A Hybrid Approach to Computer Aided Drawing Tool for Blind People

Sandra Fernando†, Bal Virdee, Karim Ouazzane

School of Computing and Digital Media, London Metropolitan University, 166-220 Holloway Rd, N7 8DB,

†[s.fernando@londonmet.ac.uk](mailto:s.fernando@londonmet.ac.uk)

**Abstract: Many software development methodologies introduced to date place users at the centre of development process. Although the user is an important asset in the software development cycle however the user-centred approach is not sufficient to develop a software product that is structurally robust and reliable. User involvement in the development process does not always guarantee resilience and a more efficient design. To address this a hybrid-based software development paradigm is proposed here where the software development cycle includes a grammar model-based compiler with user-centred approach. The efficacy of the proposed system is tested with the development of an innovative computer-aided drawing technology (SETUP09) for blind and visually impaired people and the results are compared with an existing non-hybrid-based drawing software (IC2D). The results of this study confirm SETUP09 improves user satisfaction and provides abstract and concrete level system flexibility. Provided here are guidelines of the proposed hybrid approach for software development based on a formal approach. This hybrid approach enables the software designer to evaluate the software semantics before user scrutinization. The benefits of the approach include the facilitation of alternate development pathways affording the software designer the flexibility to amend the software without incurring significant technical challenges. In fact, the proposed approach enables the creation of a structural design that is independent from the development pathways. This approach should ease the time constraint in product development and resource constraints.**

1. Introduction

According to Hornbæk [1], experts from different disciplines tackle a given problem using different approaches which is unique to the way they have been educated in their fields. Take for example the construction of a building. Structural engineers and architects bring to bear their own perspectives on what requirements the building needs to satisfy. From the architect’s perspective interactions between people inside the building should be central in the design of the building, whereas structural engineers consider the foundations, structure robustness and stability, and building materials. Likewise, there is distinct approach taken by software engineers to develop and implement a software product and a product intended for human computer interaction (HCI). We propose here a complementary approach in the development of a software product that considers interactions of people thus improving usability. The initial design phase does not necessarily require the involvement of end users according to Preece [2] if pertinent user information is available in the public domain. This information is likely to be obtained through surveys and stored in databases. Checking the software design using classical software engineering validation and verification tools is unlikely going to reveal issues experienced by the user and to usability. Therefore, involvement of end users is crucial in the validation of the software design. In this phase corrective action can be identified and rectified for the product is released to the client or the public. In some cases, it may be acceptable to use limited users for system testing without invalidating the usability. This may be due to the very small population size of people with certain impediments like for example blindness. According to Ait-Ameur [3] the software can be cooperatively validated and verified using formal and experimental HCI approaches.

To assure the system security, reliability and effectiveness, the use of formal methods is recommended in software development. Formal methods are mathematical approaches to specify, develop and verify software [9], [15]. Specification by formal methods overcome inadequacies of uncertainty of traditional informal methods and have precise grammar rules and semantics based on mathematical logic expressions that accurately define consistency, inclusiveness, specification and implementation.

According to Blackwell [4], typical system experimental HCI validation and performance metrics includes efficiency, the time it takes a person to complete a task, number of errors made, age of the user, experience of using similar technologies, etc. Sears [5] reports the need to know the demographics of the users and their prior knowledge and experience of using a similar software application. A controlled experiment is the preferred method commonly used to evaluate an HCI software applications where the user’s interaction with the computer interface is recorded. An automated formal verification of the software is also done using computational tree-logic and proof system. This verification method enables abstraction at any level unlike human computer interface verification which relies on graphical user interface.

Mixed approach has applied in the past for software system development. Close examination of these systems reveals limitations of the sole HCI approach. In [6] it is reported that blind users can create schematic diagrams by means of voice activation and screen touch. The development of this software was done without any end user input. Also, the software was verified using formal and informal methods. Because of no end user involvement in the design phase, the usability of the software is reported to be limited. Unlike [6] the authors in [7] took a user-centred where visually impaired children were used in the development of a game editor to support orientation and mobility training. End user involvement in the design phase included interviews, workshops, and prototype testing. Although with this approach the software developers were able to refine the product to the requirements of users however multiple iteration of the prototype was required, and the final product had limitations.

The researchers in [8] use a dialogue based interactive tool with blind users in the development of a search interface. This kind of user-centred approach enabled the users to give the designers formative feedback as well as appraise the design and propose their own suggestions. In this scenario it was necessary to discard the traditional communications with the designers using visual aids and replace it with appropriate vocabulary instead. In the conceptualisation stage the designers carefully used the standard script to avoid spurious variations in the results of the users. In fact, the users were given a full description of the pictorial scenario used in the study.

Language can be subjective and the creation of a software language for a product by software developers/designers needs to be unambiguous from the end users’ point of view to avoid usability issues. The development of a grammar model in computer science is not based on information garnered from interaction of people with computers. Instead, the creation of a software language is based on a formal approach rather than a user-centric approach. This requires that software designers take an objective approach where the grammar model is based on well-defined set of rules for its syntax and semantics. Moreover, the language needs to be amenable to changes and extension thus enabling the developer to adopt the language to variations in the system’s development.

