GUIDL IA: An Intelligent Assistant for Aiding Visually Impaired in Using GUIDL

Mario Konecki *, Robert Kudelić * and Hristijan Gjoreski **

* Faculty of Organization and Informatics, Varaždin, Croatia
** Jožef Stefan Institute, Ljubljana, Slovenia
mario.konecki@foi.hr; robert.kudelic@foi.hr; hristijan.gjoreski@ijs.si

Abstract - GUIDL (Graphical User Interface Description Language) system has been proposed as an effective aiding technology that enables inclusion of visually impaired computer users into activities of graphical user interface design. GUIDL system implements many aiding concepts that make creation of graphical user interfaces suitable for visually impaired. In order to make GUIDL system even more accessible to visually impaired a GUIDL IA (Intelligent Assistant) is proposed as a mean of providing a more suitable and simpler way of getting help about various parts of GUIDL system. GUIDL IA enables visually impaired to ask questions in the form of natural language compared to browsing through instruction manual which makes getting the right information easier and quicker. GUIDL IA also enables visually impaired to hear the answer through integrated text-to-speech synthesizer that enables more natural pronunciation of given answers. In this paper GUIDL IA is presented and discussed along with the results about its effectiveness.

Keywords - GUIDL, visually impaired, graphical user interfaces, intelligent assistant

I. INTRODUCTION

Programming is hard and challenging for sighted students to learn because of its abstract nature and a specific way of thinking that it requires [1; 2; 3, 4]. The challenge is much more prominent when it comes to visually impaired students [5]. Visually impaired students experience difficulties in trying to imagine various programming structures and concepts. Nevertheless, visually impaired have shown interest in programming since the very beginning of computer industry [6; 7] and their interest has not disappeared [8]. Along with this interest various means of enabling visually impaired to use computers were developed. Most commonly used aiding tools for visually impaired were based on text-to-speech synthesizers that enabled visually impaired computer users to work with computer programs because all programs’ interfaces were based on text. Some of the most known text-to-speech synthesizers are [7]:

- JAWS
- HAL Screen Reader
- COBRA
- Window Eyes
- Easy Web Browsing

These aiding tools also enabled visually impaired to participate in professional programming activities. The adoption of GUIs (Graphical User Interfaces) has brought a huge revolution to the world of computers and made computers much more accessible to a larger number of users. Unfortunately this progress has made things much more difficult for visually impaired. Existing aiding technology that was more than enough to support programming activities of visually impaired has become insufficient to support all new graphical concepts and to describe the graphical screens’ context. Along with these problems additional problems have emerged regarding tools for programming and development which have become based on graphical controls and point-and-click actions that have become necessary in order to create graphical user interfaces and to define many other settings. Writing program code still wasn’t so problematic since it was still based on text although some concepts and fragmentation of code that came along with new object-oriented programming languages and environments have made programming more complex.

Visually impaired programmers have become unable to participate as equal team members in all activities of software development especially in activities of graphical user interfaces development because textual definition of graphical user interfaces has become practically impossible and although some efforts in the area of aiding visually impaired to learn programming and perceive graphical elements have been made [9; 10; 11; 12; 13] none of them gave any long-term solution for creation of graphical user interfaces.

In order to enable visually impaired programmers to participate as equals in all activities of overall software development, including design of graphical user interfaces a GUIDL (Graphical User Interface Description Language) system with GUIDL language as its core part was developed as an aiding technology that is designed to overcome the obstacle of being unable to create graphical user interfaces in an adequate way for visually impaired [6; 7; 14]. GUIDL system is used as an aiding technology because it does not replace programming development environments. It is only used in a part of development that is dealing with graphical user interfaces creation which is done by using the GUIDL system upon which the rest of the development process is again performed in original development environment. In order to be usable GUIDL system had to meet the following requirements [7]:

- Easy Web Browsing
- Window Eyes
- COBRA
- HAL Screen Reader
- JAWS

These aiding tools also enabled visually impaired to participate in professional programming activities. The adoption of GUIs (Graphical User Interfaces) has brought a huge revolution to the world of computers and made computers much more accessible to a larger number of users. Unfortunately this progress has made things much more difficult for visually impaired. Existing aiding technology that was more than enough to support programming activities of visually impaired has become insufficient to support all new graphical concepts and to describe the graphical screens’ context. Along with these problems additional problems have emerged regarding tools for programming and development which have become based on graphical controls and point-and-click actions that have become necessary in order to create graphical user interfaces and to define many other settings. Writing program code still wasn’t so problematic since it was still based on text although some concepts and fragmentation of code that came along with new object-oriented programming languages and environments have made programming more complex.

