A Proposal for Development of Software to Support Specific Learning Difficulties

Jasna Hamzabegovic*, Damir Kalpić**
* University of Bihać,
Pedagogical Faculty/Department of Mathematics and Informatics, Bihać, Bosnia and Herzegovina
Email: hjasna@bih.net.ba
** University of Zagreb,
Faculty of Electrical Engineering and Computing, Zagreb, Croatia
Email: Damir.Kalpic@fer.hr

***Abstract* — Among the plenty of educational software, therapeutic software for dyslexia is not so abundant. The paper considers two different approaches to development of software for supporting specific learning difficulties. One could be offering of visual end user programming tools to speech and language pathologists so that they could develop specific software applications customized for their patients. The other approach could be the engagement of students majoring in educational sciences who are studying informatics as their secondary discipline to develop applications. Paper presents concept of end user development environment aimed for speech and language pathologistsas well as educational pilot-applications developed by the students.**

Keywords — end-user programming; speech and language pathologists; dyslexia; therapeutic software.

I. INTRODUCTION

One of the specific learning disability is dyslexia. The International Dyslexia Association says that Dyslexia is a language-based learning disability [1]. Dyslexia refers to a cluster of symptoms, which result in people having difficulties with specific language skills, particularly reading. According to the World Federation of Neurology, dyslexia is "a disorder manifested by difficulty in learning to read, despite conventional instruction, adequate intelligence and socio-cultural opportunity" [2].

Speech and language pathologists (SLPs) are experts who diagnose, forecast, prescribe for and/or remediate speech and/or language disorders [3]. They have the specialized knowledge and experience that are needed to identify communication disorders and provide help that children need to build their language literacy skills.

Acquiring literacy skills is the key priority for dyslexic people. Information and communication technology can help dyslexic people in all areas of literacy. The use of the appropriate software applications to support literacy skills can offer opportunities for those with dyslexia to work independently and successfully, in education, work and home environments.

Paper is addressing very important problem of developing software specialized for supporting dyslexic people. Although there exists a number of software solutions for dyslexia, the main problem targeted within this paper is lack of localized software solutions for the South-eastern Europe region. Also, the paper targets the concept of software development environment enabling non-technical users, particularly SLPs to design their own customized software for treating dyslexia.

The paper proposes one possible way how to engage wider number of stakeholders in designing and application of these kind of software. Two approaches are presented. One is related to development of end user visual development environment that would enable end users to develop and utilize applications for their needs. The other approach is to student engagement in related applications development.

II. CONTRIBUTION OF SPEECH AND LANGUAGE PATHOLOGISTS IN CREATING SOFTWARE TO HELP CHILDREN WITH DYSLEXIA

Today dyslexia affects 20% of the U.S. and 8 to 10% of the European population, according to research from the National Institutes of Health [4] and this percentage has no tendency to decline because of constant appearance of new cases. Dyslexia is a lifelong condition. However, with proper support and therapy, people with dyslexia can learn strategies how to read, write and acquire academic knowledge. With good instructions, symptoms of dyslexia can be alleviated. Today, there exist varieties of hardware[1] and software[2] tools that address this problem.

There are many educational games, designed to prevent and relieve the symptoms of dyslexia. Software to support phonics and spelling offer a range of games or activities to practise skills in reading highly frequent words, phonics and spelling. They usually use selected lists or a structured program. Such as: Wordshark, Lexion, Catch Up 1, 2 & 3, Nessy, etc.[5] However all software is not widely available. It may be due to its cost or it may not correspond to specific languages required for all the children. Research shows that most of educational games cover the English-speaking world. English is the official language in 45 countries. However,

[1] Ergonomic keyboard with colored keys, mouses and tablets, touch sensitive monitors.

