

INTRODUCTORY PROGRAMMING EDUCATION FOR VISUALLY IMPAIRED

Mario Konecki¹

¹University of Zagreb, Faculty of Organization and Informatics, Pavlinska 2, Varaždin, Croatia

Abstract

Recurring problems that are present in introductory and other programming courses have created a negative and pessimistic opinion among programming novices that enroll these courses. These problems have also created a fear about programming among programming novices which is passing from one generation to another. This fact is also supported by many researches which report that programming courses experience relatively high failure and dropout rates. There have been efforts and attempts to change this kind of situation but the results of introductory as well as other programming courses have not changed much. The problems in introductory programming courses are even more challenging in the case of visually impaired programming novices who experience many similar but also many specific problems which include problems in understanding of various programming principles and problems in remembering of programming syntax. The problems of visually impaired programming novices regarding their introductory programming education are described in this paper. Various tools and other means of making programming courses easier for programming novices are also mentioned. Finally a solution that would aid visually impaired programming novices in their first encounter with programming is presented and discussed.

Keywords: visually impaired, introductory programming education, problems, graphical user interfaces, GUIDL

-----***-----

1. INTRODUCTION

Introductory as well as other programming courses are present in all computer science studies. These courses despite of their importance report about rather high failure rates [4; 17]. This fact has caused programming novices to fear programming and to develop a negative and skeptic opinion about their programming education which has been passing from one generation of programming novices to another for years. Various efforts have been made in order to provide programming courses with means to help programming novices in understanding of abstract programming concepts and principles but the state and results of these courses have not changed in any greater amount. Programming concepts have simply proven to be too abstract and not intuitive enough to be easily comprehend by programming novices. The persisting high failure rates have also caused a relatively high dropout rates [17; 24] and further increase of fear among programming novices. This kind of state has formed an opinion about programming that is present and mostly consistent among students as well as among their teachers which both agree that programming is recognized and proven as hard to learn [3; 6; 19; 21]. The experience has shown that even the simplest programming examples are in many cases not solved by programming novices who frequently show much lower test results than it would be expected. Some of the aspects that have been recognized as an important in order for programming novices to successfully deal with programming are developed problem-solving skills and algorithmic way of thinking [11; 12] which are not inborn to programming novices who experience many problems in trying to analyze given problems and in construction of an appropriate algorithmic solutions [13].

The problems in programming courses are even more challenging for visually impaired programming novices who have many specific problems such as difficulties in perceiving the program structure or in understanding of different programming constructs and the way in which they function. Another set of problems for visually impaired is connected to usage of different programming language development environments which are in most cases based on graphics and the usage of mouse. In order to try to help programming novices to understand various programming concepts many visualization tools and techniques have been developed [25; 26]. Since this approach is not possible for visually impaired programming novices another approach had to be devised. This approach includes simplicity as its main aspect and it is also useful in the case of sighted programming novices. By making programming scope more focused and simpler the learning of programming principles should be made easier and less overwhelming. There are two approaches that are based on this principle. The first is called the sub-language approach and the second is the mini-language approach [5]. Another aspect that needs to be considered is the nature of programming examples. Most of these examples in most programming courses are of abstract nature that is not connected to practical situations and because of that programming novices have problems in trying to relate to these examples so it can be said that more common and everyday situations should be used as a basis for creating programming examples in order to make them more understandable for programming novices. This alteration in programming examples design should make them more engaging and interesting which should consequently increase the motivation of programming novices to practice and go through the given examples and tasks.

2. PROBLEMS IN PROGRAMMING EDUCATION

Programming is a complex and abstract field of study that requires somewhat different approach than most of other courses. Programming is not merely a knowledge, it is a skill [7] and as all other skills it requires a lot of time and a lot of practice in order for skill to develop and to be adopted by programming novice. Programming novices' habits are different from those that are needed to acquire a skill since most of their primary and secondary education was about remembering and reproducing some facts. Programming novices are more used to learning and remembering elements and patterns than to practice and construction of new solutions. In order to promote skill adoption which requires a lot of practice it is important to make programming interesting and more suitable from programming novices' point of view in order to increase their motivation to do programming exercises. Programming novices frequently cannot develop a working programs at the end of their studies and few of them continue to program after their programming courses [3; 15; 16]. Gomes and Mendes [6] state that problems which consider programming novices are emerging from several reasons:

