A New Generation of Communication Aids under the ULYSSES Component-Based Framework

Georgios Kouroupetroglou and Alexandros Pino
Department of Informatics and Telecommunications,
University of Athens,
Panepistimiopolis, Ilisia, GR-15784 Athens, Greece
Tel: +30 10 727 5305, Fax: +30 10 6018677
{koupe, pino}@di.uoa.gr

ABSTRACT

In this paper, we introduce a new generation of computer-based communication aids, designed and developed using state of the art software engineering models and architectures. The communicators we present are based on a component-based framework called ULYSSES that aims to simplify the integration of multi-vendor components into low cost products and maximizes modularity and reusability. Following the ULYSSES approach, one can build up powerful and reliable applications, adaptable to various user needs and requirements. For *developers* of AAC components, ULYSSES provides an engineering-for-reuse environment with guidelines and tools to build software modules, which can operate effectively and interact with each other transparently, without even being aware of each other's existence. Furthermore, ULYSSES grants a process of engineering-with-reuse for AAC system *integrators* for the selection and assembly of components on demand to build user-specific robust communicators out of pre-fabricated software parts. Thus, adding or removing characteristics and features as needed, is becoming an easy task for system AAC systems integrators. Three complete Interpersonal Communication Aids are presented as cases of ULYSSES application in this specific domain.

Keywords

Augmentative and Alternative Communication (AAC), Communication Aids, Communicators, Component Based Development, Framework Architecture

INTRODUCTION

After an initial assessment of each disabled user's communicational requirements and taking into account his neurological, intellectual, motor, and linguistic condition, a rehabilitation program is usually formed, which should often include the use of high technology special equipment. Augmentative and Alternative Communication (AAC) systems [1], [2] address individuals, with speech or motor impairments, cognitive limitations, learning disabilities and aging. Computer-based Communication Aids are software and hardware systems based on standard computers, may incorporate special input/output devices, and can focus to specific user profiles. These systems must be very flexible and adaptable in order to cover the variety of the user's special and changing needs.

Designing and developing Interpersonal Communication Aids for people with special needs, is a domain in which advanced software engineering has not been introduced. In the past two decades, some electronic aids were constructed incorporating voice recording and playback capabilities [3], [4], [5], [6], [7], [11] but these devices could not provide a vocabulary consisting of more than 30 words or phrases. Recently, a number of computer-based communication aids have been developed [8], [9], [10], [11] consisting of monolithic and rather difficult to modify software applications for people with speech impairments or other communication problems. COMSPEC [29] and ATIC [2], [27], [28] component-based approaches have made significant steps towards the new generation of Communication Aids. However COMSPEC was an AAC application generator that imposed many limitations on the user interface and the configuration possibilities of resulting communicators. Additionally after the development of a communication aid using the COMSPEC framework adding or removing features from it was very difficult. ATIC was introduced recently as a proprietary component based development environment for communication aids. Detailed comparison of ATIC with ULYSSES approach will be given in the following sections.

Communication Aids must provide direct, fast, and easy access to vital messages for expressing everyday physical needs and to a sufficient vocabulary for everyday communication requirements [12]. Modern human-computer interfaces are still using slow channels of communication, making the use of computers harder for disabled people. This is a basic technical problem that urged the development of fast and easy to use man-machine interaction techniques, efficient enough to allow disabled persons to communicate through the computer with their

Assets 2002, July 8-10, 2002 Edinburgh, Scotland

environment and the society. But a bigger problem that disabled users face with existing computer-based communicators, is the high prices, lack of localization, nonsupport of configurable and adaptable user interface and vocabulary, impossibility of adding or removing special features as needed, and various other technical shortcomings. In this paper, we propose the use of a component-based [17] framework for the design and development of software applications, which we call ULYSSES as a solution to the above problems.

In the next section details of ULYSSES framework are presented. The resulting new AAC product life cycle in then described. In the next section the implementations of specific components are given, followed by three complete Communication Aids.

COMPONENT-BASED FRAMEWORK

Modern software engineering can give several technical answers to problems needing software solutions, including: turnkey solutions, component-based framework solutions, and custom development solutions. Turnkey solutions are ready to go "out of the box", often requiring much configuration work and addressing specific and restricted areas of user needs. Custom development solutions typically involve building most of the major system functionality from scratch. This results to very expensive products to buy and to maintain. Whereas component-based framework solutions elevate the abstractions software engineers use to identify, discuss, solve and implement systems. Component-based frameworks are partial implementations, specifying the nature and way to extend the framework with pluggable components.

