# ULYSSES: A Framework for Incorporating Multi-Vendor Components in Interpersonal Communication Applications

Georgios Kouroupetroglou and Alexandros Pino
*University of Athens, Department of Informatics,*
*Panepistimioupolis, Ilissia, GR-15784, Athens, Greece*
*e-mail: koupe@di.uoa.gr*

**Abstract:** In this paper the ULYSSES framework for designing and developing Interpersonal Communication Aids is presented. It consists of a component-based architecture that aims to simplify the integration of multi-vendor components in low cost applications. For component developers ULYSSES provides an engineering-for-reuse environment with guidelines and tools to build software components, which can operate effectively and interact with each other transparently, without even being aware of each other's existence. Components with different functionality, developed by independent manufacturers can be deployed using various programming languages. Furthermore, ULYSSES grants a process of engineering-with-reuse for system integrators to take advantage of the reusable assets produced during engineering-for-reuse. Thus, selection and assembly of components on demand for building user-specific robust products out of pre-fabricated software parts is becoming an easy task for system integrators.

## 1. Introduction

Designing and developing software based Interpersonal Communication Aids that address a wide range of users is a challenge for the computer industry [7]. Traditionally, each application has its own user interface and specific functionality, and in the worst case is built in a monolithic manner [1]. Moreover, it is difficult for a single manufacturer to incorporate the technology and resources needed to build an application, which integrates adaptive and accessible user interfaces [4], [8] and multiple functionalities for all the potential users. Should this task be achieved, the result is usually an expensive and complex product, which end users can't afford to buy or learn to use and configure.

This problem is partially solved by using object-oriented [11] and component-based [9] [10] methodologies to come up with modular applications, which can be integrated, incorporating software modules with such functionality and user interface that addresses fully their users' needs [3]. In this sense, the end user buys just the parts of the application he/she needs, and deals with features and functionality he/she knows about.

In the light of the above, this paper presents a software framework, named ULYSSES, that goes one step further. Based on the aforementioned philosophy and technology it offers an infrastructure for integrating software components and their user interfaces, with various accessibility options, developed by different manufacturers, in a single robust application. Next section presents the development of the ULYSSES framework. Furthermore, a novel AT product life cycle is presented, as a result of new possibilities that arose by the use of the aforementioned software engineering architecture. Finally, present results are discussed in comparison with the ATIC architecture for Interpersonal Communication Aids [2] [3] [6] [7].

## 2. Framework Development

In ULYSSES, two main user groups can be identified: the manufacturers or software developers, and the system's integrators or resellers. Although, both these two user groups will be aware of the framework's basic characteristics, each one must know in detail different aspects of ULYSSES. Developers must concentrate on the technical aspects of the framework regarding the software engineering techniques, while integrators must focus on the proposed application life cycle and administrative tools.

The ULYSSES framework provides a specific communication protocol between software components [16]. This protocol is open and easily modified according to the application's needs for data exchange between its components. Furthermore, ULYSSES provides developers with 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 a system integrator can easily assemble and manage a specific application from various components, which will spontaneously cooperate to provide the application's functionality and user interface.

ULYSSES proposes the use of a combination of the following specifications for software component development:

a) Application Specification for MS-Windows 2000 for Desktop Applications [12].
b) Component Object Model (COM) specification [13] [17].
c) COM's extension for Component Services (COM+) [14].

The framework makes extensive use of COM+ Events and the corresponding model, which is an evolution of the client-server model (Fig. 1). ULYSSES guides developers to create their components as Publishers or Subscribers to data of the COM+ Event system [15] [16]. Additionally it provides a set of Event Classes compatible with COM+ Event Service, which implement 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 developers for testing purposes. These 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.

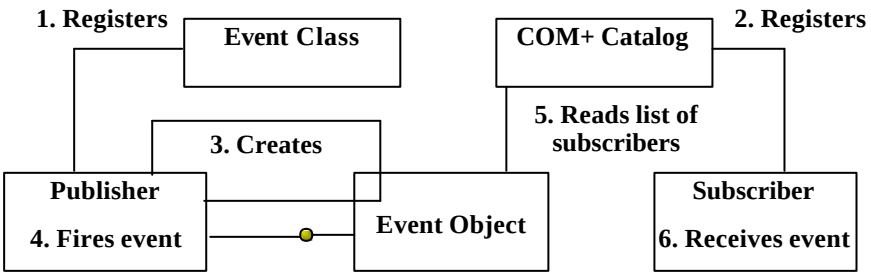


Figure 1: COM+ Event Service Architecture

To application integrators, the framework offers a detailed user guide and an installation program for assembling and maintaining a complete application as well as a World Wide

Web information center with a catalog of ULYSSES-compatible components to choose from.

