A Software Environment for the Development of Component-based Augmentative and Alternative Communication Applications

Alexandros Pino[*]

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
pino@di.uoa.gr

Abstract. As an answer to the disabled community's Odyssey to gain access to adaptable, modular, multilingual, cheap and sustainable Augmentative and Alternative Communication (AAC) products, we propose the use of the ITHACA framework. It is a software environment for building component-based AAC applications, grounded on the Design for All principles and a hybrid (community and commercial) Open Source development model. ITHACA addresses the developers, the integrators (e.g., vendors, facilitators, special educators), as well as people who use AAC. We introduce a new AAC product lifecycle, i.e., the design-develop-distribute procedures, from the developers' viewpoint and the search-select-modify-maintain procedures from the integrators' perspective. ITHACA provides programmers with a set of tools and reusable Open Source code for building AAC software components. It also facilitates AAC product integrators to put together sophisticated applications using the available on the Web, independently pre-manufactured, free or commercial software parts. Furthermore, it provides people who use AAC with a variety of compatible AAC software products which incorporate multimodal, user-tailored interfaces that can fulfill their changing needs. Several ready to use ITHACA-based components, including on-screen keyboards, Text-to-Speech, symbol selection sets, e-chatting, e-mailing, and scanning-based input, as well as four complete communication aids addressing different user cases have been developed. This demonstration showed good acceptance of the ITHACA applications and substantial improvement of the end users' communication skills. Developers' experience on working in ITHACA's Open Source projects was also positively evaluated.

Keywords. Augmentative and Alternative Communication, Open Source, Component-Based Development, Design for All, Symbolic Communication Systems

---

[*] Dissertation Advisor: Georgios Kouroupetroglou, Assoc. Professor

1 Introduction

1.1 The Problem

For people with complex communication needs, those with speech and/or motor impairments, cognitive limitations, learning disabilities and aging, daily routine as well as rehabilitation and educational programs often include the use of Augmentative and Alternative Communication (AAC) aids [1]. In the past, AAC was dominated by low-technology or non-electronic devices [2]. A few decades ago, several electronic aids with voice recording and playback capabilities were introduced in the international market. Such non-computer-based products are still widely used. Although these devices are considered very useful for the persons using AAC, they provide a limited vocabulary, need extra effort from the facilitators to add new recordings, and cannot keep up with the nontrivial progress usually achieved by their users [3].

Recently, several computer-based communication aids that support a range of symbolic communication systems, special I/O devices, configurable User Interfaces (UIs), and speech synthesis have been developed by various companies. These devices have larger vocabularies, but they support a very limited number of natural languages and it is rather impossible to add new ones. These modern computer-mediated interpersonal communication systems should be adaptable in order to satisfy the wide variety of the users' changing needs [4] and the specific user profiles. Nevertheless, traditionally, software application developers in the domain of communication aids are creating stand-alone, monolithic applications based on their studies of user needs and market research. Retailers do not get actively involved in the development or in the configuration and adaptation process of communication aids. The only possible feedback in the product life cycle is between the end user and the reseller and that feedback is difficult to propagate to the developer. Furthermore, AT products are very few and expensive due to the small market and the lack of software reuse, as many manufacturers develop the same functionalities and features from scratch again and again. Throughout each product's life cycle, from the original idea to the end user, there is no significant feedback and evaluation. Finally, finding the right product for specific user needs is a difficult task due to the dispersed information and selling points.

Other problems that disabled users face with existing computer-based commercial solutions include: absence of multilingual support, lack of proper support for customization, and difficulty in adding or removing functionalities or components when needed. Moreover, designing and developing interpersonal communication aids for people with disabilities is a domain for which modern software engineering approaches such as those that combine Component-Based Development (CBD) and the Open Source model that lower the development costs have not been applied. DeRuyter et al. [5] refer to the development of Open Source software that runs on mainstream computers, as a better alternative in order to provide maximum flexibility and accessibility. ITHACA framework is addressing these problems by combining the Design for All, Open Source, and CBD approaches.

1.2 Related work

Projects COMSPEC [6] and ACCESS [7], have made significant steps towards CBD. ComLink and ATIC were two component-based approaches produced by these projects as an answer to the problems of the AAC market. Although both these frameworks were characterized "open", which meant that third-party developers could theoretically develop compatible components, they were essentially Closed Source, as their code was not freely available. The AAC industry's response was not encouraging, as no third party components were delivered to enrich the basic component collection that accompanied these two frameworks.