[2] Word processors, portable and spelling dictionaries, programs to help with reading and writing.

most people of the world speak Mandarin Chinese, in 27 countries French is spoken, in 20 Spanish, in 17 Arabic, etc. Parents in Bosnia and Herzegovina are not able to obtain any software in their language that they could use at home as a successful therapy, because this type of software does not exist. Croatian Dyslexia Association with the Ministry of Science, Education and Sports issued in 2008 the first interactive CD for teaching to read and write, made in Adobe Flash technology for children older than kindergarten age, including first and second grade of elementary school (6-8 years). In the year 2011, the second interactive CD *Let's Read Together 2* for students from second to fourth grades was issued [6].

Dyslexia manifests differently and requires individual and specific approach for each case. Existing of software applications aimed for children with reading difficulties does not fit all. The software does not match symptoms or problems to be treated, or it does not conform to language requirements. A possible solution is inclusion of SLPs in the development of software applications for therapy. It would be desirable to offer a software development environment, accessible and easy to use for SLPs who have acquired only a basic knowledge of informatics. It would allow them to develop multimedia and interactive applications for the therapy of children with reading difficulties. In such case, the SLPs, as the persons most competent to provide successful rehabilitation, could dedicate them to develop individual applications for each patient, unlike educated programmers who are designing applications for dyslexia in general, because they find them more profitable. Inclusion of SLPs in the creation of the application software could result in a large collection of software applications to cover numerous different symptomatic forms of dyslexia. They could cover multiple languages, and engage SLPs around the world in usage and upgrading of this software. Another reason for building a software development environment is the reduction of the digital divide[3] between the dyslexic in developed and underdeveloped societies, and English and non-English speaking countries. Including of SLPs in development of this software would create a community whose motivation, creativity and innovation capabilities might significantly exceed the contribution of hired professional programmers in providing applications to support reading and learning for students with dyslexia. In addition, this would reduce the impact of unfavourable demographic factors, such as place of residence or socioeconomic status of the family. It could also help to overcome language barriers and obstacles stemming from a peculiar social order.

A. End User Programming

Within software engineering, a special discipline has developed. Its aim is to encourage the development of software by the end user (EUD[4]). EUD represents "a set of methods, techniques and tools that allow users of software systems, acting as non-professional developers, to create, modify or extend application software for their own use" [8]. Specifically, EUD allows end users to design and adapt the user interface and functionality of the software. This is very important because end users know their own context and needs and they are best aware of changes and developments in their field. Through EUD, end users can configure the software to fit their needs better than it would have been possible without the EUD. There exist much more end users than professional programmers. The estimates are that 30 users correspond to a single professional programmer [9]. EUD allows the formation of much larger corps of people who take part in writing of software.

However, EUD is different from traditional software development, and the attempt to access EUD by imitating the traditional way does not lead to successful results. End users do not receive training in programming languages, in development processes, modelling and diagram notation, like the professionals do. They often lack sufficient time or motivation to learn the traditional techniques, because the end users usually want to write code to achieve some ad-hoc short-term goal in order to facilitate the solution of a current problem. Accordingly, EUD supports and responds to demands by offering the proper tools, structure and development processes that are useful and quick to learn and easy to integrate into the existing practice.

Today, end users create different software. Some programming environments that are used by end users encompass: budgetary systems, web authoring tools and graphical languages for creating different simulations [10, 11, 12, 13]. The research results of the US Bureau of Census and Bureau of Labor from 2005 show that in the United States there are about 3 million professional programmers, more than 12 million people who are programming at work, and over 50 million of those using spreadsheets and databases [14]. Based on the statistical data from Bureau of Labor, professor Mary Shaw from Carnegie Mellon University in the opening lecture at the 7th Conference of engineering software for Europe (2009) estimated that in year 2012 over 90 million Americans would use a computer at work. Of this number, only 3% would be professional programmers. In these 90 million are not included unemployed end-users who use computers at home or non-American population. The current estimates of the number of computer users in the world differ heavily but they are counted in billions. End user programming is progressing rapidly, resulting in millions of end users that create their own software.

B. Custom programming for SLPs

Most SLPs do not possess the technical knowledge needed to create their own software. So, most of the software applications are created by experts in software development, who are not experts in the field of speech and language pathology. Software that is available for SLPs' use, often does not meet their individual therapeutic needs.