- Programming demands a high abstraction level
- Programming needs a good level of both knowledge and practical problem-solving techniques
- Programming requires a very practical and intensive study, which is quite different from what is required in many other courses (more based in theoretical knowledge, implying extensive reading and some memorization)
- Usually teaching cannot be individualized, due to common classes' size
- Programming is mostly dynamic, but usually taught using static materials
- Different students have different learning styles and can have several preferences in the way they learn while teachers' methodologies many times don't take into consideration the student's learning styles
- Programming languages have a very complex syntax with characteristics defined for professional use and not with pedagogical motivations

Problems in programming education are frequently reported as recurring [18] and they obviously need further attention. In the case of visually impaired programming novices one of the problems is how to perceive the structure of written programs. The task of writing and debugging programs is challenging for sighted programming novices and it is much more difficult for visually impaired to go back and forth in a program, change its elements and their order and remember where particular element is located. Visually impaired programming novices require tasks to be as simple as possible and as engaging as possible since other aiding concepts such as visualization cannot be used.

3. VISUALLY IMPAIRED AND PROGRAMMING

Visually impaired have been interested in computers and programming since the very beginning of computer industry. Along with this interest there have been efforts to provide visually impaired computer users with adequate aiding tools. Various text-to-speech synthesizers were developed and this aiding tools along with various Braille typewriters were sufficient for visually impaired to use computers and perform programming tasks without problems since all user interfaces were based on text. The development of GUI (Graphical User Interface) has however created a variety of inconveniences and problems for visually impaired computer users and programmers since existing aiding technology was not able to deal adequately with graphical elements and layouts. The context of screen was very hard to describe and professional programmers encountered a variety of problems in the area of development of graphical user interfaces since programming languages' development environments were not suitable to be used in this area without the usage of point-and-click actions. Despite the problems that have become even more prominent in the following years as visual technology has become more and more complex, the interest of visually impaired for programming has remained present throughout the years [1] to this day. American Foundation of the blind programmers counts over 130 blind programming professionals who are registered as members [1]. Programming is also recognized as a promising career opportunity by many countries [28] and Europe in general [27].

There are several well-known aiding tools that are used to make computer usage available to visually impaired [22; 20; 29; 30]:

- JAWS
- HAL Screen Reader
- COBRA
- Window Eyes

Beside the mentioned screen-readers there are also several aiding solutions that are used for visualization of graphical elements to visually impaired such as Audiograph [2] that combines a variety of sounds and their pitches in order to represent graphical shapes and tactile panels [23] which are actually multiline Braille panels which are used to represent graphical elements by lowering or elevating pins that are placed in every single row of this kind of panels. An examples of this kind of technology is shown in Fig. 1 [23]. Finally there are some aiding tools that are designed to help visually impaired in some areas of programming education. Some of these tools are:

- APL (Audio based Programming Language) which is as an educational tool that provides its own environment for teaching visually impaired about the basic programming concepts
- Javaspeak tool which is designed to describe the program code syntax and organization by usage of audio output

- SODBeans (the Sonified Omniscient Debugger in Netbeans) which includes its own compiler/debugger with added audio features
- WAD (Wicked Audio Debugger) is a special debugger developed for Visual Studio which incorporates audio features

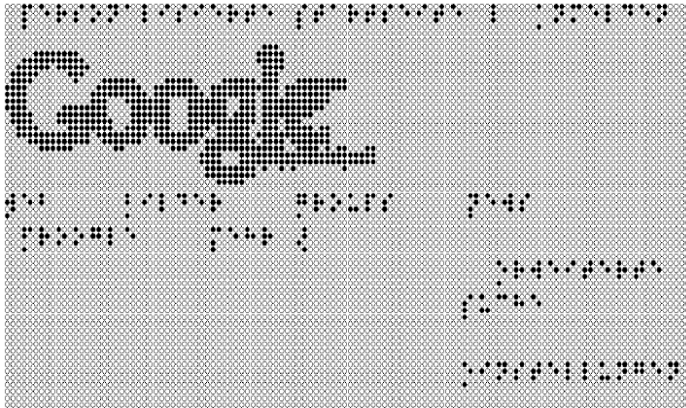


Fig. 1. Tactile panel

Programming education of visually impaired is actual and important issue that needs more attention and more solutions that will enable visually impaired programming novices to be able to pursue programming career that was and still is a promising job opportunity for visually impaired computer users. Equal opportunities are also one of important issues and providing visually impaired programmers novices with aiding tools that will enable them to learn and perform all activities of overall software development in order to be an equal team members compared to their sighted colleagues is something that also deserves more attention and effort.

