

Drawing for Blind Learners: Assistive Technology for Graphical Design

James Ohene-Djan, Sandra Fernando

*Department of Computing, Goldsmiths, University of London
New Cross, SE14 6NW, England*

J.Djan@gold.ac.uk,
map01wf@gold.ac.uk

Abstract—One of the main problems faced by blind learners is a lack of drawing technologies that support images and diagram drawing without the help of a sighted support worker. Even though some technologies have been experimented with in the past, blind learners have not been keen on tactile drawing due to: the difficulty of the drawing task, the length of time taken to complete a simple task, and the inefficiency of the drawing experience. This paper presents a set of grammar tools that can be used by blind people to find spatial orientation, as well as knowledge and representation of the interactive drawing environment, to improve the understanding of a concept or a subject matter. The grammar can be adopted and personalised in education and learning when presenting technical diagrams. The grammar enables users to present graphics by mapping the cognitive visualisation of blind people into spatial information on a computer screen. This technique promotes an interactive and easy drawing environment to build objects, associations, and layout information by zooming, navigation, and grouping. It will lead to future possibilities such as 3D world modelling, printing, and multisensory integration of inputs and output methods.

Keywords— Graphics manipulation; grid-based drawing, navigation for blind users; grammar for graphics; blind drawing

I. INTRODUCTION

One of the main problems faced by blind learners is a lack of drawing technologies that support images and diagram drawing without the help of a sighted support worker. Even though some technologies have been experimented with in the past, blind learners have not been keen on tactile drawing due to: the difficulty of the drawing task, the length of time taken to complete a simple task, and the inefficiency of the drawing experience. This paper presents a set of grammar tools that can be used by blind people to find spatial orientation, as well as knowledge and representation of the interactive drawing environment, to improve the understanding of a concept or a subject matter. The grammar can be adopted and personalised in education and learning when presenting technical diagrams. The grammar enables users to present graphics by mapping the cognitive visualisation of blind people into spatial information on a computer screen. This technique promotes an interactive and easy drawing environment to build objects, associations, and layout information by zooming, navigation, and grouping. It will lead to future possibilities such as 3D world modelling, printing, and multisensory integration of inputs and output methods.

II. RELATED WORK

People perceive information by selecting, interpreting, and organising data in their minds [5]. Sound and sight provide complex information of a world that is inherently multisensory. Auditory information is mainly sequential whereas visual information is mainly simultaneous. Absence of visual information limits the ability of visually impaired people to build a perception of the world.

Research conducted in Italy in 2009 [12] related to vision issues experienced by blind people illustrates that activities can be learned and performed in the absence of sight. Individuals with no visual experience acquire knowledge of shapes, movements, and location of objects through non-visual sensory modalities. Brain plasticity research conducted in France in 2004 [8] shows that the primary visual cortex (PVC) appears to be persistently active in a mental imagery task when only verbal instructions were given to subjects who were blind from birth. This finding is very helpful in developing a sensory substitute tool such as the Blind drawing technology Pictures and diagrams can be constructed using low-tech manual solutions such as cardboard, pins, rubber strings, swell papers and other physical items to build tactile images. But there could be complexity associated with these techniques. Some pictures need a lot of paper for printing tactile objects that could contain an enormous amount of information, and thus the process could be expensive. Therefore other sensory channels such as audio and haptic technology have experimented with unimodal or multimode information presentation [9].

Haptic technology is still immature when compared with speech technology. Some of the common problems of touch-screen graphics manipulation are: the time it takes to complete a drawing compared with an audio drawing tool, difficulties in finding exact locations, navigating on a digital screen or a drawing sheet, finding the starting and ending points, and finding relative positioning. Nevertheless speech-based solutions are relatively mature, low cost, fairly easy to use, and take less time to complete in a learning environment [6] [13] [14] [15]. If a message is conveyed by voice, it has to be short and easy to understand due to the possibility of exceeding working memory capacity. Thus the processing load, disorientation, and redundancy effects have to be removed for good engagement or learning. Therefore distributing the information load across multiple channels is

advantageous [5] [10]. A good drawing enables the user to remember and access the floor plan, retrieve the objects in the floor plan, join them and make sense of them.