A sustainable and continuous way of introducing new software in therapeutics is still possible. The idea is to create a development environment that should help the SLPs to develop software for own aims, without having

[3] "Digital divide" is division between those who have and those who have no access to the new IT, to the Internet and communication technologies [7]

[4] End-User Development

to acquire the standard programmers' knowledge and skills.

Visual development environment (VDE) is the suitable solution for SLPs enabling them to desing interactive multimedia content to help their students in development of skills in reading and writing. It should be possible for them to design their applications using ready-proven components. The construction of the tested components will result in obtaining higher quality of software. Many large companies have enabled user-customized programming through different programming systems intended for the end user, e.g. flexible word processing, spreadsheets and presentations that are very popular.

This, however, has not yet happened in the area of therapeutic software. This market niche is small and it should rely on public funding. However, SLPs would have great benefits as well as their patients who would use this software. Today, there are several visual programming languages available,such as Kodu, Alice, Scratch and others that allow the creation of interactive multimedia games and educational applications. However, they are mainly aimed for general purpose, while for a specific therapy, SLPs should have different software.

The problem could be simplified by supplying software components with a high level of functionality, by presenting technical concepts in a way that these concepts are close to natural speech mental models and by integrating all aspects of the tools necessary for development.

Research on the needs of end users and cases from the field has led to results that indicated the use of visual programming.

Visual Programming Language (VPL) is a programming language that allows users to make software applications by using graphical elements rather than text commands. Depending on the type and extent of visual expression, they are divided into visual languages based on icons, based on forms and based on the diagrams. Visual programming environment provides graphic elements (icons) which are manipulated by the user and the user interactively participates in construction of the program.

C. *The concept of the visual development environment intended for SLPs*

Nowadays, *Microsoft Office* and *Paint* belong to the most popular programs. Despite of their simplicity in using, they are also very functional. As the majority of these programs are close to SLPs, we conclude that future visual development environment should resemble them (Fig. 1).

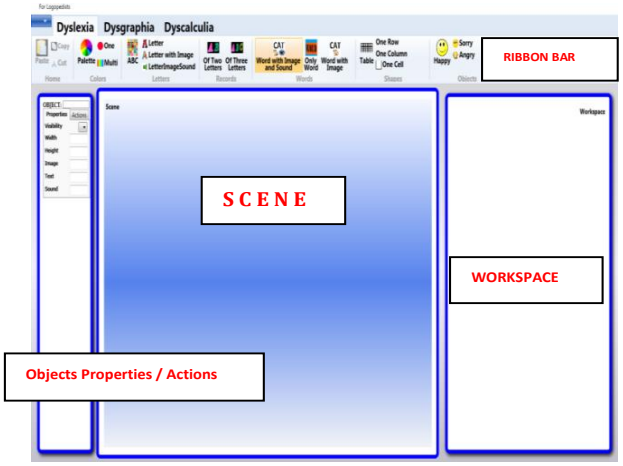

Fig. 1. GUI for SLPs

VDE includes graphical editing of user interface elements on a "drag-drop" principle and a WYSIWYG[5] editor (Fig. 2).

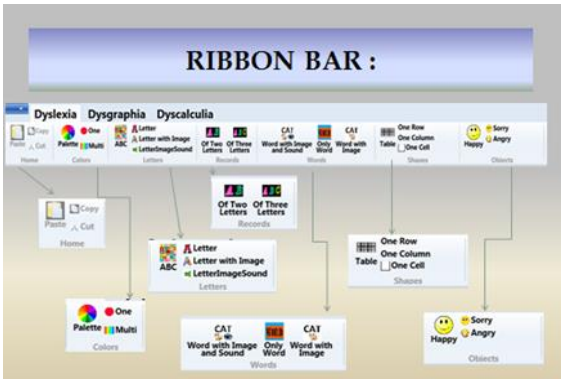

Fig. 2. Toolbar

VDE is based on an event driven visual programming language with back-end code associated to the various objects that responded to user-initiated events. SLP defines possible user actions and determines the corresponding object behaviour, i.e. its reaction as response. The objects to be treated are: letter (with or without accompanying sound and/or picture), syllable (two- or three-lettered), word (with or without accompanying sound and/or picture) (Fig. 3), row, column, table etc. An object disposes with a specified set of actions, behaviour and properties.