4. GUIDL AS AIDING TOOL

Two aspects that are of great importance when talking about aiding visually impaired programming novices in their programming activities are simplicity of programming language and its usage and the degree in which it is possible to make tasks interesting and engaging for visually impaired in order to increase their motivation and willingness to do programming tasks in order to adopt a sufficient level of programming skills. In order to achieve the mentioned goals the GUIDL (Graphical User Interface Description Language) system [8; 9; 10; 14] is proposed as an aiding tool in introductory programming activities of visually impaired programming novices. GUIDL system is designed as an aiding tool that helps visually impaired programmers in creating graphical user interfaces in a way that is sufficiently simple and intuitive to be practically applicable. GUIDL language is the basis of GUIDL system which is designed to simplify the description of graphical elements as much as possible. A portion of GUIDL language grammar in EBNF [31] form is given below [14]:

```
formcode = 'Frm ' | 'frm ' ;
formend = ('End' | 'end'), [eol];
controlname = quote, word, quote, eol;
word = alphabeticcharacter, {alphabeticcharacter | digit};
```

```
formattributes = frmcommonattributes,
windowstateattribute, {colorattribute};
frmcommonattributes = textattribute,
frmrestcommonattributes;
frmrestcommonattributes = frmsizeattribute,
locationattribute;
frmsizeattribute = sizecode, (frmsize | frmwidth, ws
,frmheight), eol;
locationattribute = locationcode, xposition, ws, yposition,
eol;
frmsize = 'frmsize1' | 'frmsize2' | 'frmsize3'
```

Beside the simple syntax GUIDL system has built-in aiding concepts and features that make development of usable graphical layouts easier. These concepts include [14]:

- Predefined gradual form sizes
- Predefined gradual graphical element sizes
- Predefined values of width/height attribute
- Possibility to divide forms into quadrants
- Possibility to position graphical elements into one of form quadrants
- Possibility to define the position offset of forms
- Possibility to define the position offset of graphical elements
- Detection of problems with position of graphical elements (graphical element out of form boundaries)
- Automatic correction of problems with form dimension and position (form out of screen boundaries)

By its simplicity and aiding concepts GUIDL system enables visually impaired programming novices to be included in graphical design which was not appropriately supported by existing aiding solutions and in this way makes their first programming steps less overwhelming because of intuitive tasks that produce concrete results compared to more abstract logical or mathematical tasks. It also enables visually impaired to share their work in this area with others as a quick result of their first coding activities. In this way the very beginning of their programming actions becomes simpler, more interesting and more engaging which results in greater number of solved exercises and in greater motivation to continue dealing with programming. Another important aspect of GUIDL system that should provide additional motivation is that created graphical user interfaces can be translated into other desired programming language format that can be included into corresponding programming environment as a part of actual project. This enables programming novices to truly share their results even as a part of some real solutions. All GUIDL code is syntactically checked by its scanner, lexer and parser which create a basis for translation of GUIDL code into some other format (for example VB.NET or C#). Along with syntactical analysis the logical analysis is also performed which applies mentioned aiding concepts in order to prevent some logical layout errors (for example overlapping graphical elements or form position that results in a part of form not being visible on the screen). Conceptual model of GUIDL system and steps of its functionality are shown in Fig. 2 and Fig. 3 [14].

GUIDL system has been tested as an aiding solution on 32 participants who have been given several examples and tasks upon which they were given a chance to describe their experience with GUIDL system. The conclusion was that GUIDL system has proven to be simple enough as well as engaging and interesting in order to maintain the focus and attention of visually impaired programming novices in their first programming steps. This first impressions about programming are of vital importance because upon them the visually impaired programming novices form their attitude towards programming in general upon which depends whether visually impaired programming novices will continue to learn programming as something interesting or something that is frustrating.