The ULYSSES framework was the product of the subsequent project AENEAS [8-10]. The main difference from its predecessor 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 used a widely available and known infrastructure and messaging system that was embedded in the (Operating System) OS, and a simpler object model, making its guidelines and specifications straightforward. Nevertheless, ULYSSES had the main drawback that the AAC industry needed to be accustomed to its proprietary guidelines and code, in order to comply with the framework. This is very unlikely to happen, especially when the framework is closed-source.

The World Wide Augmentative and Alternative Communication (WWAAC) project has contributed towards the direction of Open Source development, in the domain of Internet accessibility for AAC users [11]. The most important contribution of the WWAAC project was the Concept Coding Framework (CCF) [12]. CCF provides direct support for symbol users on web pages through its open-sourced concept coding infrastructure and protocol. The vision of concept coding is that instead of images and symbols having to be transferred from one computer to another, it should be possible to transmit a unique code designating the meaning of the symbol needing to be transferred. Nevertheless, the WWAAC project did not develop a framework for building applications such as ComLink, ATIC, ULYSSES, and ITHACA, and its resulting applications were not compatible with any of these frameworks.

OATSoft [13] is an Open Source software repository dedicated to Assistive Technology (AT). It currently lists more than 150 Open Source projects and it is the most important initiative in this domain. Project:Possibility is a similar initiative that hosts more than 10 Open Source AT projects. Both initiatives have an active community formed, but neither hosts an Open Source component-based AAC development framework project.

1.3 The ITHACA-based AAC Software Lifecycle

The ATIC architecture [7] proposed a different life cycle that solved some of the aforementioned problems and introduced an extended role for communication aids resellers. They were considered as an important user group (namely the integrators) having an essential part in the life cycle of the developed products, with the task of assembling the whole AT system from available software components and suitable I/O devices and techniques.

ULYSSES introduced the important role of the Internet as a widely accessible medium for gathering and propagating information about the framework and available software components and I/O devices. In ATIC, the stores that were specialized in AT products played the role of the component repository. ULYSSES replaced the traditional stores with a specialized website offering a higher degree of availability, variety and flexibility.