Fig. 3. Objects

.

[5]Acronym of : **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et

Every object reacts to its own set of events (e.g. one left click, double left click, one right click, drag&drop), with associated actions (e.g. play sound, animate image, create an empty row, create one row with letters, etc) because all the objects have not the same associated set of events and actions. Therefore, one left click on the object "Letter" will activate the associated mp3 file. In such case, after a one left click on the letter, the user will hear how it is pronounced.

By dragging the icons from the toolbar and customizing their characteristics, SLP would create an exercise for the child (Fig. 4).

Description of the program logic: When SLP clicks on the control in the Toolbar (e.g. Word with Image and Sound), the selected control appears in the WorkSpace (in the right hand side of the window). SLP forms object from this control by adjusting its properties through selection of source files for images, sound and text and by determining its height and width. Using drag&drop, the created object is brought to the scene.

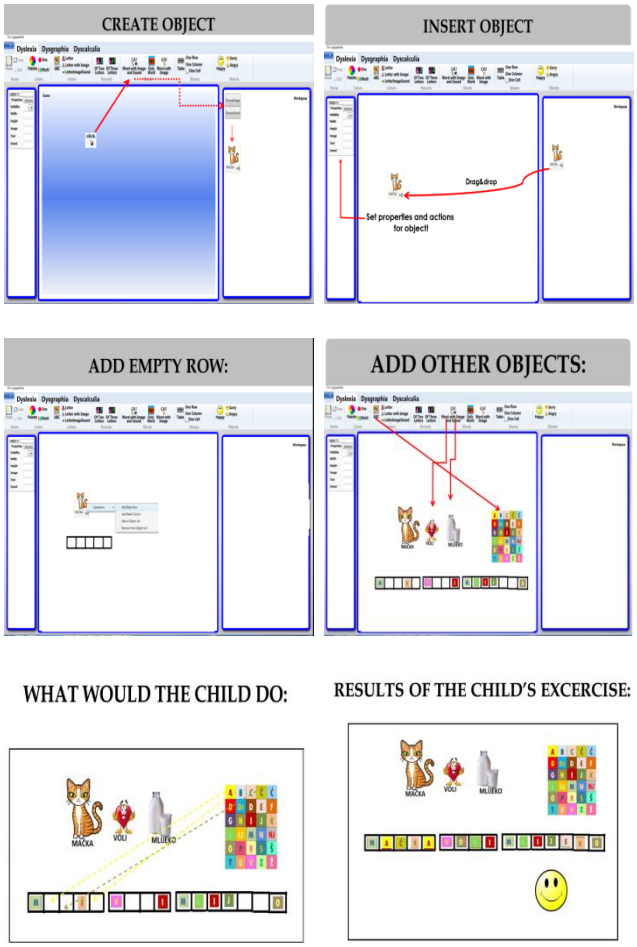

Fig. 4. Create an exercise for the child

With a right click on the object in addition to properties settings, SLP can also define future object behaviours as response to a user's action. For example, one click on the object may activate a sound file with pronunciation of a word. The properties and behaviours can be adjusted also in the left hand side of the window. Depending on the object type, with right click on the object, the SLP can enable some operations (Fig. 5). In this way, the SLP, with a right click on the word dragged to the scene, will be able to choose one of the operations: Add Blank Row, Add Row with Letters, Add Fill Row, Delete Object, etc.