According to Hopkins [13] "a software component is a physical packaging of executable software with a well-defined and published interface". D'Souza and Wills [14] define component as "A coherent package of software artifacts that can be independently developed and delivered as a unit and that can be composed, unchanged with other components to build something larger". Similarly, Szyperski [15] provides the following definition: "A software component is a unit of composition with contractually specified interfaces and explicit content dependencies only. A software component can be deployed independently and is subject to composition by third parties". Other authors form similar definitions, with the focus on the interface and physical packaging. Previously, the definition of component was broad to include virtually any artifact of the software development process, such as documents, tests, source code, and parameter files. The definitions given here focus on the deployment aspect of components and their ability to be combined to form larger systems. In particular, the emphasis on well-defined interfaces, separate from their implementation, is critical to the success of components in loosely coupled systems.

Booch [16] defines a component as "a physical and replaceable part of a system that conforms to and provides realization of a set of interfaces". Booch further defines a framework as: "An architectural pattern that provides an extensible template for applications within a domain". D' Souza [14] puts these concepts together and describes component-based frameworks this way: "In general, a component-based framework is a collaboration in which all the components are specified with type models; some of them may come with their own implementations. To use the framework you plug in components that fulfill the specifications".

We introduced the ULYSSES framework for the AAC domain, building on Object Oriented (OO) technologies [21], creating a better case for potential reuse and abstraction. In ULYSSES, two main user groups can be identified: the AAC manufacturers or software *developers*, and the AAC system's *integrators* or Communication Aids resellers. Although, both these user groups will be aware of the framework's basic characteristics, each one must know in detail different aspects of ULYSSES. Developers (including software companies, individual programmers, and broader rehabilitation products companies) will concentrate on the technical aspects of the framework regarding the software engineering techniques, programmatic interfaces, and guidelines. On the other hand integrators (including resellers, rehabilitation consulting centers, specialized therapists) have to focus on the proposed product life cycle, integration methods, and administrative tools for installing, configuring, and maintaining the applications.

The ULYSSES framework provides a specific communication protocol between software components. This protocol is open and easily modified according to the application's needs for data exchange between its components. The protocol consists of a set of interfaces used as a common channel of propagating strings of data, usually characters, words, sentences or complete messages that the user wants to communicate. Components that need access to data (Write or Read) under ULYSSES just have to implement these interfaces and Publish (write) or Subscribe (read) on these "communication channels". Furthermore, ULYSSES provides an AAC component module with the infrastructure, guidelines and tools to build software components, which can operate effectively in the selected computer platform and interact with each other transparently, without even being aware of each other's existence. As a result an integrator can easily assemble and manage a specific Communication Aid from various independently developed components, which will spontaneously cooperate to provide the communicator's functionality and user interface.

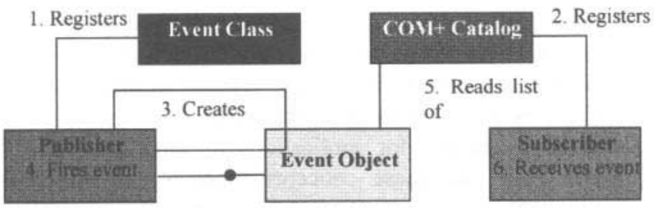

Figure 1: COM+ Event Service Architecture.

ULYSSES proposes the use of a combination of the following specifications, models and services for software component development in the domain of Interpersonal Communication:

a) *Application Specification for Microsoft Windows 2000 for Desktop Applications* [18]. Microsoft Windows 2000 was chosen as the Operating System (OS) on which ULYSSES-based Communication Aids run, because of the advanced accessibility options it provides, the user-friendliness, the increased system stability, and the incorporated technological infrastructure, which we could use to make the resulting communicators more flexible and efficient. Producing cheap and easy to install Communication Aids was one of our goals, and the large installed basis and availability of this OS would help achieving it. Furthermore, users would not need to buy a new computer platform or have to install an exotic or hard to use OS to use our systems.
b) *Component Object Model (COM) specification* [19] from Microsoft was chosen rather than Common Object Request Broker Architecture (CORBA) defined by the Object Management Group (OMG) or Enterprise Java Beans (EJB) from Sun Microsystems, firstly because it was self-contained in the OS and did not need any additional infrastructure or libraries to be installed. Also, the wide spread of the OS and the object model to both final users and developers was considered to be very important. Additionally, most software developers have experience in programming applications using this model and its services or OS-embodied libraries.
c) *COM's Extension for Component Services (COM+)* [20]. The framework makes extensive use of COM+ Events and the corresponding model, which is an evolution of the client-server model (Fig. 1).