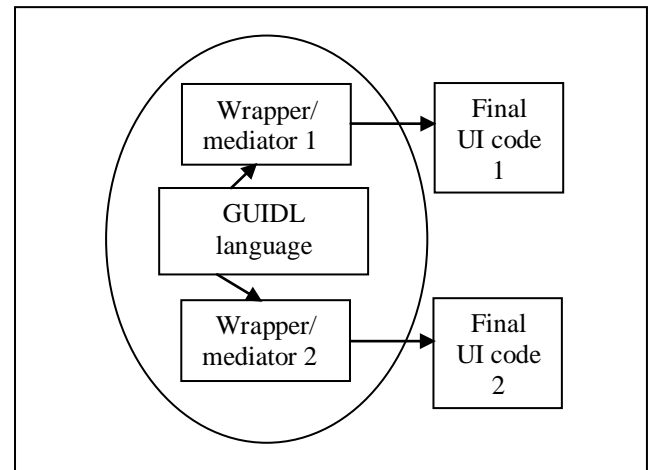


Fig. 2. GUIDL system conceptual model

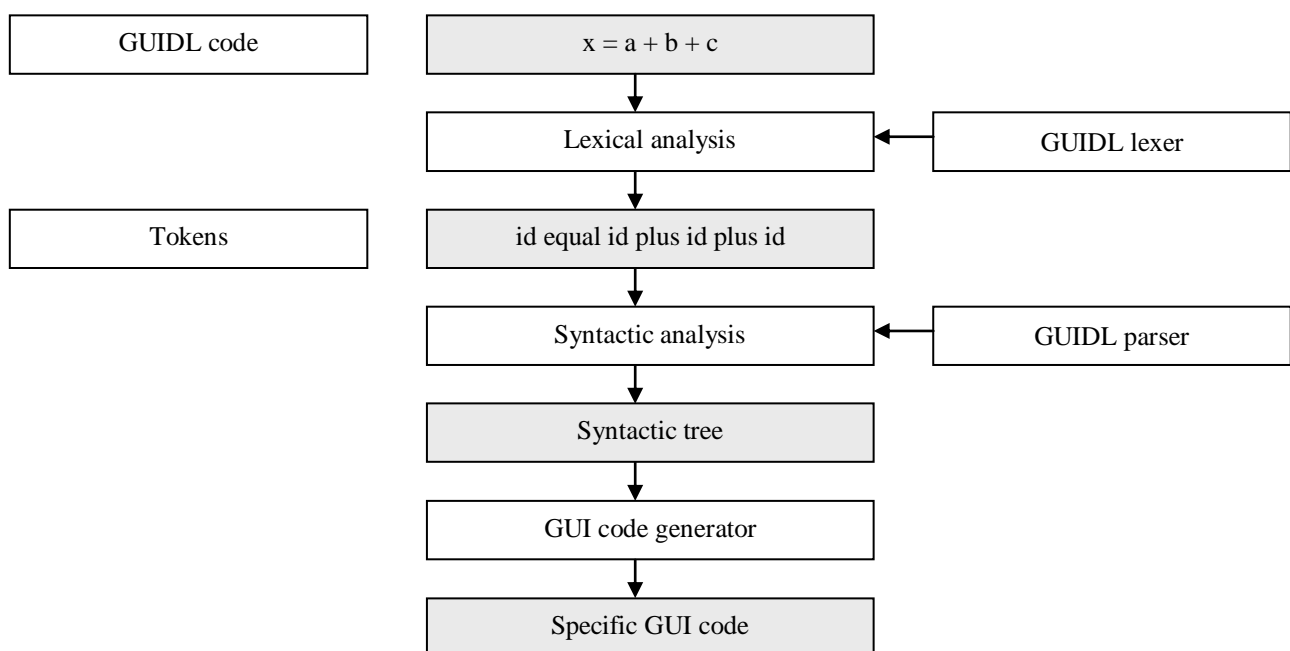


Fig. 3. Functionality of GUIDL system

Finally upon the first impression and consequently the rest of programming education also depends whether visually impaired programming novices will continue to deal with programming as a programming professionals or will they turn to other fields of computer science instead. Some aspects that have been reported as important during the research were proper instructions as well as proper documentation with examples that was given to all participants. They stated this aspect as an important part of using any of aiding tools which in many cases lack sufficiently detailed instructions. They have also stated that the syntax of GUIDL language was intuitive and simple enough to be used even without any previous experience with this language. All this indicates that GUIDL system provides a promising aiding tool for visually impaired that can be used in their first programming steps.