According to Chase [9], computer aided design systems is a passive tool based on design decisions of the user, however systems based on formal grammar approach are active tools that allow software developers to innovate and rectify constraints identified by users. Moreover, formal grammar systems allow developers to investigate various software design scenarios and can accommodate new functionalities required by the user. It should be noted that development of such systems is continuing to evolve.

Both software designers as well as the end users need to have control in the application and invocation of grammar in the development of a well-managed and interactive tool. Hence in [9] users are allowed to transform design rules and sub-set of rules. The interface model in [9] has been developed to capture user interactions with various grammar models and and provided are various scenarios for grammar usage. This approach involves the users and gives them partial control in developing the grammar model and the rules. The resulting system can be updated more rapidly and reflects the actual user requirements. However, the software developer needs to ensure that erroneous results are mitigated.

In a formal approach logical notations can be used to trace transitions carried out for a particular instruction, identify erroneous or missing instructions, and thereby validate the software design. As it’s not possible to check user interaction and ergonomics using formal validation methodology, grammar-based specifications have been verified with HCI based experiments in [10], [11]. In [12] the software developer uses regenerative grammar model to check the system has applied the correct set of rules to generate a sentence that is grammatically correct. In [13] the authors apply formal models to the design process of a user interface for an assistive technology application. The advantages of using a formal approach are shown to includes consistency across multiple platforms, reachability, and completeness.

This article uses a hybrid method that is based on a formal set of abstract rules specifically created for developing software for assistive technology applications. The hybrid approach is demonstrated on an innovative assistive technology referred to as SETUP09 which was developed for blind and visually impaired (BVI) users. The proposed system comprises navigational, and computer assisted drawing technique. Screen navigation technology in SETUP09 to enable BVI users to create digital art and scientific diagrams. Accuracy and effectiveness of SETUP09 has been compared with an existing digital non-hybrid-based drawing tool (i.e., IC2D).

1. ****Hybrid Method**** for SETUP09

A formal approach was taken here in the development of an assistive technology product specifically for blind and visually impaired people. The development cycle here incorporated user and stakeholder verification. The user-centred approach in the design of the product is an iterative process that allowed the designer to focus on the user’s needs in each phase of the design process. This is important to develop a product that is highly accessible and usable by the target audience.

First a new classification diagram was developed that consisted of different branching of the proposed research. Such a diagram was needed to group grammar categories. The direction of the research for the proposed computer-aided drawing system included investigating various models including cognitive, perceptual and assistive techniques. These models underpin the categories found in secondary research related to assistive technology development in [14]. Cognitive model for exploration of space and navigation by BVI users was used to define the spatial grammar. Perceptual model of outline drawings for blind people was used to define shape grammar. Existing assistive technology models were used to define the standards, theories and frameworks of product design. Usability grammar needs to follow specific standards and regulatory requirements according to ISO standard 9241-171. In fact, formal language is constituted from a combination of space language, shape language and usability language.