Microsoft's Component Object Model (COM+) was chosen as the basis of our architecture for several reasons. We anticipate the spread of COM+ as a standard very soon, covering the majority of software developers. One of the most important services that COM+ provides in MS Windows 2000 is the Component Management Console, which is a powerful tool for managing and maintaining COM+ applications. ULYSSES suggests the use of this console by system integrators, especially when it comes to synchronisation and configuration communication issues between the components of a Communication Aid.

ULYSSES guides developers of AAC components to create their software modules as Publishers or Subscribers to data provided through the COM+ Event system [22]. Additionally it provides a set of Event Classes compatible with COM+ Event Service, which implements a communication interface between software components. A developer who uses these Event Classes following straightforward guidelines, can create either functionality or user interface components, which can interoperate with other ULYSSES-compliant components, even coming from different developers, when integrated into the same application. Furthermore, ULYSSES provides ready to use software components to AAC developers for testing the communication of their components with the Event system and other components. These test or "template" components serve as Publishers, Subscribers or both Publishers and Subscribers of data, and can simulate an application environment for verifying the correct operation of the component being tested.

For integrators of Interpersonal Communication Aids, the framework also offers a detailed user guide and an installation program for assembling and maintaining a complete aid, as well as a World Wide Web information center with a catalog of ULYSSES-compatible components to choose from. Finally, a large database of symbolic and natural communication languages is available, and ready to be connected with ULYSSES components to provide all necessary items (translations, symbols, pictures, video) needed for interpersonal communication applications.

ULYSSES framework is intrinsically Internet-ready. This means that it provides all the necessary infrastructure and support for components that implement remote synchronous (chat) or asynchronous (e-mail) interpersonal communication capabilities using available Internet technologies and data transfer protocols [23], [24].

PRODUCT LIFE CYCLE

Apart from software engineering issues addressed by ULYSSES framework and technological innovations implemented during its application, we proposed a new product life cycle (Fig. 2).

Traditionally, software application developers in the domain of communication aids were creating stand-alone, monolithic applications based on their studies of the user needs and market researches. Retailers who were selling these products did not get much involved in the development process or in the configuration and adaptation process of communication aids according to user requirements. The only possible feedback in the product's life cycle was between the end user and the developer and that feedback was difficult to propagate. Furthermore, the Assistive Technology market was poor due to the lack of software reuse leading many manufacturers to develop the same application functionalities and features from scratch. Each product's life cycle was isolated all the way from the