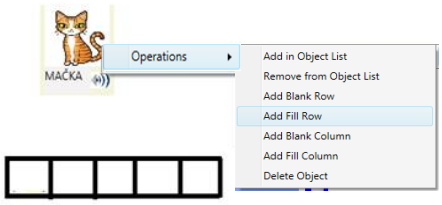

Fig. 5. Operations associated with object

The idea of a future visual development environment and its application for development of therapeutic applications are the result of an analytical and empirical research performed by the author with the target population of SLPs. Their behaviour was analysed in order to identify the control logic and to specify the elements and actions they use in natural language in everyday use. Based on these abstractions, a new language and environment would be designed. After this general approach, it was analysed how the SLPs describe the behaviour of the future development environment, which should support their development of therapeutic applications and how these applications should look like.

The technology to enable the construction of such a development environment could be Microsoft .NET Framework[6]. The .NET Framework is a large set of classes used in programming of various types of applications. A very popular type are the Windows Presentation Foundation (WPF) applications. Microsoft Visual Studio development environment for professional programmers supporting in addition to textual, also the visual programming is itself a WPF application. As future development environment should enable exactly the visual programming, it appears that WPF as a part of .NET Framework could be a good choice and that is where our future activities will be directed.

III. Student Projects of Creating Educational Applications

Many large companies such as *Microsoft* have enabled customized programming to millions of users through a variety of special purpose software systems, e.g. flexible word processing (*Word*), spreadsheet (*Excel*), create and use databases (*Access*), presentations (*PowerPoint*). These tools are very popular. In favour of this is the information published by Michael Schultz, the director of *Microsoft Marketing* at the International Consumer Electronics Show in Las Vegas in 2009 that about half a billion people were using *Microsoft Office* [15].

End user programming is underrepresented in the field of educational applications. Professional developers are not concerned with the development of this type of

[6] 2002. first version .NET 1.0, 2012. last version .NET 4.5

software. This market is small, because the funds allocated by the state are insufficient. Such is the situation in Bosnia and Herzegovina.

IT studies at all the educational colleges generally fall into a group of secondary information studies, where students study computer science together with another discipline. These studies educate future teachers who transmit their knowledge to the younger generation in elementary and secondary schools.

A. Students' projects

At the Faculty of Education within the University of Bihać there exists the Department of Mathematics offering specialisation track in Mathematics and Informatics.

The aim of the study at this Faculty is to produce modern informatics teachers who will share the contemporary knowledge and experience with younger generations. These teachers should also acquire practical knowledge and programming skills, enabling them to create interactive educational software for their own needs and the needs of colleagues in other fields. The current curriculum emerged as the product of great efforts of a single enthusiastic teacher. He succeeded to teach students without any prior programming knowledge or greater ambition in IT, to reach the same level as developers of simple *Windows Presentation Foundation* (WPF) and *Silverlight* applications, in only three semesters. All the students proved their knowledge by creating their own WPF applications. Great success of the implementation of this curriculum and the students' strong desire and willingness to produce modern *Windows* applications have resulted in an initiative that the new curriculum, to be implemented the next year, introduces a course called *Programming of educational applications*, which is not taught on any other pedagogical faculty in Bosnia and Herzegovina. We believe that students of a pedagogical faculty who are studying IT and Didactics and Methods are predisposed to become developers of modern educational applications. In favour of this, some of the student applications can witness.

Developed applications are rich with multimedia, like images, sound, video and animations. They have been developed in the WPF technology as standalone desktop applications. They can be realised also as Silverlight web applications and as Windows Phone mobile applications. To define the user interface, the declarative XAML language was used, while C# processes in the background the required functional tasks and caters for interactivity. These both components, each powerful in its way, are mutually complementary and they enable the creation of rich interactive multimedia applications. The tools that the students were using for development of these pilot applications are MS Visual Studio 2010 (for programming of the applications functionality) and MS Expression Blend 4 (for design of user interface).

Students have used the rich WPF animation system for various purposes. For example, in educational programs, the animation is used for visualization of matrix multiplication, for computing of determinants (Fig.6), visualization of the Pythagorean rule, calculation of the cube area (Fig.7) and similar mathematical processes with full interactivity between the user and the program.

