AAATE) G.Anogianakis, C.Buhler and M.Soede (eds.), IOS Press, 1997, pp 200-205.

[4] Demasco, P., E.J. Ball, S.Tyvand, D.Blodgett, W.Bradley, J. Dunaway, Z.Kazi, Towards Modular AAC Software: An Object-Oriented Architecture, RESNA International '92, June 6-11, 1992, pp. 119-121.

[5] Dorr, B.J. Machine Translation: A View from the Lexicon, MIT Press, 1993.

[6] Emiliani, P.L., J.Ekberg, G.Kouroupetroglou, H.Petrie, and C.Stephanidis, The ACCESS Project: Development Platform for Unified Access to Enabling Environments, Proceedings of ICCHP' 96, (International Conference on Computers and Handicapped People), Linz Austria, October 1996, pp. 69-75.

[7] Fayad, M.E., and D.C. Schmidt, Object-Oriented Application Frameworks, Communications of the ACM, Vol. 40, No. 10, October 1997, pp. 32-38.

[8] Goodenaugh-Terpantier, Design Goals for Augmentative Communication, Assistive Technology, Vol. 6.1/1994, pp. 3-9.

[9] Johnson, R.E., and B.Foote, Designing reusable classes, J. Object-Oriented Programming 1, 5 (June/July 1988), pp. 22-35.

[10] Kouroupetroglou G., A.Paramythis, A.Koumpis, C.Viglas, A.Anagnostopoulos and H.Frangouli, Design of Interpersonal Communication Systems Based on a Unified User Interface Platform and a Modular Architecture, Proceedings of the TIDE Workshop on User Interface Design for Communication Systems, Brussels, July 7, 1995, pp. 8-17.

[11] Kouroupetroglou G., C.Viglas, A.Anagnostopoulos, C.Stamatis and F.Pentaris, A Novel Software Architecture for Computer-based Interpersonal Communication Aids, Proceedings of the 5th ICCHP'96 (International Conference on Computers and Handicapped People), Linz Austria, October 1996, pp 715-720.

[12] Kouroupetroglou G., C.Viglas, C.Stamatis, and F.Pentaris, Towards the Next Generation of Computer-based Interpersonal Communication Aids, in: Advancement of Assistive Technology, G.Anogianakis, C.Buhler and M.Soede (eds.), IOS Press, 1997, pp 110-114.

[13] Lumdalv, M., D. Svanaes, Comspec - Towards a Modular Software Architecture ans Protocol for AAC Devices, 1st TIDE Congress, 1993.

[14] J.Rumbaugh, M.Blaha, W.Premerlani, F.Eddy and W.Lorensen, Object-oriented modeling and design, Prentice Hall, 1991.

[15] Savidis, A., A. Paramythis, D. Akoumianakis, C. Stephanidis, Designing user-adapted interfaces: the unified design method for transformable interactions, ACM Conference on Designing Interactive System: Processes, Methods and Techniques (DIS '97), Amsterdam, The Netherlands, 18-20 August 1997, pp. 323-334.

[16] Spitzer, T., Component Architectures, DBMS,Vol. 10, No. 10, September 1997, pp. 56-68.

[17] C.Stephanidis and G.Kouroupetroglou, Human Machine Interface Technology and Interpersonal Communication Aids, Proceedings of the Fifth COST 219 Conference, Tregastel-France, June 7-8, 1994, pp. 165-172, ISBN 951-33-0004-8.

[18] Stephanidis, C., A. Savidis, D. Akoumianakis, Unified Interface Development: A step towards User Interfaces for All, in: Advancement of Assistive Technology, G.Anogianakis, C.Buhler and M.Soede (eds.), IOS Press, 1997, pp. 29-33.