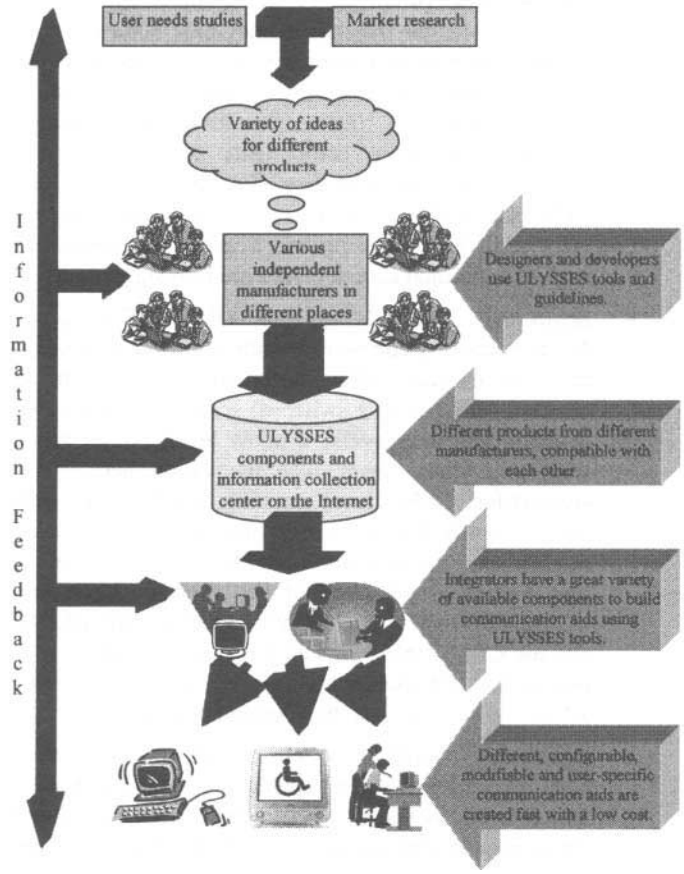

Figure 2: Proposed product life cycle.

original idea to the end user. Finally, to find the right products for specific user needs was a difficult task due to the spread of information and selling points.

The ATIC architecture of the TIDE-ACCESS project [2], [27], [28], proposed a different life cycle, which solved some of these problems and introduced an extended role for communication aids resellers. They were considered as an important user group – target of the architecture – and part of the life cycle of the developed products, having the task to assemble the whole AT system from available software components and suitable I/O devices and techniques. In the context of the ATIC architecture an object-oriented and component-based approach was used also, making use of the MS Windows 3.1 operating system object architecture and infrastructure.

ATIC can be considered as the previous generation in the domain of ULYSSES framework. The difference from the ULYSSES approach was that ATIC used a proprietary Message Manager and a complex communication protocol between components, making the conformance with the specific architecture difficult for the developers. On the other hand ULYSSES uses widely accepted and used operating system infrastructure and messaging system, and a simpler object model, making its guidelines and specifications straightforward to follow. Furthermore ULYSSES introduced the important role of the Internet as a widely accessible medium for gathering and propagating information about the framework and also about the available ULYSSES-compliant software components and I/O devices. The role of this component "bank" in ATIC was assigned to the selling points, the stores that were specialized in Assistive Technology products. ULYSSES provided the possibility to replace the stores with a specialized Internet site offering a higher degree of variety and flexibility. Finally, the new generation framework integrated the specific architecture with the state-of-the-art technology taking advantage of the great new possibilities and features offered by MS Windows 2000 operating system and the accompanying technological infrastructure.

IMPLEMENTATIONS

Several complete Interpersonal Communication Aids systems were implemented for real users based on the ULYSSES framework approach and guidelines and tools of the framework while making extensive use of component reuse. User interfaces addressing both disabled and able-bodied users, like switch interfaces combined with scanning techniques, touch screens combined with vocabulary selection sets, speech synthesis and typical windows based interfaces were used. Multiple combinations of components implementing various functionalities and user interfaces for Interpersonal Communication applications, revealed the power and flexibility of the ULYSSES framework. Computer Mediated Interpersonal Communication (CMIC) applications under ULYSSES may incorporate various kinds of components [26]: a) with their own visible GUI, like an on-screen keyboard or a message editor, b) with a speech user interface (SUI), like the output of a text-to-speech system, c) with a combination of GUI/SUI, and d) without any particular UI, like a natural language syntactic parser, which operates on the background correcting grammatically and syntactically the final output.

We have already developed the following ULYSSES-compliant, interoperating components for use in Interpersonal Communication Applications:

- **On-Screen Keyboards**. Displayed in a separate window on user's screen they are configurable to alphabetic, QWERTY or sorted by most frequently accessed letters layouts. For accessibility, either scanning techniques or alternative input devices are supported. They can also include numbers, punctuation marks, operators etc. as needed. Final users, their helpers or system integrators can modify and configure the size and colors of the window and all buttons and fonts, as they like. Pressing any button causes the component to send the corresponding character to the appropriate interface and make it available to all other components.