ULYSSES framework was applied on Interpersonal Communication Aids. Several complete systems were implemented based on the philosophy, guidelines and tools of the framework while making extensive use of component reusability [18]. User interfaces addressing both disabled and able-bodied users, like speech synthesis, switch interfaces combined with scanning techniques, touch screens combined with vocabulary selection sets [5] 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.

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

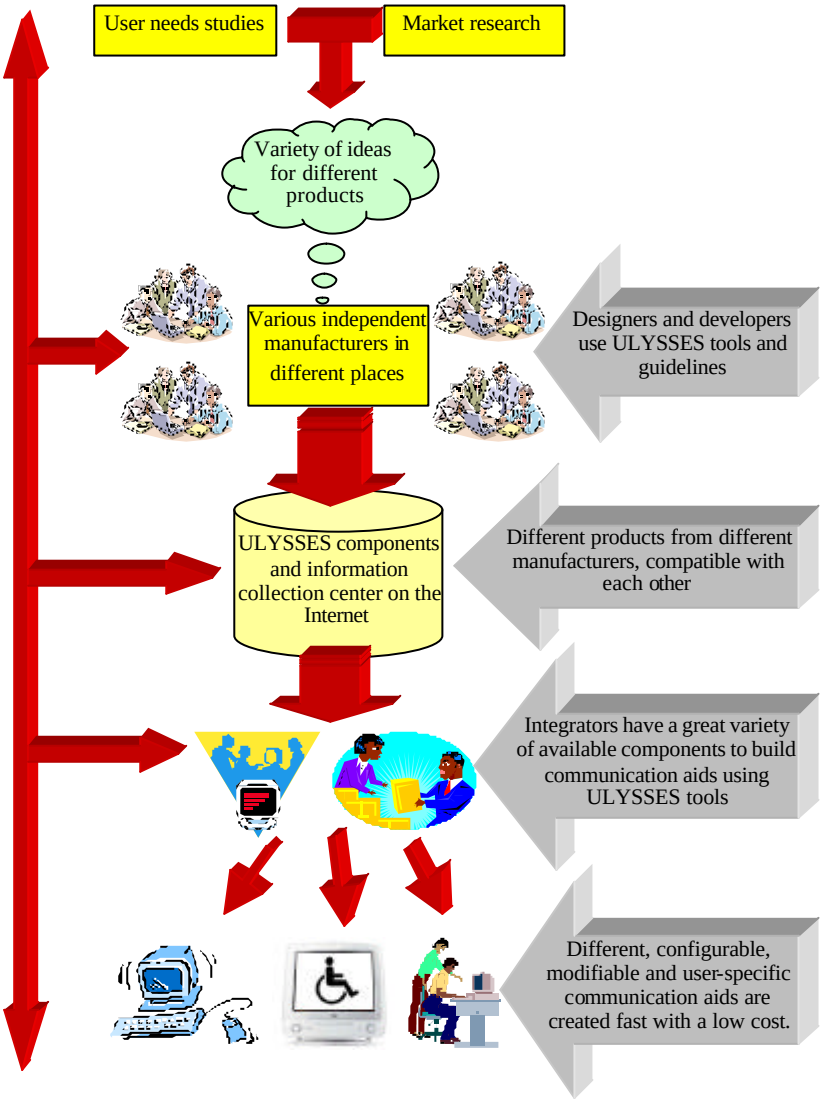


Figure 2: ULYSSES approach for the life cycle of Interpersonal Communication Aids.

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, Assistive Technology market was poor due to the lack of software reuse leading many manufacturers to develop the same application functionality and features from scratch. Each product's life cycle was isolated all the way from the 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.

## 3. Discussion

The ATIC architecture of the TIDE-ACCESS project [3] 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 our 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 of 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 selves 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 Windows 2000 operating system and the accompanying technological infrastructure.

## 4. Conclusions

The ULYSSES component-based framework for designing and developing Interpersonal Communication Aids was presented for both developers and system integrators. A novel AT product life cycle was also presented and discussed in comparison with the ATIC architecture [3]. 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.

## Acknowledgements

Part of the work reported in this paper was carried out within the framework of the AENEAS project (Contract No: 98AMEA19), partially funded by the EPET II Programme of the Greek General Secretariat of Research and Technology.

## References


[1] G.Kouroupetroglou, A.Anagnostopoulos, G.Papakostas, C.Viglas and A.Haroupias, The BLISPHON Alternative Communication System for the Speechless Individual, *Proc. of ESCA Conf. Speech and Language Technology for Disabled Persons*, Stockholm, May 31-June 2, 1993, pp. 107-110.