Visually impaired programmers have become unable to participate as equal team members in all activities of software development especially in activities of graphical user interfaces development because textual definition of graphical user interfaces has become practically impossible and although some efforts in the area of aiding visually impaired to learn programming and perceive graphical elements have been made [9; 10; 11; 12; 13] none of them gave any long-term solution for creation of graphical user interfaces.

In order to enable visually impaired programmers to participate as equals in all activities of overall software development, including design of graphical user interfaces a GUIDL (Graphical User Interface Description Language) system with GUIDL language as its core part was developed as an aiding technology that is designed to overcome the obstacle of being unable to create graphical user interfaces in an adequate way for visually impaired [6; 7; 14]. GUIDL system is used as an aiding technology because it does not replace programming development environments. It is only used in a part of development that is dealing with graphical user interfaces creation which is done by using the GUIDL system upon which the rest of the development process is again performed in original development environment. In order to be usable GUIDL system had to meet the following requirements [7]:

- Easy Web Browsing
- Window Eyes
- COBRA
- HAL Screen Reader
- JAWS

These aiding tools also enabled visually impaired to participate in professional programming activities. The adoption of GUIs (Graphical User Interfaces) has brought a huge revolution to the world of computers and made computers much more accessible to a larger number of users. Unfortunately this progress has made things much more difficult for visually impaired. Existing aiding technology that was more than enough to support programming activities of visually impaired has become insufficient to support all new graphical concepts and to describe the graphical screens’ context. Along with these problems additional problems have emerged regarding tools for programming and development which have become based on graphical controls and point-and-click actions that have become necessary in order to create graphical user interfaces and to define many other settings. Writing program code still wasn’t so problematic since it was still based on text although some concepts and fragmentation of code that came along with new object-oriented programming languages and environments have made programming more complex.

Visually impaired programmers have become unable to participate as equal team members in all activities of software development especially in activities of graphical user interfaces development because textual definition of graphical user interfaces has become practically impossible and although some efforts in the area of aiding visually impaired to learn programming and perceive graphical elements have been made [9; 10; 11; 12; 13] none of them gave any long-term solution for creation of graphical user interfaces.

In order to enable visually impaired programmers to participate as equals in all activities of overall software development, including design of graphical user interfaces a GUIDL (Graphical User Interface Description Language) system with GUIDL language as its core part was developed as an aiding technology that is designed to overcome the obstacle of being unable to create graphical user interfaces in an adequate way for visually impaired [6; 7; 14]. GUIDL system is used as an aiding technology because it does not replace programming development environments. It is only used in a part of development that is dealing with graphical user interfaces creation which is done by using the GUIDL system upon which the rest of the development process is again performed in original development environment. In order to be usable GUIDL system had to meet the following requirements [7]:

- Easy Web Browsing
- Window Eyes
- COBRA
- HAL Screen Reader
- JAWS
• Easiness of usage
• Intuitive, simple and easy to understand syntax
• Independence of descriptive language
• Extensibility of descriptive language

GUIDL system has been designed to make creation of graphical user interfaces possible in a suitable manner for visually impaired programmers and to include them into all activities of programming development as equals. It is as such based on social model of aiding technology [15; 16]. Its GUIDL core language is also designed to be very simple and easy to use since easiness is one of the most important aspects of any aiding technology. In order to make learning about GUIDL system and its usage even easier an intelligent assistant is proposed that enables its users to ask a variety of questions about GUIDL system and to get a quick and clear answer opposed to going through manual page by page. In this way the usage of GUIDL becomes even more efficient and suitable for all visually impaired programmers. In this paper proposed intelligent assistant is described and a research about perception of visually impaired programmers regarding such solution is presented and discussed. This paper also gives some ideas for further research in this area.