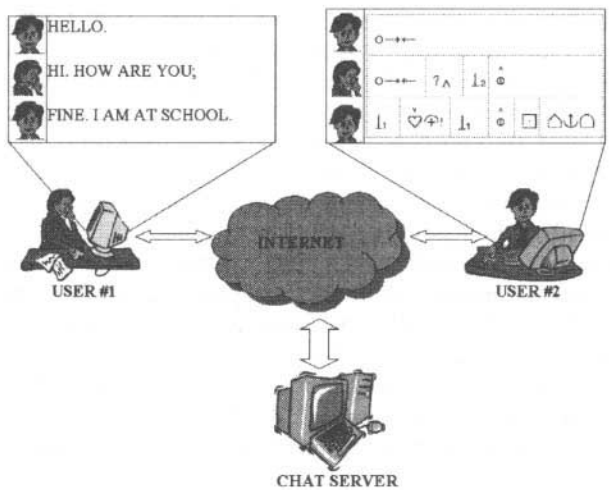

Figure 3: Chat component functionality

- **Word Selection Sets**. Rectangular selection sets of configurable dimensions (vertical bar, horizontal bar, 3x2, 6x4, 8x8, 8x10) and vocabulary. For accessibility, either scanning techniques or alternative input devices are supported. Final users, their helpers or system integrators can modify and configure the size and colors of the window and all buttons and fonts, as well as the contents (words or phrases) of the buttons, as they like. Using a separate vertical or horizontal bar with buttons they can also toggle between many vocabulary sets. All natural languages (having a Unicode alphabet) are supported. Pressing any button causes the component to send the corresponding word or phrase to the appropriate interface and make it available to all other components.
- **Symbol Selection Sets**. They resemble with the Word Selection sets described above, but on each button the symbol of a meaning of a graphic communication system is displayed. Any kind of jpeg, gif, bmp, or tiff image can be displayed on buttons. Sizing of the symbols is automatic according to the size of buttons and button size is automatically configured according to the window size. All symbolic communication systems (like BLISS, PCS, MAKATON, OACKLAND, REBUS, LEXIGRAMS, PIC, SIGSYM, PICSYM, and personal language) are supported. Pressing any symbol-button causes the component to send the corresponding lexical meaning (in the selected natural language) to the appropriate interface and make it available to all other components.
- **Symbol or Text Editors.** A simple and easy to use editor displays in a separate window the messages that users compose. Either it displays symbol or text messages, it provides a simple button interface to correct the displayed message deleting wrong characters or symbols or completely clear the display area and start composing a new message. The same interface also offers scrolling capabilities, when the message is bigger than the display area. Other buttons on the component's interface are used to print the message, send it to a Speech Synthesis component, or to the ULYSSES inter-component communication interface for other components to process. The component's User Interface (UI) as with all other ULYSSES-compliant UIs is configurable in whatever color and size the user prefers. Scanning techniques, as well as traditional access modes are also supported.
- **Scanning.** The UIs of all ULYSSES-compliant components are compatible with scanning techniques that this module offers. Automatic (one switch) or directed (three or five switches) scanning is supported in three scanning levels: window level, button group level and button level. The innovation here is the support for window level scanning, where components not previously known to each other or to the scanning module can be synchronized and selected by a disabled user with a single click of any switch. Other non-ULYSSES windows that may be open are not affected and the COM+ infrastructure gave us the possibility to realize this. Of course, scanning speed is configurable as well as highlight and background colors.
- **Syntactic parser.** When using Symbolic Communication Systems, like the ones aforementioned, the resulting messages are usually lacking the correct syntactic and grammatical formation. Many of these systems don't even have syntax and grammar similar to natural languages. Furthermore, many cognitive impaired users, or users with learning problems are not able to from correct sentences. We have developed a novel technique for expanding spontaneous telegraphic input to well-formed sentences, by adopting a feature-based surface realization for Natural Language generation [25]. This component repairs grammatically and syntactically ill sentences while maintaining the semantics. It does not have any UI, so it runs on the background completely transparently for the user.
- **Speech Synthesis.** For speech synthesis the DEMOSTHeNES modular and scalable speech composer been developed. DEMOSTHeNES' architecture constitutes an extension to current Text-to-Speech systems' structure that enables an open set of module-defined functions to interact with the under processing text at any stage of the text-to-speech conversion. Details on its implementation are given in [30], [31], [32].
- **Chat.** This component offers synchronous remote communication possibilities, using either symbols or natural language, to users who were not able to "chat" on the Internet before. Using standard Internet technology for data transfer and disengaging users from the burden of complicated configuration screens and difficult to access User Interfaces, the Chat

component uses simple button-based interaction to carry out online conversations. Users who are part of these conversations may not even communicate in the same manner (for example one user may use BLISS and the other may use natural English language) due to the database-driven real time translation module that is connected to this component (Fig. 3). The chat mode is activated with a simple button click, the Internet access procedure is automated and the user just picks the person wants to chat with, clicking on a set of his favorite friends photographs or just clicking on a name.