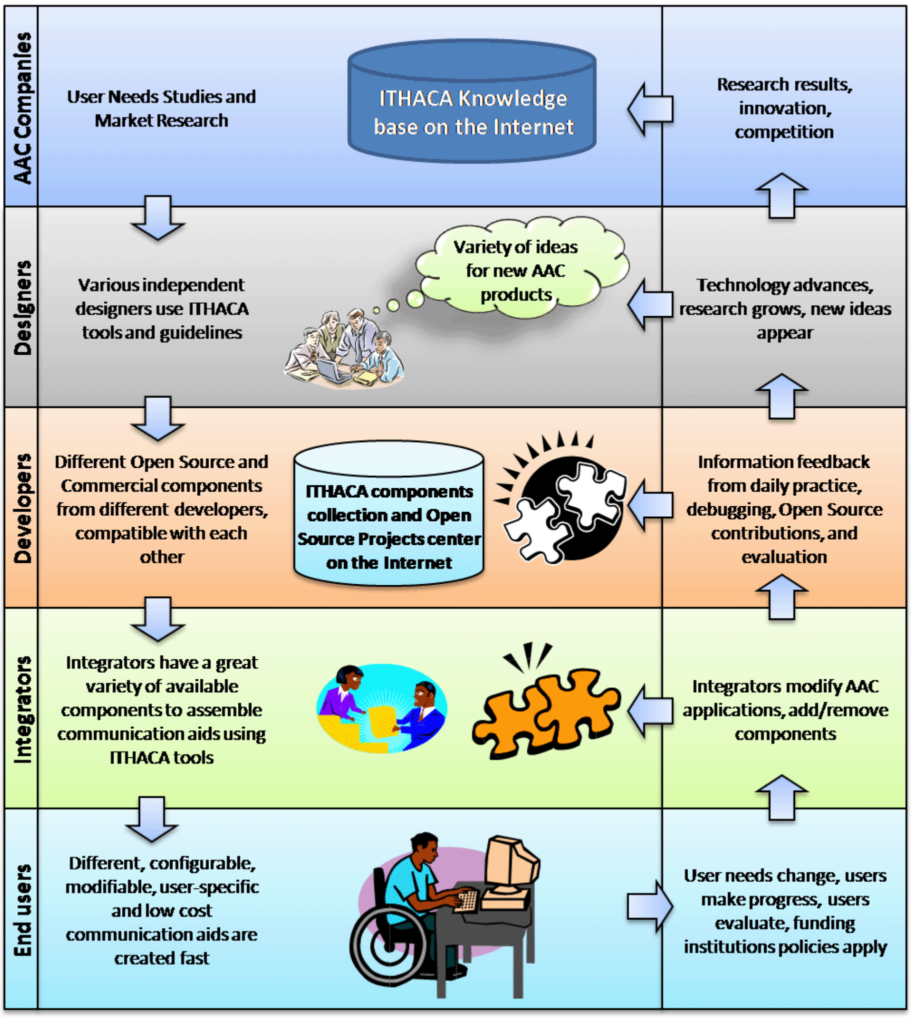

Fig. 1. ITHACA AAC product Life Cycle

With ITHACA, the decision as to which AAC device or software is purchased, how instruction is provided, and how the device is maintained and developed typically involves many individuals, including the person who uses AAC, family members, communication and education professionals, and funding agencies [14]. All stakeholders included in the AAC product lifecycle can participate and contribute to Open Source software development, both on technical and non-technical aspects. Developers concentrate on the technical aspects of the framework regarding the Open Source software engineering techniques, interfaces, and guidelines. On the other hand, integrators focus on the proposed product life cycle, integration methods, and administrative tools for installing, configuring, modifying and maintaining the applications.

ITHACA proposes an extended and upgraded AAC product life cycle by introducing a Web-based AAC knowledge base and the Web-based component inventory (Figure 1). Most importantly, the online ITHACA components may be Open Source (community or commercial) or Closed Source, thus modifying the lifecycle to a new form of hybrid Open Source/Closed Source product lifecycle. A substantial aspect of the new life cycle is the high importance that is given to information propagation between all stakeholders and all stages.

The framework allows for a new viewpoint from the side of AAC production stakeholders. The following procedures are positively affected: *Design, Development,* and *Distribution* of AAC applications. From the end-users' ITHACA also allows for innovative approaches in lifecycle stages: *Search, Selection, Modification,* and *Maintenance* of AAC products.

2 The ITHACA Framework

What mainly sets ITHACA apart from the aforementioned projects is that it proposes a hybrid approach, by combining CBD, a free Open Source framework and core components, along with community and commercial Open Source peripheral components in order to increase competition, while maintaining the interests of software manufacturers for this market. Universal Design or Design for All completes the framework, by offering the important aspects of adaptability and multiple users' needs inclusion [15]. Two main user groups of the ITHACA framework are identified: the AAC manufacturers or software developers, and the AAC systems integrators or communication aids resellers. Although both user groups are aware of the basic characteristics of the framework, each must know different aspects of ITHACA in detail.

2.1 Technological Background

ULYSSES – the framework on which ITHACA is based – proposed the use of a combination of the following specifications, models and services for the development of components and software applications [9]: Application Specification for Microsoft Windows 2000 for Desktop Applications, Component Object Model (COM) specification, and COM's Extension for Component Services (COM+) .

ITHACA proposes an update of ULYSSES specifications and guidelines by replacing Application Specification for Microsoft Windows 2000 for Desktop Applications with the contemporary Microsoft UI Automation for upgraded application accessibility. This Microsoft specification was chosen against its older version Microsoft Active Accessibility and IBM's IAccessible2 as the best practice for designing and developing Windows-based accessible applications. ITHACA also adopted COM and COM+ specification as they were updated and supported by the .NET programming environment.

The PC with Microsoft Windows platform was chosen as the OS on which ITHACA-based communication aids run, because of the advanced accessibility options it provides, the user-friendliness, the increased system stability, and the high support for special I/O devices. Due to the large installed basis and availability of the OS, users will not need to buy a new computer or have to install a non-common or hard to use OS to use ITHACA-based communication aids.

ITHACA framework makes extensive use of COM+ Events [16] and the corresponding model, which is an evolution of the client-server model. COM+ was chosen as the basis of the framework's architecture because its services are widely used and accepted by the developer's community. One of the most important features that COM+ provides is the Component Management Console, which is a powerful tool for managing and maintaining COM+ applications; this tool clears many technical problems and restrictions previous frameworks had.