II. GUIDL SYSTEM

GUIDL system as an aiding technology has been developed in order to enable visually impaired programmers to work in actual up-to-date technologies rather than having an isolated aiding ecosystem. GUIDL system with its GUIDL language provides easy and understandable way of creating graphical user interfaces that can be translated into desired programming language development environment format by the GUIDL system’s mediators which in the end leaves visually impaired programmers with interfaces that can be included into development environments in which some actual projects are being developed. In this way visually impaired are actually included into real projects rather than just having a way to show their ideas in isolated environment without any actual contribution to the real projects. Conceptual model of GUIDL system functioning is shown in Fig. 1 [7; 14].

GUIDL system has been developed to be simple to use with simple syntax of GUIDL language which is designed to be easy and intuitive to use. In order to provide an aiding mean in its full sense GUIDL system includes a number of aiding concepts which are aimed to make creation of graphical user interfaces simple and intuitive for visually impaired. These aiding concepts include [14]:
• Predefined gradual sizes of forms
• Predefined gradual sizes of graphical elements
• Predefined width/height attribute values
• Division of forms into quadrants
• Possibility to position graphical elements into one of form quadrants

Figure 1. GUIDL system’s conceptual model

• 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)

The aim of these aiding concepts is to catch those errors that occur on the logical level and are not of syntactical nature. These kind of errors can occur because it is very hard not to make them without seeing the overall screen context. By using these concepts visually impaired are able to create valid graphical user interfaces that do not have overlapping concepts, objects out of screen boundaries, or too small forms. Very important aspect of GUIDL system that is necessary in order for visually impaired to be able to organize objects on the screen is the possibility to divide the form into quadrants in which a certain object can then be placed. This makes organizing of form’s elements much easier which in the end results in better and more polished interface. These features combined with simple and accessible GUIDL language make this aiding technology usable and suitable for visually impaired. A part of GUIDL language grammar in EBNF form is shown below [14]:

```plaintext
• project = projectcode, controlname, form;
• projectcode = 'Project ' | 'project ';
• form = formcode, controlname, formattributes, [controldeclarations], formend;
• formcode = 'Frm ' | 'frm ';
• formend = ('End' | 'end'), [eol];
• controlname = qoute, word, qoute, eol;
• word = alphabeticcharacter, {alphabeticcharacter | digit};
• formattributes = frmcommonattributes, windowstateattribute, {colorattribute};
```
• \texttt{frmcommonattributes = textattribute, frmrestcommonattributes;}
• \texttt{frmrestcommonattributes = frmsizeattribute, locationattribute;}
• \texttt{frmsizeattribute = sizecode, (frmsize | frmwidth, ws, frmheight), eol;}
• \texttt{locationattribute = locationcode, xposition, ws, yposition, eol;}
• \texttt{frmsize = 'frmsize1' | 'frmsize2' | 'frmsize3'}
• \texttt{locationcode = 'Location = ' | 'Location=' | 'location = ' | 'location=';}
• \texttt{xposition = 'left' | 'center' | 'right';}
• \texttt{yposition = 'top' | 'middle' | 'bottom';}

Although GUIDL system has been designed to be simple to use some effort is needed to learn its rules and concepts. In order to provide visually impaired with something more than just textual or Braille manual an intelligent assistant is proposed as an aiding solution for helping visually impaired to learn and use GUIDL system.

III. GUIDL INTELLIGENT ASSISTANT

GUIDL IA (Intelligent Assistant) is proposed as a suitable aiding mean that is aimed at making GUIDL system easier to learn and use. GUIDL IA is designed to appear in a small part of the screen and to provide an easy way for visually impaired programmers to ask questions about GUIDL system in a form of natural language that they use in everyday conversation. In order to make GUIDL IA simple to use its interface consists of just two fields, one for input of questions and one read-only field for displaying the answers. In this way it is very easy to change focus from one field to another by using the keyboard. The interface of GUIDL IA is shown in Fig. 2. Another feature that is implemented in GUIDL IA is a possibility of opening additional materials in order to clarify or give more details about some aspect of GUIDL system. This feature is shown in Fig. 3. Using GUIDL IA is quite simple. It is sufficient to just enter any GUIDL system related question by typing it and hit the enter key to get the textual answer.