Kamel and Landay have introduced the IC2D product that divides the screen into nine navigable smaller workspaces [7]. There are pallets for users to select shapes, types, and colours. In order to make more precise selections, each of the nine cells is then divided into a further nine cells, which in turn are divided into nine more cells. Annotations are given for meaningful semantics so that the user can build an accurate mental image. System Kevin [3] enables users to read, edit, and create diagrams using N^2 chart. This system does not keep track of screen layout information but retains the mandatory information needed for a simple DFD (Data Flow Diagram). System PLUMB [4] uses linked lists and Heap's algorithms to store data in a data structure and to access it in a sequential manner. When the user selects "insert node" a new node is inserted to the existing link list, which would be acknowledged to the user. Some systems explore the different levels of the UML(Unified Modelling Language) diagrams, their hierarchic, floor plan, and spatial information, with a special emphasis on semantically annotating the data which results in interoperability of intranet and internet systems [11].

III. INTRODUCTION TO SETUP09

The approach of SETUP09 models the core of a drawing language, such as a compiler design. The introductory specification is to build and interpret user commands upon execution. Computer graphics are visualised by formal languages hence a formal drawing language is defined in this paper to help with blind drawing technology [1]. Examples of user commands are illustrated in this document using concrete syntax. The designed abstract compiler is an open language and can be redesigned to recognise commands from many other languages to address wider communities.

The grammar of the graphics language could be expanded based on user need. Some users might benefit from simple objects such as shapes and other users might benefit from flow diagrams. The architecture in Fig. 1 has three main parts: input handling, grammar formulation, and output handling. Inputs are extracted by Braille machines, keyboards, microphones, and touch pads. Outputs are demonstrated onscreen, and/or as image verbalisers, and/or as 3D printing/raised printing. The grammar is constructed by user commands breaking into tokens and parser validation using abstract and concert syntax. The interpretation of user commands is embedded into concrete classes that manipulate the graphics needed by the user.

A. Conceptual framework of the compiler design

When creating an image on the screen, which reflects what the blind user wants, the system follows several steps. User commands are first stored in an array of strings separating every single input, including spaces and commas. User commands are then reorganised by storing in an array list of tokens. A token has a type and a value. When the value is

recognised the programme assigns a type. If the token value is not recognised it passes an error message.

Tokens are then passed to a Main Parser function that translates the token into concrete syntax. The Main Parser function calls on a Secondary Parser function to retrieve its concrete syntax, while the Main Parser runs through an indefinite loop by checking the command list entered by the user. The Secondary Parser function reads through the array list of tokens while it meets a matching concrete syntax. Concrete commands are released as a Statement List. A Statement List has one or many statements.