Fig. 6. Calculation of the determinants

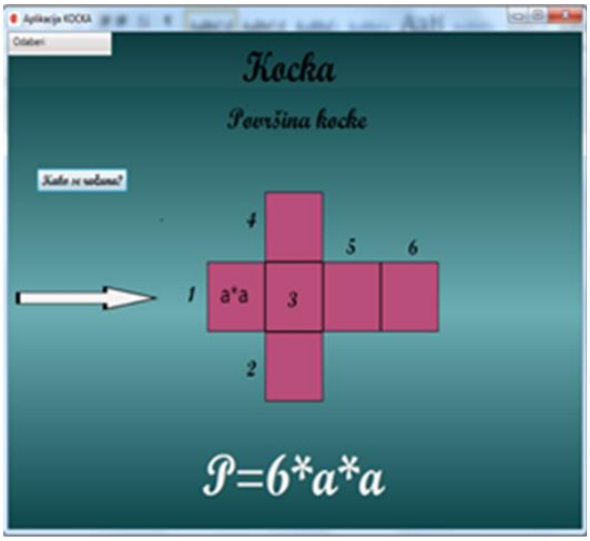

Fig. 7. An animated view of the procedure of calculating area of cubes

Some students were engaged in producing entertainment programs like games, in interactive colouring books where a child, while applying a colour, hears the name of the colour (Fig. 8) or in simple visual multimedia book featuring the four seasons of the year (Fig. 9).

The topics covered in the presented pilot applications were chosen by students themselves, basing on their personal preferences. Crucial was their acquaintance with the processed contents, and their affinity towards a certain field, like mathematics, games for kindergarten children, etc. These software applications have been initially designed to evaluate students, but the aim is to perform their user evaluation with SPLs, to improve the

applications regarding evaluation results, and finally to use the applications in clinical practice.

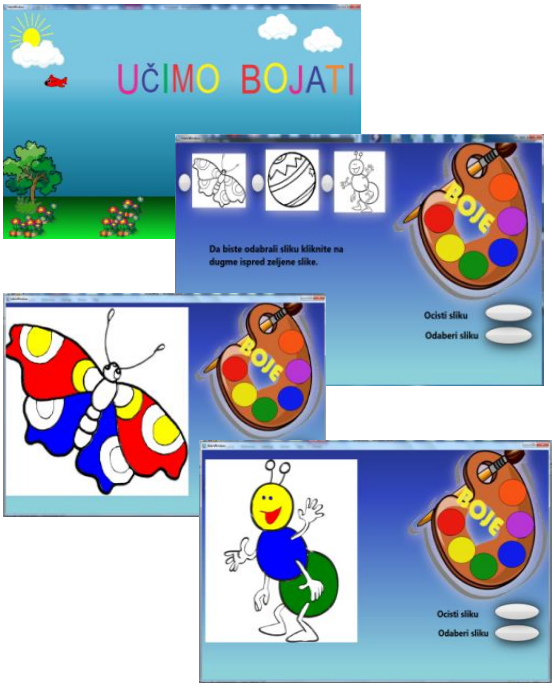

Fig. 8. An interactive colouring book

Fig. 9. An animated book featuring the seasons

B. Perspectives and possible use

This curriculum is in its infancy and will be further optimized to become better in terms of efficiency and quality of results. Students will become better programmers. They are a great resource. They can be included in the course *Programming of educational applications*, in the working process of creating software for children with disabilities and children with specific learning disabilities. In that case, it is necessary to provide their co-operation with educational and rehabilitation experts, with psychologists and pedagogues. We witness the fact that there are few of such application programs, and the need for them is present. Participation of students into these kind of projects raise their self-confidence and increase their social awareness. They will gain extensive programming experience and maybe some of them would decide to engage in professional production of educational software in order to modernize the educational process and make it more attractive. Tablets are increasingly entering the classrooms, and we are unprepared. The development of educational software and software to support children with disabilities, unfortunately, has not yet attracted enough attention.