The ITHACA framework produces .NET applications that use COM+ services, and 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 is based on the consideration that the fundamental data used in AAC are the "Concepts" (an idea widely accepted in the AAC domain). Abstract/logical concepts can engage various data types at the presentation level; that is, concepts may be represented by strings (i.e., words or phrases), video, sound, or icons. A concept that is conveyed from one component to another, locally or remotely, may be processed and may change data type and/or language between components, making its management rather complex.

To simplify the situation, we defined a base language (database of concepts) named "Interlingua", i.e. a pseudo-language based on English, in which all concepts can be represented as character strings. Language-independent components can communicate using Interlingua, while output components or language-aware (natural or symbolic) components use the equivalent representation (in any form) of the Interlingua concept in their own language or symbolic system according to the database's relations. The database of natural and symbolic languages (including Interlingua, photograph, sound or video-based languages) is the heart of the framework and allows simplification of inter-component communication by using strings only. Final users and/or facilitators and educators can add new content to the database, and add or modify concepts of the Interlingua.

The ITHACA concept transmission protocol consists of a set of interfaces used as a common channel for propagating strings of data, i.e., characters, words, sentences or complete messages that the user composes. This communication protocol and interfaces, in cooperation with the ITHACA database and the Interlingua concepts helps to overcome translation and inter-component communication related challenges that AAC component-based framework research has encountered for many years.

2.2 ITHACA for Developers

ITHACA guides developers of AAC components to program their software modules as Publishers or Subscribers to data provided through the COM+ Event Service. A developer, who uses the corresponding Event Classes following straightforward guidelines, can create components incorporating their own GUI, Speech User Interface (SUI) or plain transparent functionality, which can interoperate with other ITHACA-compliant components that may have been created by other developers.

An executable program, also provided with ITHACA, initializes and configures the AAC application when run for the first time, and activates all components and interfaces every subsequent time that it is executed. The framework's executable program, all DLLs, along with their Open Source code and guidelines, can be downloaded from ITHACA's website[1].

Furthermore, ITHACA provides ready-to-use software components to AAC developers for testing the communication of their components with the Event Service and other components. These test or "template" components serve as Publishers, Subscribers or both Publishers and Subscribers of data. Test components are also Open Source offering basic framework code examples, and can create a real application environment in order to verify the correct operation of the component being tested.

Finally, ITHACA framework is intrinsically Internet-ready, i.e., it provides all the necessary infrastructure and support for components that implement remote synchronous (chat) or asynchronous (e-mail) interpersonal communication using available Internet technologies and data transfer protocols [17].

2.3 ITHACA for Integrators

An integrator can easily assemble and manage communication aids from various independently developed components, which cooperate to provide the application functionality and UI. For integrators of interpersonal communication aids, ITHACA offers a detailed user guide and an installation program as well as a World Wide Web information center with a catalog of ITHACA-compatible components (free or commercial) to choose from. Furthermore, the database of symbolic communication systems and natural languages is also available to integrators, and ready to be connected with ITHACA components to provide concept mapping for interpersonal communication applications. For a specific AAC application, only parts of the database are needed and can be downloaded separately.

2.4 ITHACA Components

The most common functionality in AAC products has already been developed using the ULYSSES framework, in the form of pluggable components [9]. All these components are reprogrammed, converted to ITHACA-compliant .NET modules, Open Source and freely available on the ITHACA website. Based on Open Source software

---

1 http://speech.di.uoa.gr/ithaca/

development projects, software companies and hobbyist programmers including special education professionals contributed to the final result. The code reuse was extensive and this allowed for better UI refinement and better interoperability between all modules. The main components we developed are: *Virtual Keyboard, Word Selection Set, Symbol Selection Set, Symbol or Text Editor, Scanning, Syntactic parser* [18], *Text-to-Speech System* [19], *Chat, E-mail, Database/Translation* [17].

3 Framework's Evaluation

For a nine months period, 20 programmers and researchers were involved in the process of designing and developing ten framework core components and eleven ITHACA compliant AAC components. The developers' team consisted of 6 pregraduate, 5 postgraduate and 4 doctoral students of the Information and Telecommunications Department of the University of Athens. Additionally, 2 professional developers (employees of a major software company) also participated in the development team. The project leaders' team consisted of 3 researchers who were previously involved in ATIC and ULYSSES, and currently in ITHACA project. This team also played the frameworks' designers', core components' developers', and integrators' role. Forum and online chatting were used for inter- and intra-team communication. The programming environment was .NET-based, and Visual Basic, C++ and C# programming languages were used.