- **E-mail.** This component has the same operation principles with the Chat component, with the difference that it realizes asynchronous remote communication with the well-known e-mail functionality.

By assembling complete applications making combinations of all kinds of the aforementioned components, we can come up with a wide range of products addressing various user needs and communication requirements. We present three typical configurations of our communicators' series, addressing different users groups:

1. Natural Language Communicator

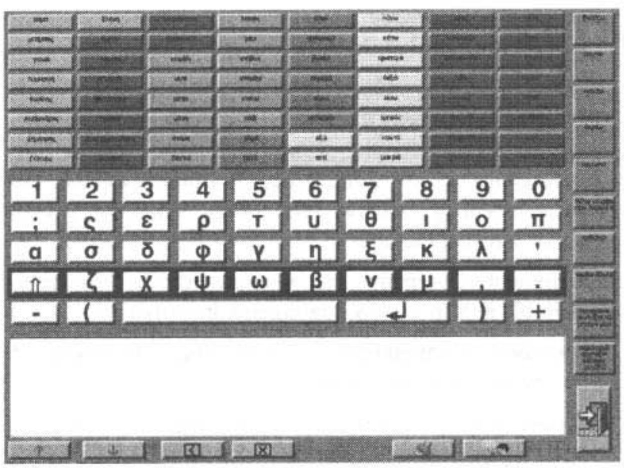

Figure 4: Natural Language Communicator.

Users of this system (Fig. 4) are disabled individuals who have severe motor impairments and cannot use typical computer access devices, like mice or keyboards. Users can understand, read and write a natural spoken language but they may not be able to speak or write in the traditional manner. This Communication Aid runs on a standard computer (Minimum required: Pentium III 600, 128 MB RAM memory, 12,1'' TFT Display, PS/2 Keyboard port) running MS Windows 2000 Professional, and connected through the PS/2 Keyboard Port to a Switch Interface with a single switch. The switch can either be a typical (like Big Red or Jelly Bean) switch or any other device or sensor that can produce a single input signal (motion sensors, eye blink sensors, sip and puff switches etc.). Components used in this configuration are:

- On-Screen Keyboard
- 8x8 Multiple Word Selection Set
- 1x10 Frequently Used Phrases Selection Set (Direct Speech)
- Text Editor
- Speech Composer
- Automatic Scanning

2. BLISS Communicator

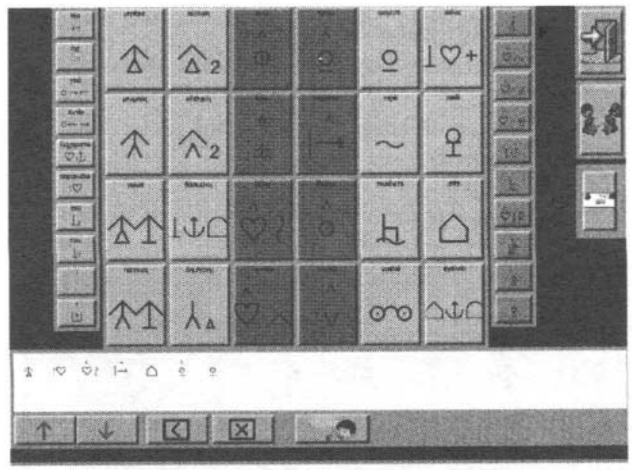

Figure 5: BLISS Communicator.

Typical users of this system (Fig. 5) are individuals with cognitive limitations or learning disabilities. Users may also have motor impairments but they are able to use the TouchScreen to make selections. Users cannot understand, read and write a natural spoken language and may be unable to speak. This Communication Aid runs on a standard computer (Minimum required: Pentium III 600, 128 MB RAM memory, 17'' TouchScreen) running MS Windows 2000 Professional. Components used in this configuration are:

- 6x4 Multiple Symbol Selection Set
- 1x10 Emergency Symbol Selection Sets (Direct Speech)
- 1x10 Dialog Management Symbol Selection Sets (Direct Speech)
- Symbol Editor
- Syntactic Parser
- Speech Composer
- Chat
- E-mail