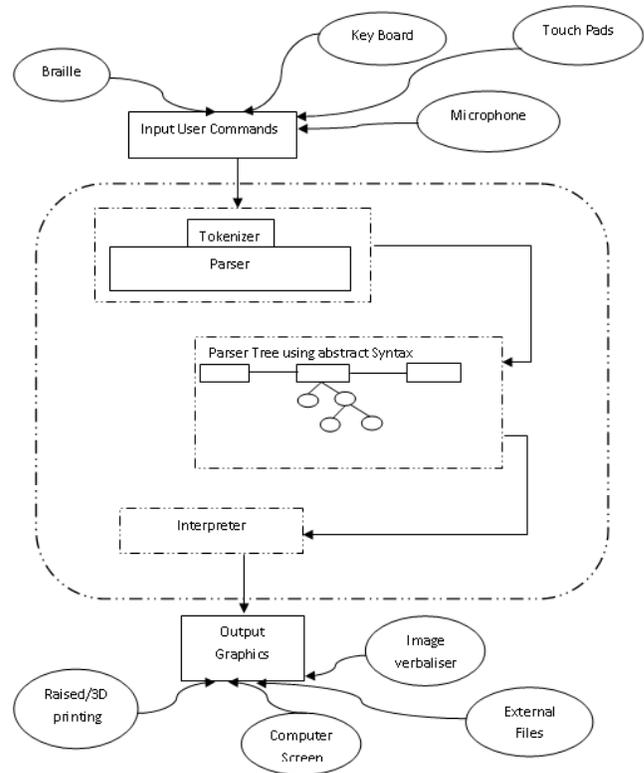


Fig. 1. Architecture of SETUP09 System

B. Modelling the grammar of the drawing language

The user commands of SETUP09 enable the manipulation of graphics, navigation, and actions to construct images on the screen. These user commands have been designed by exploring blind people's cognitive capabilities and linguistic behaviour. The commands are intuitive, interactive, and easy to learn. Below are explanations of some user commands.

The areas of the drawing screen are divided into North, South, East, West, North-East, South-East, South-West, North-West, and Centre. An area on the screen has nine different points such as North, South, East, West, North-East, South-East, South-West, North-West, and Centre. Users can enter one or many commands at the user prompt to manipulate an image. This is referred to as a Statement or Statement List. The BNF (Backus-Naur Form) grammar can be improved by adding many more commands such as save, delete, undo, redo,

erase, import, export, etc. For the purpose of this paper only some commands are discussed.

- To get the focus of an area of a screen: Zoomin [name of the area]
- To extract the focus out of an area: Zoomout
- Users can directly call library objects by their primitive names, such as circle, rectangle, etc.
- A line can be manipulated by calling it: line [point1][point2]
- An arc can be manipulated by calling it: arc [point1][point2] [angle]
- A drawing can be defined by giving it a name and a set of commands.
- Users can directly call user-defined objects by their given names, such as mycircle, myrectangle, etc.
- A point on the screen can be assigned to a variable. These variables can be used as a reference point to draw lines and write text.
- Text can be written on the screen by directly calling a point or user-defined point.

Grammar in Backus-Naur Form

```

<statementList>-><abstractStatement> | <abstractStatement>, <statementList>
<abstractStatement>-><zoomout> | <zoomin> | <line> | <assign> | <write> | <arc> |
<shape> | <call> | <define> | <>nullStatement>

<call>->'call'<name>
<define>->'define', <name>, <StatementList>
<zoomout>->'zoomout'
<zoomin>->'zoomin'<location>
<assign>-><variable>='<point>
<write>->'write'<point><variable>
<line>->'line', <point> | <variable>, <point> | <variable>
<arc>->'arc', <point> | <variable>, <point> | <variable>, <angle>

<location>->N|S|E|W|NE|NW|SE|SW|C
<point>->N|S|E|W|NE|NW|SE|SW|d
<angle>->L|R|T|B
<shape>->circle | triangle | box
<variable>->String | integer
<name>->String
  
```

Fig. 2. System Grammar in BNF

The grammar in Fig. 2 is demonstrated in Backus-Naur Form (BNF) and describes formal notation for encoding grammar intended for a blind drawing language. This

grammar shows the basic user inputs for which the Java compiler is designed for the purpose of this paper [1] [2].