Table 1. Developers' rating results

Measure	Rating	Indicators	mean	std dev
Functionality	**4.39**	*Suitability*	4.82	0.39
		Accurateness	4.35	0.61
		Interoperability	4.00	0.87
Maintainability	**4.09**	*Required support*	3.24	0.66
		Changeability	4.65	0.49
		Stability	3.65	0.61
		Testability	4.82	0.39
Reliability	**3.73**	*Maturity*	3.59	0.80
		Fault tolerance	3.53	0.80
		Recoverability	4.06	0.83
Portability	**4.35**	*Adaptability*	4.65	0.49
		Installability	3.59	1.18
		Replaceability	4.82	0.39
Usability	**4.31**	*Undestandability*	4.65	0.61
		Learnability	4.71	0.47
		Operability	3.59	1.12

Several models in the literature for evaluating Open Source projects [20], Open Source software quality [21], as well as Open Source repositories success, were considered for the selection of the aspects of ITHACA that were evaluated. A combination of features and measures was used for a subjective evaluation of the quality framework by the 17 participating developers, using an online survey with 5 measured domains and a 1-5 Likert rating scale. The scale was defined as: 1=very bad; 2=bad; 3=moderate; 4=good; and 5=very good. A total number of 16 indicators were rated. The ISO 9126 standard that classifies software quality in a structured set of characteristics was taken as a reference, though some modifications were made to the set of sub-characteristics that the standard suggests in order to better suit an Open Source framework evaluation as opposed to a software application evaluation. Opinions were also asked in the form of open type questions, investigating ITHACA's functionality, maintainability, reliability, portability, and usability. The most important findings are summarized in Table 1.

In order to confirm the breadth of software that can be produced as a proof of concept of the ITHACA framework, we have further conducted a number of demonstrations with real users. Making combinations of the aforementioned components, we assembled a range of customized AAC applications addressing various user needs and communication requirements. We briefly summarize our observations regarding the experiences of the users, their family members, and their teachers. Our goal is not to provide a formal evaluation of the resulting systems, but to provide a sense as to how the systems were received by the individuals for whom they were designed.

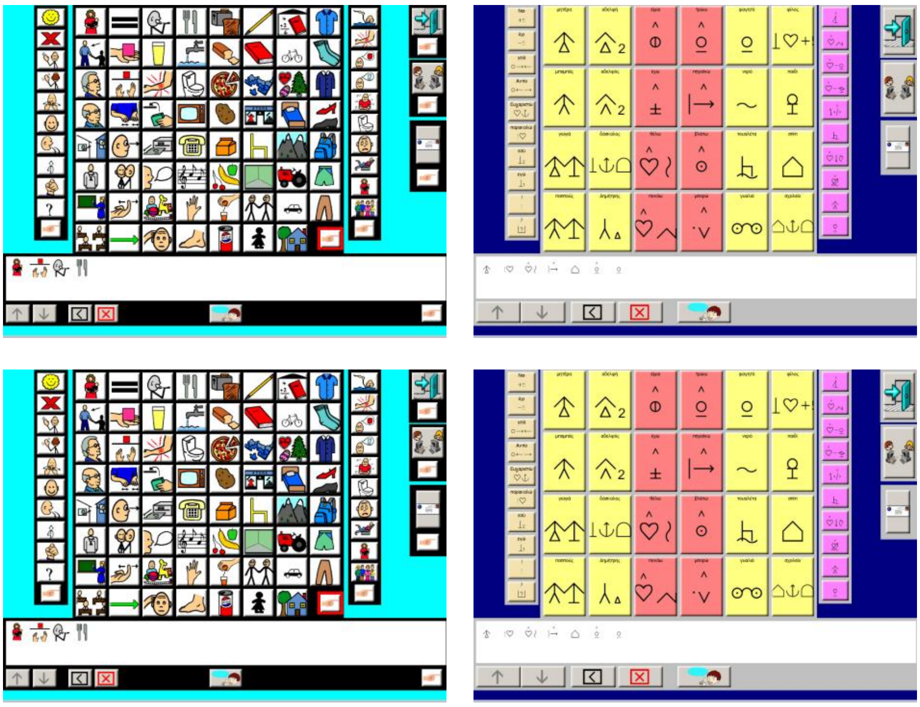

Fig. 2. The four demonstration AAC applications

Four users were selected from a special education and rehabilitation center in Greece; they were all children monitored in their special school environment, and the appropriate communication aids were selected following the main guidelines proposed in [22]. All participants were diagnosed as having cerebral palsy, with different symptoms that ranged from mild to very severe. All users had severe speech, and none of the users had ever used computer-based AAC before. The aim was to help all four users communicate in their Greek-speaking environment, firstly at school and secondly at home. Screenshots of the final GUIs of the four applications are illustrated in Figure 2.