![GUIDL IA interface](image)

Since read-only field of GUIDL IA that displays the answers is sometimes not appropriate for a larger amount of information it is sometimes also convenient to open a web page or other web resource with additional information about the asked question and the given answer. In this way visually impaired are presented with more detailed information which can include a larger portion of example GUIDL code and in that way make it easier to understand certain elements of GUIDL system. GUIDL IA incorporates certain aspects of intelligent and expert systems that enable visually impaired programmers to use this assistant in a natural way just as they would communicate with their peers. This includes an extensive tolerance to various forms of questions. For example, all following forms of the same question are accepted and understood by GUIDL IA:

- Tell me how to create buttons?
- What is button and how it is used?
- I am not able to create a button.

Aside from extensive tolerance to different forms of questions GUIDL IA also supports variations of answers so that the same question can have many forms of answer that are randomly displayed which in the end results in somewhat more interesting form for GUIDL IA users. Another aspect that is built into GUIDL IA is a possibility to hear the answer in audio format. Converting text to speech is something that visually impaired computer users have been using for years and although this feature is supported by existing aiding technology which is mostly based on various text-to-speech synthesizers having its own variation of text-to-speech support leaves space for further customization and improvement of this feature and making it even more suitable for GUIDL IA. It is however still possible to use existing text-to-speech tools to convert textual GUIDL IA answers to audio format. Making the usage of GUIDL system even more simple with support of intelligent assistant has a goal to make GUIDL system even more effective and efficient in its main task of enabling visually impaired programmers to create graphical user interfaces.

IV. GUIDL IA EFFECTIVENESS

GUIDL IA has been included into GUIDL system in order to test its level of usefulness in making GUIDL system easier to use. The research was conducted among 24 visually impaired programmers who were given a
GUIDL system with GUIDL IA and were asked to try to do several tasks and report whether GUIDL IA has made using GUIDL system easier compared to using GUIDL system’s standard documentation. Research about the experience of participants with GUIDL IA was conducted by using a questionnaire with several questions about their attitude towards provided intelligent assistant and its way of functioning. All answers to given questions were based upon the Likert scale (5 - I strongly agree, 1 - I strongly disagree). Research results are shown in Table I.

The research results have shown that visually impaired programmers find GUIDL IA to be simple and useful tool that helps GUIDL system users in an interesting way that is quick and effective.

V. CONCLUSION

To learn how to program is very challenging for visually impaired who have nevertheless been included in computer and programming activities since the very beginning of computer usage. For years developed aiding technology that was designed to enable visually impaired to use computers was sufficient to support programming activities of visually impaired since all computer programs’ interfaces were based on text. With the occurrence of graphical user interfaces in computer industry visually impaired programmers have been left in difficult position because of graphical nature of software development environments, usage of point-and-click actions to set various settings, more complex object-oriented concepts and mostly because of usage of mouse to design graphical interfaces. Visually impaired programmers that were an equal team members found themselves unable to create graphical user interfaces in a suitable manner. In order to include visually impaired into all parts of overall software development a GUIDL system has been developed as an aiding solution which does not replace actual development environments but serves as an aiding technology in a part of software development that includes creation of graphical user interfaces. Although GUIDL system has been made simple it still requires some effort to learn how to use it. To make this process even simpler than it is an intelligent assistant GUIDL IA is proposed that enables its users to ask questions in natural language and to get an instant textual answer with a possibility of opening additional web resources to clarify given answers even further. In order to conclude about the effectiveness of GUIDL IA a research has been conducted in which visually impaired programmers where given GUIDL system with GUIDL IA for use on several tasks and were then asked to report about their experience. Research results have shown that GUIDL IA is simple, interesting and useful aiding solution for making usage of GUIDL system easier and quicker. Further improvements of GUIDL IA and research on a larger scale with additional objective measurements will be a part of future research.

REFERENCES