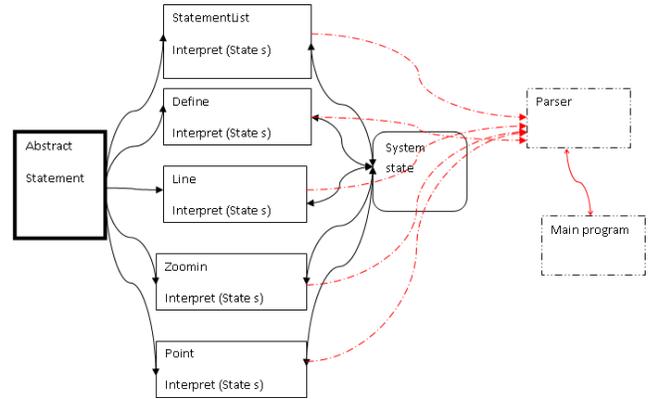


Fig. 3. Program classes of the compiler

The parent abstract class named Abstract Statement inherits subclasses in Fig. 3. The parent class has an abstract method named Interpret, which executes a drawing. This method in the abstract class is designed to do the drawing required by the user. The method Interpret needs parameters of the current state of the system such as x and y positions, zoom size, zoom level, and other important system information to execute a drawing, and returns a new system state to the system at the end of a drawing. All subclasses have their own Interpret method to interpret user commands.

C. Graphics construction

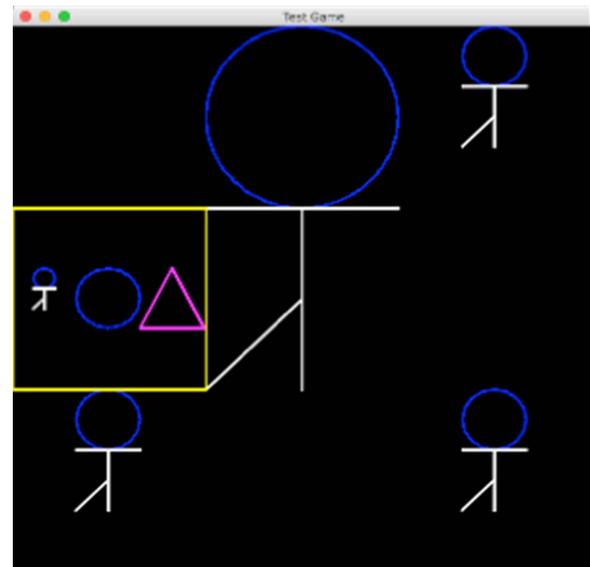


Fig. 4. System output of various user commands

System initiates a new **state**, a frame or panel at the start of the program. Every time a user inputs a command, the system converts the user command to its abstract syntax and the concrete classes perform the interpretation of user commands by drawing the images such as zoomin, line, circles, define. SETUP09 System outputs are captured in Fig. 4, Fig. 5, Fig. 6, Fig. 7, and Fig. 8 for various user commands.

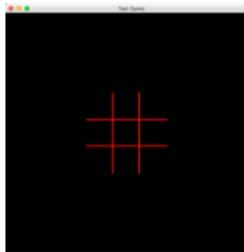


Fig. 5. System output
User Command: zoomin C

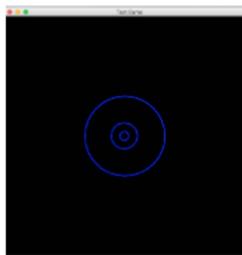


Fig. 6. System output
User Command: zoomin C,
circle, zoomin C,
circle, zoomin C, circle