4 CONCLUSIONS

We have presented the integrated ITHACA approach for the development of computer-based AAC applications. ITHACA consists of an Open Source, component-based framework that aims to simplify the integration of multi-vendor components into low cost AAC products and maximizes modularity and reusability [23]. ITHACA suggests technical (CBD), as well as business and management (Open Source, combined free and commercial) models for AAC assistive technology support provision. Following the Design for All approach, developers can build reliable components, adaptable to various user needs and requirements.

The mixed model that includes an Open Source framework, running on a mainstream proprietary OS and a mainstream computer, changes the life cycle of AAC products. It allows for Open Source, Closed Source, free, and commercial components, to compete for a place in the end-users application. Community volunteer work in an Open Source project context could be part of the solution for the high costs of AAC products. This way, the cost of debugging or even modifying software components remains low also, thus making both the development process and the final applications more sustainable [25]. Furthermore, AAC researchers have the opportunity to easily test their novel ideas and technology in an Open Source, component-based integrated environment without having to develop additional infrastructure.

ITHACA is a result of several research projects and the entire life cycle we introduced could be realized only if all stakeholders embraced this framework or a similar one. This could only be achieved with joined efforts by companies, institutes, funding organizations and people who use AAC. An important aspect for the success of this life cycle is the continuous flow of feedback information from all stakeholders and stages of the life cycle in all directions. The modified production and maintenance procedures should be coupled with a business model and a central administration system in order to be complete. A hybrid Open Source/Commercial model of management is proposed, meaning that an Organization (profit or non-profit) should coordinate development and distribution procedures. It is expected that the adoption of such an approach by the AAC industry and community will lead to affordable AAC products and upgrade their quality and variety.

The developers' evaluation showed that the potential contribution of the framework to the AAC domain is considered very important. The extensive code reuse, as well as the good functionality of the framework, was highly appreciated. To reach a critical mass of basic components and functionality and to attract enough interest and more development resources is a concern for all Open Source initiatives, and the success of their dissemination is crucial. The listing of ITHACA in OATSoft and Project:Possibility websites, or SourceForge , the largest hub for Open Source development projects, will be our next step in order to disseminate the framework as an Open Source project.

Multiple combinations of ITHACA-compliant Open Source components, implementing various functionalities and UIs for interpersonal communication applications have been used to assemble four different applications utilizing different accessibility options as well as I/O devices and interaction techniques, revealing the flexibility of the model used. The demonstration of the four systems and real users' and their facilitators' comments showed that interacting computer-based AAC is always a new and interesting way of learning and communicating. Progress was made even after more traditional methods had reached their limits. Involving users and facilitators in the development, testing, debugging, and evolving procedures was considered very valuable and effective.

5 References

1. Beukelman, D.R., Mirenda, P.: Augmentative and Alternative Communication: Management of Severe Communication Disorders in Children and Adults (3rd edition). Paul H. Brookes Publishing Co., Baltimore (2005)
2. Reichle, J., York, J., Sigafoos, J.: Implementing Augmentative and Alternative Communication: Strategies for Learners with Severe Disabilities. Paul H. Brookes Publishing Co., Baltimore (1991)
3. Cumley, G.D., Swanson, S.: Augmentative and Alternative Communication Options for Children with Developmental Apraxia of Speech: Three Case Studies. Augmentative and Alternative Communication 15, 110-125 (1999)
4. Light, J.: Toward a Definition of Communicative Competence for Individuals Using Augmentative and Alternative Communication Systems. Augmentative and Alternative Communication 5, 137–144 (1989)
5. Deruyter, F., McNaughton, D., Caves, K., Bryen, D.N., Williams, M.B.: Enhancing AAC Connections with the World. Augmentative and Alternative Communication 23, 3, 258-270 (2007)
6. Lundälv, M., Hekstra, D., Stav. E.: Comspec - a Java based Development Environment for Communication Aids. In: Placencia Porrero, I., Ballabio, E. (eds.) Improving the Quality of Life for the European Citizen, pp. 203-207. IOS Press, Amsterdam, (1998)
7. Kouroupetroglou, G., Viglas, C., Stamatis, C. Pentaris, F.: Towards the Next Generation of Computer-based Interpersonal Communication Aids. In: 4th European Conference for the Advancement of Assistive Technology, AAATE97, pp. 110-114. IOS Press, Amsterdam (1997)
8. Kouroupetroglou, G., Pino, A.: ULYSSES: A Framework for Incorporating Multi-Vendor Components in Interpersonal Communication Applications. In: 6th European Conference for the Advancement of Assistive Technology, AAATE 2001, pp. 55-59. IOS Press, Amsterdam (2001)