3. MAKATON Communicator

Users of this system (Fig. 6) are individuals with severe cognitive limitations or learning disabilities. Users may also have severe motor impairments and cannot use typical computer access devices, like mice or keyboards. Users cannot understand, read and write a natural spoken language and may be unable to speak. This Communication Aid runs on a desktop computer (Minimum required: Pentium III 800, 128 MB RAM memory, 17'' Display, PS/2 Keyboard port) running MS Windows 2000 Professional, and connected through the PS/2 Keyboard Port to a Switch Interface with a single switch. The switch can either be a typical (like Big Red or Jelly Bean) switch or any other device or sensor that can produce a single input signal (motion sensors, eye blink sensors, sip and puff switches etc.). Components used in this configuration are:

- 3x2 Multiple Symbol Selection Set
- 1x10 Emergency Symbol Selection Sets (Direct Speech)
- 1x10 Dialog Management Symbol Selection Sets (Direct Speech)
- Symbol Editor
- Speech Composer

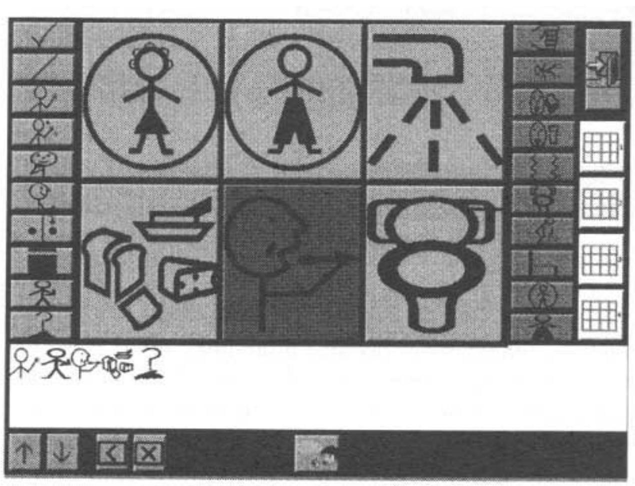

Figure 6: MAKATON Communicator.

CONCLUSIONS

In this paper, we have presented the ULYSSES approach for the development of a new generation of computer-based communication aids. ULYSSES consists of a component-based framework that aims to simplify the integration of appropriately licensed multi-vendor components into low cost AAC products and maximizes modularity and reusability. Following the ULYSSES approach, one can build up powerful and reliable applications, adaptable to various user needs and requirements. The implementation of a number of ULYSSES compliant components has been also presented along with three complete Interpersonal Communication Aids utilizing various accessibility options as well as different input/output devices and interaction elements. Multiple combinations of components implementing various functionalities and user interfaces for interpersonal communication applications only revealed the power and flexibility of the framework. It is expected that the adoption of such an approach by the interpersonal communication industry will cut development costs due to high level of reuse.

Finally, a number of ongoing field trials are in progress in order to evaluate the effectiveness and acceptance of ULYSSES based implementations of communicators with users of cognitive limitations as well as motor and learning disabilities.

ACKNOWLEDGMENTS

Part of the work reported in this paper was carried out within the framework of: a) the M-PIRO project (contract IST-1999-10982), funded by the IST Programme of the European Union and b) the AENEAS project (contract 98AMEA19), funded by the EPET II Programme of the Greek General Secretariat of Research and Technology.

REFERENCES

1. Kouroupetroglou, G., Anagnostopoulos, A., Papakostas, G., Viglas, C., and Haroupias, A. The BLISPHON Alternative Communication System for the Speechless Individuals, in *Proceedings of ESCA Conference for Speech and Language Technology for Disabled Persons* (Stockholm, Sweden, June 1993), 107-110.
2. Kouroupetroglou, G., Viglas, C., Stamatis, C. & Pentaris, F. Towards the Next Generation of Computer-based Interpersonal Communication Aids in *Proceedings of the 4th European Conference for the Advancement of Assistive Technology, AAATE97* (Porto Carras, Greece, October 1997), IOS Press, 110-114.
3. Attainment Company Inc Communication Products. Available at http://www.attainmentcompany.com
4. Crestwood Communication Aids Inc. Available at http://www.communicationaids.com
5. Ablenet Inc. Communication Products. Available at http://www.ablenetinc.com
6. Access First Inc. Communication Products. Available at http://www.accessfirst.net
7. Mayer Johnson Inc Communication Products. Available at http://www.mayerjohnson.com
8. Assistive Technology Inc. Communicators. Available at http://www.assistivetech.com
9. Dynavox Systems LLC. Communicators. Available at http://www.dynavoxsys.com
10. Enkidu Research Communicators. Available at http://www.enkidu.net
11. Parentke Romich Company Communication Products. Available at http://www.prentrom.com

12. Andona, M., Stefanidis, C., and Kouroupetroglou, G. Access to Lexical Knowledge in Interpersonal Communication Aids. *Augmentative and Alternative Communication 15* (December 1999), 269-279.