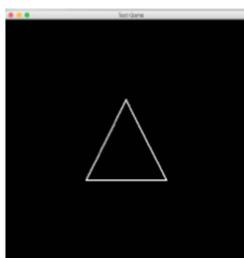


Fig. 7. System output
User Command:
zoomin C,
line N SE, line SE SW,
line SW N

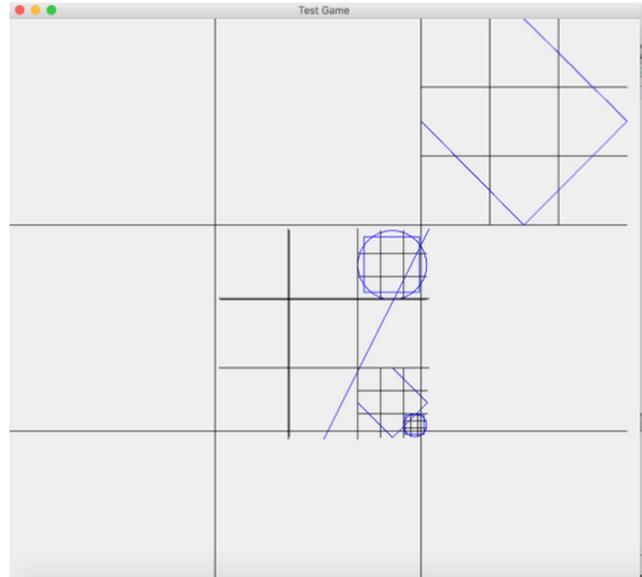


Fig. 8. System output for the program below

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    //create an initial state
    State newState = new State(g);
    //call zoomin
    newState = new Zoomin(new Point("C")).interpret(newState);
    //call line
    newState = new Line(new Point("NE"), new Point("S")).interpret(newState);
    //define mylines
    StatementList L = new StatementList(new Line(new Point("N"), new Point("E")),
        new StatementList(new Line(new Point("E"), new Point("S")),
            new StatementList(new Line(new Point("S"), new Point("W")), null));
    newState = new Define("MyLines", L).interpret(newState);
    //call zoomin again
    newState = new Zoomin(new Point("SE")).interpret(newState);
    //call mylines
    newState = new Call("MyLines").interpret(newState);
    //define mycirclebox
    StatementList M = new StatementList(new Circle(),
        new StatementList(new Rectriangle(), null));
    newState = new Define("MyCircleBox", M).interpret(newState);
    newState = new Zoomin(new Point("SE")).interpret(newState);
    //call mycircle
    newState = new Call("MyCircleBox").interpret(newState);
}
```

Fig. 9. Main Java program with hardcoded user inputs

IV. FUTURE WORK

For the purpose of this paper the SETUP09 model is introduced with possible grammar that programmers and designers are able to improve upon when developing the SETUP09 product for assistive technology users. Different input and output technologies illustrated in the architecture of SETUP09 are yet to be implemented and tested in different platforms and with a target audience.

Another main area is the system error detection and error correct arena [9] as this is an important element of successful blind drawing architecture. The final step would be to test the system with prospective blind users when the system is fully developed. There could be iterative developments based on test results before releasing the model to the assistive technology users.

This type of high-level interaction can help to overcome the limitation of drawing due to the absence of visual modality and helps to reduce the cognitive load required to reason the floor plan details and sizes [14]. The SETUP09 system can be used to draw scientific diagrams during classroom activities, as well as for graphics presentation and visualisation of imaginative thinking such as art in general.

Furthermore the visual modality may not be suitable in all contexts of activity, such as locating objects in hidden surfaces where alternatives must be sought [14]. Even sighted people could benefit using the techniques (location, size, object tracking) presented in the paper when structuring images on non-visual surfaces. The SETUP09 technique of drawing is an easy and efficient method to build a cognitive image of a person's mind and helps people with no vision by contributing to the cognitive presentation of blind image processing.

V. CONCLUSION

This paper demonstrates an architecture in which blind people manipulate graphics through a computer system. The grammar is introduced in a way that could be customised and reused in different contexts and other spoken languages. The architecture enables users to define objects and reuse when required, find a specific point on the screen by relative location tracking, and specify size by relative screen dimensions. The division of the surface into nine different sections works as the primary formulation of location as well as sizes of unseen surfaces. The abstract model aims to answer statements such as:

- The ability of the software to generate a cognitive map in a blind person's mind by going through a virtual environment.
- The ability of the software to give a clear reflection of a blind user's mental image by a computer-generated picture map.

The program answers these queries by going through a set of user inputs, tokens abstractions and concrete grammar. The architecture is complete in the sense that it is expressive enough to override all design decisions of the language construction. Attention is drawn to the fact that the design of the grammar, parser, and interpreter could be presented separately from the input interaction, as well as output of the graphics. Therefore alternative and advancing technologies in

different contexts can help blind and sighted people to conceptualise the grammar required to manipulate desired graphics.

ACKNOWLEDGEMENT

This research is funded by MACS charity, UK.

REFERENCES

- [1] A. W. Apple, *Modern Compiler Implementation in Java*, Cambridge University Press, 2002
- [2] P. Blackburn, J. Bos Johan, Kristina Striegnitz, "Learn Prolog Now!", 2012, Available: <http://www.learnprolognow.org/lpnpage.php?pageid=top>
- [3] P. Blenkhorn, D. G. Evans, "Using Speech and Touch to Enable Blind People to Access Schematic Diagrams", *Journal of Network and Computer Applications*, Vol. 21, 1, January 1998, 17-29, doi.org/10.1006/jnca.1998.0060
- [4] M. Calder, R. F. Cohen, J. Lanzoni, N. Landry, J. Skaff, "Teaching Data Structures to Students who are Blind", *Proc. 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, June 2007, Dundee, Scotland, UK, doi.org/10.1145/1269900.1268811
- [5] M. A. Hersh, M. A. Johnson, A. Michael, *Assistive Technology for Visually Impaired and Blind People*, Berlin; London: Springer, 2008
- [6] J. Sanchez, M. Saenz, et al., "Enhancing Navigation Skills Through Audio Gaming", *Proc. CHI 2010, Human Factors in Computing Systems*, New York, USA, 2010, 3991-3996, doi.org/10.1145/1753846.1754091
- [7] H. M. Kamel, J. A. Landay, "Sketching Images Eyes-free: A Grid-Based Dynamic Drawing Tool for the Blind", *Proc. Fifth International ACM Conference on Assistive Technologies*, July 2002, Assets '02, 33-40, Edinburgh, Scotland, doi.org/10.1145/638249.638258
- [8] S. Lambert, E. Sampaio, Y. Mauss, C. Scheiber, "Blindness and Brain Plasticity: Contribution of Mental Imagery?" *Cognitive Brain Research*, Vol. 20, 1, 1-11, 2004, Publication: Elsevier Science B.V., Amsterdam
- [9] M. H. Cohen, J. P. Giangola, J. Balogh, *Voice User Interface Design*, Boston, Mass.; Harlow: Addison-Wesley, 2004
- [10] C. Nicholas, M. Lusine, B. Leonardo, "A Multimodal Interface Device for Online Board Games Designed for Sight-Impaired People", *IEEE Transactions of Information Technology in Biomedicine*, 14, No. 2, March 2010
- [11] O. Metatla, N. Bryan-Kinns, T. Stockman, "Constructing Relational Diagrams in Audio: The Multiple Perspective Hierarchical Approach", *ASSETS*, 08, 2008
- [12] E. Ricciardi, D. Bonisno, "Do We Really Need Vision? How Blind People See the Actions of Others", *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, Vol. 29, 31, 2009, 9719, Publication: Society for Neuroscience
- [13] S. Zhu, J. Feng, "Investigating Grid-Based Navigation: The Impact of Physical Disability", *ACM Transaction on Accessible Computing*, Vol. 3, 1, Article 3, September 2010
- [14] T. Ishihara, H. Takagi, T. Itoh, C. Asakawa, "Analyzing Visual Layouts for Non-Visual Presentations - Document Interface", *ASSETS* 06, 2006, ACM 1-59593-290-9/06/0010
- [15] Y. Lee, C. Yao, C. Hsieh, J. Wu, Y. Hsieh, C. Chen, R. Liang, Y. Chen, "Egg Pair: A Hearing Game for the Visually Impaired People Using RFID", *IEEE 17th International Symposium on Consumer Electronics (ISCE)*, 2013, doi.org/10.1109/ISCE.2013.6570237, IEEE Std. 802.11, 1997