On the other hand, every child with disabilities could become more creative, efficient and due to deliberated ways, achieve more. A child can learn to think better, remember better, and to detect the ways to compensate less developed skills.

Young developers would have an incentive to develop their own programming skills, and the kids would get fun in plenty of interactive multimedia content of various character.

IV. Conclusion

End user programming is a discipline of software engineering that can still bring more benefits. Although researchers and software engineers have done a lot to enable the end users to create their own applications, some fields are still uncovered. One of them is the area of therapeutic software for the prevention and treatment of dyslexia. Today, this software is still developed by professional programmers, and they are unfortunately scarce or not suitable for non-English users. Including of SLPs in the design of software applications suited for the needs of the therapy they pursue, could provide a considerable collection of amusing, educational and therapeutic applications, available to children and their parents, and other colleagues SLPs.

In a multitude of various software, software applications for children with disabilities do exist but only rudimentary in some non-English regions. As students of Mathematics and Informatics at the Faculty of Education, University of Bihać (Bosnia and Herzegovina), through their regular undergraduate programme learn how to develop educational applications, the same students could be engaged to develop this type of applications in their study projects. Their voluntary involvement could gradually provide a collection of interesting and simple therapeutic software that would be available to all children, and particularly to

those with disabilities. This could facilitate their education and rehabilitation, and decrease the digital divide between them and their peers who do not face any educational problems and who regularly use computers for entertainment and education.

REFERENCES

[1] http://www.interdys.org/, accessed February 2013.

[2] "What Is Dyslexia?" AVKO Education Research Foundation, http://www.avko.org/Info/dyslexia/what_is_dyslexia.htm , accessed July 2012.

[3] http://www.asha.org/policy/RP1976-00129.htm, accessed May 2013.

[4] http://kidshealth.org/parent/medical/learning/dyslexia.html, accessed January 2013.

[5] http://bdatech.org/learning/i-c-t-for-literacy/, accessed May 2013.

[6] http://www.hud.hr/w-tekstovi/w-literatura.html , accessed July 2011.

[7] M.Canellopoulou-Bottis, K. Einar Himma, The Digital Divide: "A Perspective for the Future", The Handbook of Information and Computer Ethics, Edited by Kenneth Einar Himma and Herman T. Tavani, New Jersey, 2008.,p:621-638

[8] H.Lieberman, F. Paterno and V.Wulf (eds.) 2006. "End User Development", Springer

[9] M.Shaw, The keynote slides : "The Challenge of Pervasive Software to the Conventional Wisdom of Software Engineering", The 7th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE) August 24-28 2009., Amsterdam (http://spoke.compose.cs.cmu.edu/shaweb/)

[10] M.Burnett, S.Chekka, R.Pandey: FAR: An End-User Language to Support Cottage EServices. In: Human-Centric Computing Languages and Environments, pp. 195–202. IEEE, Los Alamitos (2001)

[11] H.Lieberman, (ed.): Your Wish Is My Command: Programming By Example. Morgan Kaufmann Publishers, San Francisco (2001)

[12] J.Pane, B.Myers, L. Miller : Using HCI Techniques to Design a More Usable Programming System. In: Proc. IEEE Human-Centric Computing Languages and Environments, pp. 198–206. IEEE, Los Alamitos (2002)

[13] A.Repenning, A.Ioannidou.: AgentCubes: Raising the Ceiling of End-User Development in Education through Incremental 3D. In: IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 27–31. IEEE, Los Alamitos (2006)

[14] C.Scaffidi, M. Shaw and B.Myers, "Estimating the Numbers of End Users and End User Programmers," in IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05). 20-24 September, 2005. Dallas, Texas: pp. 207-214.

[15] M.Schultz, "Microsoft Office Is Right at Home." Jan. 2009.Web: http://www.microsoft.com/en-us/news/features/2009/jan09/01-08cesofficeqaschultz.aspx , accessed August 2012.