5. CONCLUSION

Programming courses are present in all computer science studies as a courses of great importance. The results however show that these courses experience rather high failure as well as dropout rates. Programming concepts which are abstract and not intuitive along with complex syntax of programming languages that are used in education make programming courses hard for programming novices. The problems in programming education are even more challenging when dealing with visually impaired programming novices who experience many specific problems such as understanding the structure and dynamics of program code. Since programming is a skill it requires a lot of practices and that means doing a lot of programming tasks. To do so programming novices have to be properly motivated. Simplicity has been identified as an important aspect in programming education for visually impaired as

well as making programming tasks less abstract and more interesting which is of great importance in order to achieve greater willingness of visually impaired programming novices to practice programming. The decision whether one will continue to deal with programming or not greatly depends on the very first steps taken in the field of programming which calls for greater attention and more effort invested in their design.

In this paper GUIDL system has been presented as a suitable solution for making programming courses easier and more understandable for visually impaired programming novices. GUIDL system makes programming tasks more interesting because it shifts the nature of these tasks to a more concrete form of graphical user interfaces which programming novices can share with their colleagues. GUIDL system also enable visually impaired programming novices to include their graphical user interfaces in real projects because it gives them a mean of translating their work into format that can be included into programming environments that are used in practice. This makes programming more engaging and simpler which results in increased motivation and does not overwhelm programming novices. By using GUIDL visually impaired programming novices are able to go through their first programming steps in a more suitable manner which in the end results in greater willingness to continue to deal with programming which has been recognized as one of promising career opportunities for visually impaired who are interested in computers and their usage. Further research about additional aspects and components that are suitable to be a part of GUIDL system as well as further testing of GUIDL system will be a part of future work.

REFERENCES

- [1] Alexander, S. (1998). Blind Programmers Face An Uncertain Future. *ComputerWorld*, 32(44), 86-87.
- [2] Alty, J. L., & Rigas, D. I. (1998). Communicating Graphical Information to Blind Users Using Music: The Role of Context. In C.-M. Karat, A. Lund, J. Coutaz, & J. Karat (Eds.), *CHI '98 Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 574-81). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.
- [3] Baldwin, L. P., & Kuljis, J. (2001). Learning programming using program visualization techniques. In D. Eileen (Ed.), *Proceedings of the 34th Annual Hawaii International Conference on System Sciences* (pp. 1051-1058). IEEE, Washington, DC, USA.
- [4] Bennedsen, J., & Caspersen, M.E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2), 32-36.
- [5] Brusilovsky, P., Kouchnirenko, A., Miller, P., & Tomek, I. (1994). Teaching Programming to Novices: A Review of Approaches and Tools. In T. Ottoman, & I. Tomek (Eds.), *Proceedings of ED-MEDIA 94 - World Conference on Educational Multimedia and Hypermedia* (pp. 103-110). Association for the Advancement of Computing in Education, Vancouver, BC, Canada.
- [6] Gomes, A., & Mendes, A. J. (2007). An environment to improve programming education. In B. Rachev, A. Smrikarov, & D. Dimov (Eds.), *Proceedings of the 2007 international conference on Computer systems and technologies* (pp. 88:1-6). ACM, New York, USA.
- [7] Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (pp. 53-58). LTSN Centre for Information and Computer Sciences, Loughborough, UK.
- [8] Konecki, M., Kudelić, R., & Radošević, D. (2010). Challenges of the blind programmers. In B. Auer, M. Bača, & M. Schatten (Eds.), *Proceedings of the 21st Central European Conference on Information and Intelligent Systems* (pp. 473-476). FOI, Varaždin, Croatia.
- [9] Konecki, M., Lovrenčić, A., & Kudelić, R. (2011). Making Programming Accessible to the Blinds. In P. Biljanović, K. Skala, S. Golubić, N. Bogunović, S. Ribarić, M. Čičin-Šain, D. Čišić, Ž. Hutinski, M. Baranović, M. Mauher, & L. Ordanić (Eds.), *Proceedings of the 34th MIPRO International Convention on Computers in Technical Systems* (pp. 180-184). IEEE.
- [10] Konecki, M. (2012). A New Approach Towards Visual Programming for the Blinds. In P. Biljanović, Z. Butković, K. Skala, S. Golubić, N. Bogunović, S. Ribarić, M. Cicin-Sain, D. Cisić, Z. Hutinski, M. Baranović, M. Mauher, & J. Ulemek (Eds.), *Proceedings of the 35th MIPRO International Convention on Computers in Technical Systems* (pp. 1076-1081). IEEE.
- [11] Konecki, M., & Petrić, M. (2014). Main problems of programming novices and the right course of action. In T. Hunjak, S. Lovrenčić, & I. Tomičić (Eds.), *Proceedings of the 25th Central European Conference on Information and Intelligent Systems* (pp. 116-123). FOI, Varaždin, Croatia.
- [12] Konecki, M., & Mrkela, V. (2014). Algorithmic thinking and animated interactive presentation of sorting algorithms in education of students. In V. Rajkovič, M. Bernik, & U. Rajkovič (Eds.), *Proceedings of the 17th International Multiconference Information Society - Education in Information Society* (pp. 105-112), IJS, Ljubljana, Slovenia.
- [13] Konecki, M. (2014). Problems in programming education and means of their improvement, *DAAAM International Scientific Book 2014*, in press.
- [14] Konecki, M. (2014). Inclusion of visually impaired in graphical user interface design. In R. Piltaver, & M. Gams (Eds.), *Proceedings of the 17th International Multiconference Information Society - Intelligent Systems* (pp. 54-57). IJS, Ljubljana, Slovenia.