9. Kouroupetroglou, G., Pino, A.: A New Generation of Communication Aids under the ULYSSES Component-based Framework. In: 5th International ACM SIGCAPH Conference on Assistive Technologies, ASSETS 2002, pp. 218-225. ACM Press, New York (2002)
10. Kouroupetroglou, G., Pino, A., Viglas, C.: Managing Accessible User Interfaces of Multi-vendor Components under the ULYSSES Framework for Interpersonal Communication Applications. In: International Conference on Human-Computer Interaction, HCI 2001, pp. 185-189. Lawrence Erlbaum Associates, Inc., Mahwah (2001)
11. Poulson, D., Nicolle, C.: Making the Internet Accessible for People with Cognitive and Communication Impairments. Universal Access in the Information Society 3, 1, 48-56 (2004)
12. Judson, A., Hine, N., Lundälv, M., Farre, B.: Empowering Disabled Users through the Semantic Web. In: 1st International Conference on Web Information Systems and Technologies, WEBIST, pp. 162-167. INSTICC Press, Portugal (2005)
13. Judge, S., Lysley, A., Walsh, J., Judson A., Druce S.: OATS - Open Source Assistive Technology Software - a Way Forward. In: 12th Biennial Conference of the International Society for Augmentative and Alternative Communication (on CD-ROM). ISAAC (2006)
14. Rackensperger, T., Mcnaughton, D., Krezman, C., Williams, M., D'Silva, K.: When I First Got it, I Wanted to Throw it over a Cliff: The Challenges and Benefits of Learning Technology as Described by Individuals who use AAC. Augmentative and Alternative Communication 21, 165-186 (2005)
15. Savidis, A., Stephanidis, C.: Inclusive development: Software Engineering Requirements for Universally Accessible Interactions. Interacting with Computers 18, 1, 71-116 (2006)
16. Platt, D.: The COM+ Event Service Eases the Pain of publishing and Subscribing to Data. Microsoft Systems Journal 14, 9 (1999)
17. Viglas, C., Kouroupetroglou, G.: An open machine translation system for augmentative and alternative communication. In: Miesenberger, K., Klaus, J., Zagler, W.L. (eds.) ICCHP 2002. LNCS 2398, 698-706. Springer, Heidelberg (2002)
18. Karberis, G., Kouroupetroglou, G.: Transforming Spontaneous Telegraphic Language to Well-formed Greek Sentences for Alternative and Augmentative Communication. In: Vlahavas, I.P., Spyropoulos, C.D. (eds.) SETN 2002. LNCS 2308, 155-156. Springer, Heidelberg (2002)
19. Xydas, G., Kouroupetroglou, G.: The DEMOSTHeNES Speech Composer. In: 4th ISCA Tutorial and Research Workshop on Speech Synthesis, pp. 167-172. Kluwer Academic Publishers, Dordrecht (2001)
20. Crowston, K., Howison, H., Annabi, J.: Information Systems Success in Free and Open Source Software Development: Theory and Measures. Software Process Improvement and Practice 11, 2, 123-148 (2006)
21. Stamelos, I., Angelis, L., Oikonomou, A., Bleris, G.: Code Quality Analysis in Open-Source Software Development. Information Systems Journal, 2nd Special Issue on Open-Source, 12, 1, 43-60 (2002)
22. Woltosz, W.S.: A Proposed Model for Augmentative and Alternative Communication Evaluation and System Selection. Augmentative and Alternative Communication 4, 233-235 (1987)
23. Spinellis, D., Szyperski, C.: How Is Open Source Affecting Software Development? IEEE Software 21, 1, 28-33 (2004)
24. Tate, K.: Sustainable Software Development: An Agile Perspective. Addison-Wesley Professional, Boston (2005)