13. Hopkins, J. Component Primer. *Communications of the ACM 43,* 10 (October 2000), 27-30.

14. D' Souza, D. and Wills, A.C. *Objects, Components and Frameworks with UML: The Catalysis Aproach.* Addison Wesley, Reading, MA, 1999.

15. Szyperski, C. *Component Software. Beyond Object-Oriented Programming.* Addison Wesley Longman Ltd, 1998.

16. Booch, G. *The Unified Modeling Language User Guide.* Addison Wesley, Reading, MA, 1998.

17. Brown, J. *Component-based software engineering.* IEEE Computer Society Press. 1996.

18. *Application Specification for Windows 2000 for Desktop Applications.* MSDN Library. Available at http://msdn.microsoft.com/library/specs/w2kcli.htm

19. *Component Object Model (COM).* Microsoft Platform SDK. Available at http://msdn.microsoft.com/library/en-us/com/comportal_3qn9.asp

20. *COM+ (Component Services).* Microsoft Platform SDK. Available at http://msdn.microsoft.com/library/en-us/cossdk/htm/complusportal_9o9x.asp

21. Kirtland, M. Object-Oriented Software Development Made Simple with COM+ Runtime Services. *Microsoft System Journal,* November 1997.

22. Platt, D. The COM+ Event Service Eases the Pain of Publishing and Subscribing to Data. *Microsoft Systems Journal,* September 1999.

23. Viglas, C., Stamatis, C., and Kouroupetroglou, G. Remote assistive interpersonal communication exploiting component based development, in *Proceedings of the XV IFIP World Computer Congress* (Vienna, Austria, 1998), 487-496.

24. Kouroupetroglou, G., Viglas, C., and Metaxaki-Kossionides, C. Web Enabled Teaching Aid for Non-orthographic Languages, in *Proceedings of the 11th International Conference Society for Information Technology & Teacher Education, SITE 2000* (San Diego, CA, February 2000). 1962-1967.

25. Karberis, G. and Kouroupetroglou, G. Transforming Spontaneous Telegraphic Language to well-formed Greek Sentences for Alternative and Augmentative Communication, in *Proceeding of SETN-02 Conference on Artificial Intelligence* (Thessaloniki, Greece, April 2002) to appear in *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2002.

26. Kouroupetroglou, G., Pino, A., and Viglas, C., Managing Accessible User Interfaces of Multi-Vendor Components under the ULYSSES Framework for Interpersonal Communication Applications, in *Proceedings of Universal Access in Human-Computer Interaction (UAHCI),* (New Orleans, LA, August 2001), 185-189.

27. Kouroupetroglou, G., Viglas, C., Anagnostopoulos, A., Stamatis, C. & Pentaris, F. A Novel Software Architecture for Computer-based Interpersonal Communication Aids, in Klaus, J., Auff, E., Kresmer, W. & Zagler, W. (Eds) *Proceedings of ICCHP'96* (Linz, Austria, July 1996), 715-720.

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

29. Tyvard, S., Morton, L., Stegavik, H., and Stav, E. *Technical Specification of Comspec* (Release 1), 1995.

30. Xydas, G., and Kouroupetroglou, G. The DEMOSTHeNES Speech Composer, in *Proceedings of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis* (Perthshire, Scotland, September 2001).

31. Xydas, G., and Kouroupetroglou, G. Text-to-Speech Scripting Interface for Appropriate Vocalisation of e-Texts, in *Proceedings of EUROSPEECH 2001* (Aalborg, Denmark, September 2001), 2247-2250.

32. Xydas, G., and Kouroupetroglou, G. Augmented Auditory Representation of e-Texts for Text-to-Speech Systems in V. Matousek et al (Eds), *Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence* 2166, Springer-Verlag, 2001, 134-141.