- [15] Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppälä, O., Simon, B., & Thomas, L. (2004). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, 36(4), 119-150.
- [16] McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., Laxer, C., Thomas, L., Utting, I., & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4), 125-140.
- [17] Nikula, U., Gotel, O., & Kasurinen, J. (2011). A motivation guided holistic rehabilitation of the first programming course. *ACM Transactions on Computing Education (TOCE)*, 11(4), art. 24.
- [18] Pea, R. D. (1986). Language-independent conceptual "bugs" in novice programming. *Journal of Educational Computing Research*, 2(1), 25-36.
- [19] Peng, Wu. (2010). Practice and experience in the application of problem-based learning in computer programming course. In L. Yuting (Ed.), *Proceedings of the International Conference on Educational and Information Technology (ICEIT)* (pp. 1:170-172). IEEE.
- [20] Pitt, I. J., & Edwards, A. D. N. (1996). Improving the usability of speech-based interfaces for blind users. In *Proceedings of the Second Annual ACM Conference on Assistive Technologies (ASSETS)* (pp. 124-130). ACM, Vancouver, BC, Canada.
- [21] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Journal of Computer Science Education*, 13(2), 137-172.
- [22] Rosmaita, B. J. (2006). Accessibility First!: a new approach to web design. In *ACM Proceedings of the 37th SIGCSE technical symposium on Computer science education* (pp. 270-274). ACM, New York, NY, USA.
- [23] Rotard, M., Taras, C., & Ertl, T. (2008). Tactile web browsing for blind people. *Multimedia Tools and Applications*, 37(1), 53-69.
- [24] Smith, D. C., Cypher, A., & Tesler, L. (2000). Programming by example: novice programming comes of age. *Communications of the ACM*, 43(3), 75-81.
- [25] Smith, P. A., & Webb, G. I. (2000). The efficacy of a low-level program visualization tool for teaching programming concepts to novice C programmers. *Journal of Educational Computing Research*, 22(2), 187-216.
- [26] Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education (TOCE)*, 13(4), 15.
- [27] bfiSteiermark, "European Labour Market Report", available at <http://eurochance.brailcom.org/download/labour-market-report.pdf>, accessed: 28th October 2014.
- [28] "The employment of blind and partially-sighted persons in Italy: A challenging issue in a changing economy and society", available at http://www.euroblind.org/media/employment/employment_Italy.doc, accessed: 28th October 2014.
- [29] "Screen Reader COBRA", available at <http://www.baum.de/cms/en/cobra/>, accessed: 28th October 2014.
- [30] "GW Micro - Window-Eyes", available at <http://www.gwmicro.com/Window-Eyes/>, accessed: 28th October 2014.
- [31] "Information technology - Syntactic metalanguage - Extended BNF", ISO/IEC 14977, 1996, available at: <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>, accessed: 28th October 2014.