[2] C. Viglas, C. Stamatis and G. Kouroupetroglou, Remote assistive interpersonal communication exploiting component based development, *Proceedings of the XV IFIP World Computer Congress*, Vienna, 1998, pp.487-496.

[3] G. Kouroupetroglou, C. Viglas, C. Stamatis and F. Pentaris, Towards the next generation of computer-based interpersonal communication aids, *Proceedings of AATE 97, Porto Carras, Greece*, 1997, pp. 110-114.

[4] G. Kouroupetroglou, A. Pino and C. Viglas, Managing Accessible User Interfaces of Multi-Vendor Components under the ULYSSES Framework for Interpersonal Communication Applications, *Proceedings of Universal Access in Human-Computer Interaction (UAHCI)*, New Orleans, Louisiana, 5-10 August 2001.

[5] M.Andona, C.Stefanidis and G.Kouroupetroglou, Access to Lexical Knowledge in Interpersonal Communication Aids, *Journal of Augmentive and Alternative Communication*, 15 (1999) 269-279.

[6] G.Kouroupetroglou, C.Viglas, A.Anagnostopoulos, C.Stamatis and F.Pentaris, A Novel Software Architecture for Computer-based Interpersonal Communication Aids, in J.Klaus, E.Auff, W.Kresmer and W.Zagler (eds) *Interdisciplinary Aspects on Computers Helping People with Special Needs*, ISBN3-486-23797-7, Oldenbourg, Munhen and Wien, Proc. of ICCHP'96 - 5th International Conference on Computers Helping People with Special Needs", July 17-19, Linz, pp. 715-720..

[7] P.-L. Emiliani, J.Ekberg, G.Kouroupetroglou, H.Petrie and C.Stefanidis, Development Platform for Unified Access to Enabling Environments, in J.Klaus, E.Auff, W.Kresmer and W.Zagler (eds) *Interdisciplinary Aspects on Computers Helping People with Special Needs*, ISBN3-486-23797-7, Oldenbourg, Munhen and Wien, Proc. of ICCHP'96 - 5th International Conference on Computers Helping People with Special Needs", July 17-19, Linz, pp. 69-75.

[8] C. Stephanidis (ed) User interfaces for all, Lawrence Erlbaum Ass., ISBN 0-8058-2967-9, 2001.

[9] J. Brown, Component-based software engineering, IEEE Computer Society Press, ISBN: 0-8186-7718-X, 1996.

[10] D. Platt, The COM+ Event Service Eases the Pain of Publishing and Subscribing to Data, *Microsoft Systems Journal*, Sep.1999: http://msdn.microsoft.com/library/periodic/period99/com+event.htm

[11] M. Kirtland, Object-Oriented Software Development Made Simple with COM+ Runtime Services, *Microsoft System Journal*, November 1997: http://www.microsoft.com/msj/1197/complus.htm

[12] MSDN Library/Specifications/Application Specification for Windows 2000 for Desktop Applications http://msdn.microsoft.com/library/default.asp?URL=/library/specs/w2kcli.htm

[13] MSDN Library/Specifications/Component Object Model (COM) Specification 0.9, http://msdn.microsoft.com/library/default.asp?URL=/library/specs/w2kcli.htm

[14] MSDN Library/Specifications/Distributed Component Object Model Protocol - DCOM/1.0, http://msdn.microsoft.com/library/default.asp?URL=/library/specs/distributedcomponentobjectmodelprotocoldcom10.htm

[15] MSDN Library/Platform SDK/Component Services/ COM+ (Component Services), http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/cossdk/betaintr_6qan.htm

[16] MSDN Library/Technical Articles/ Component Object Model/COM+/COM+ Technical Series: Loosely Coupled Events, 1999, http://msdn.microsoft.com/library/default.sdp?URL=/library/techart/compluscouple.htm

[17] S. Williams, C, Kindel, The Component Object Model: A Technical Overview, MSDN Library/Technical Articles/Component Object Model/, 1994, http://msdn.microsoft.com/library/default.asp?URL=/library/techart/msdn_comppr.htm

[18] D. Souza, BJ. Whalen, P, Wilson, Implementing Side-by-Side Component Sharing in Applications (Expanded), MSDN Library/Technical Articles/Windows Platform/Windows 2000/Implementing Side-by-Side Component Sharing in Applications, 1999, http://msdn.microsoft.com/library/default.asp?URL=/library/techart/sidebyside.htm