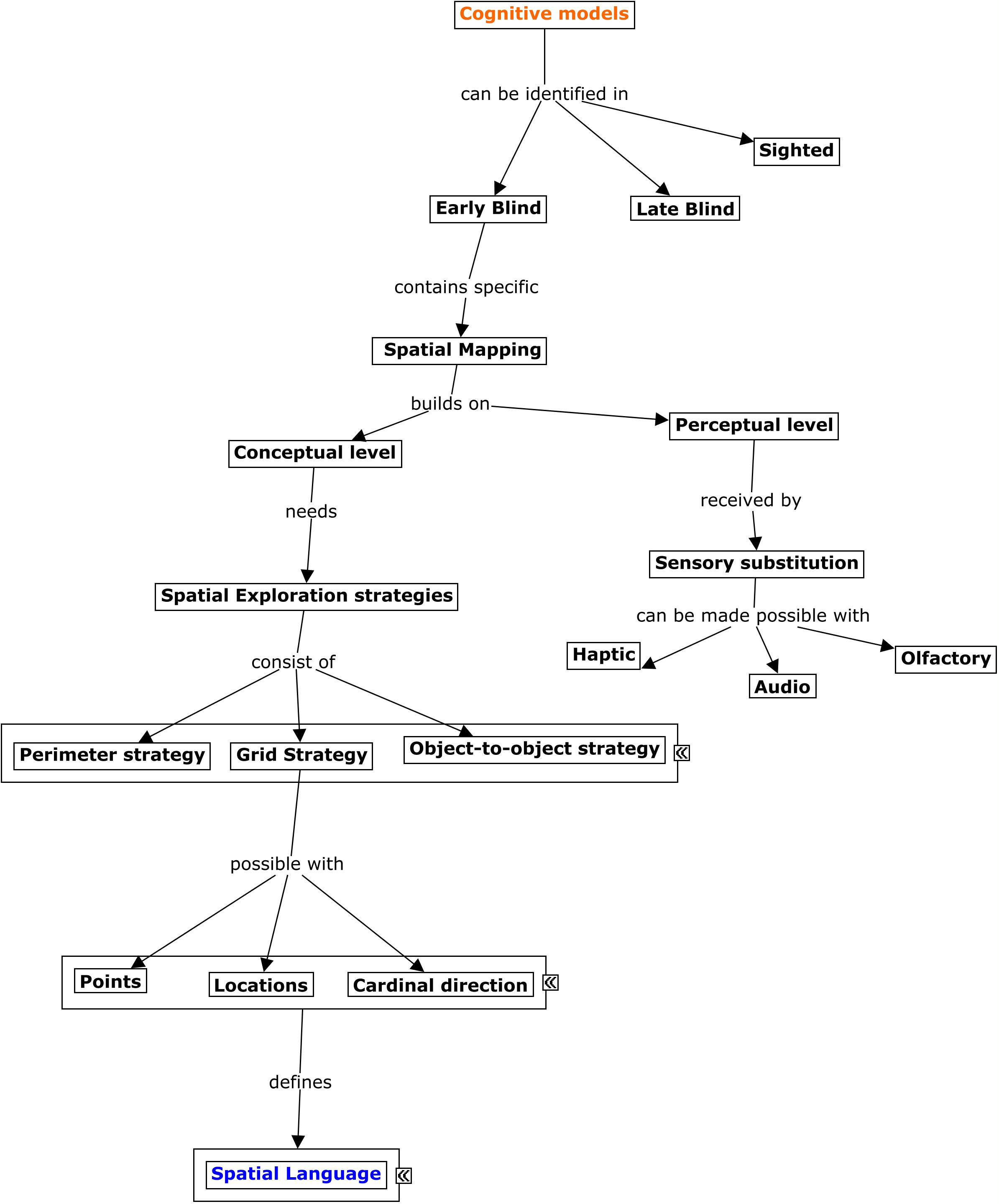
Shape language is used in art and 3D graphics to describe shapes we are familiar with. This involves identifying points on the screen and connecting the points with lines or curves to generate some specified or arbitrary shapes. This drawing can be grouped and replicated when needed. Space language facilitates navigating on a screen and to reference specific region(s) on the screen or point. This language allows the user to describe the depiction of a drawing or artwork. With the usability language the cognitive burden of users is minimized. Usability language enables users to work independently with no predilection for personal assistance. This language promotes self-learning and if required the system can provide specific guidance during an activity, and accidently deleted work can be retrieved. The proposed system passes all linguistic inputs through a lexical analyzer to retrieve tokens and then performs semantic analysis to ascertain what the token means. Semantic matching is performed by a parser function that also tackles any errors. After performing these functions, the system then generates the user defined computer aided drawing/artwork.

* 1. Spatial language considerations

Spatial language used in the proposed SETUP09 system technique stems from a classification method as depicted in Fig. 1. The proposed system is based on a cognitive model that is underpinned by behavioural data of individuals with signs of early blindness. The system exploits the established perceptual and conceptual techniques in spatial mapping. Conceptual level mapping of drawings by visually impaired users gives them the confidence to explore various strategies to visualise the object depicted on the screen. To this end the grid is important as it allows users to locate positions on the screen and thereby assuredly navigate over. Moreover, with the cognition of screen grid the user can acquire relationships between spatial locations on the screen.

1. Components of the proposed model (SETUP09)

Components constituting the proposed SETUP09 system are shown in Fig. 2. SETUP9 is developed specifically to enable blind and visually impaired people to create drawings and artwork with no external personal assistance. Here the spatial, shape and usability language components feed into a new context-free grammar model. The outputs from the grammar model and the text/speech/braille are used to recognize the input content. The recognition results are then semantically matched and validated to generate the user drawing artwork.



*Fig.1. Classification diagram of spatial grammar.*

1. Opportunities afforded by formal models

Software design that uses a formal model is not able to couple modality and concrete functionalities. However, conceptual framework can link system analysis, logic-based value judgements and different value expositions. Another conceptual framework will create a different system. Formal model is not the only way to characterize a computer aided drawing system for blind people (CADB). Alternate characterization can be implemented using non grammar based and grammar-based systems. Provided in this paper is CADB characterization that is based on an abstract model. The advantages of this type of characterizations include:

1. *Modality independence*: The system can be used with formal grammar, and it allows various formats of the input, and the resulting output is not delimited.
2. *Content and output independence:* The proposed grammar is independent of the resulting drawing and artwork presentation.
3. *Easily defined art*: The model can identify and connect arbitrary shapes and defined objects.
4. *Dynamic art generation*: The drawing/artwork is generated automatically with application of specified grammatical language by the user.
5. *Free screen navigation*: Screen movement is facilitated using grid-based screen navigation functionality.
6. *Drawing functionality*: The proposed model defines the range of actions for CADB and the availability of various drawing functionalities.

The proposed grammar-based system is intended for a specific application which is not necessarily suitable for other software applications. Some of the common impediments encountered in software product development and upgrade are limited time and budget, and lack of expertise. The proposed grammar-based model should help circumvent these issues.

Diagram

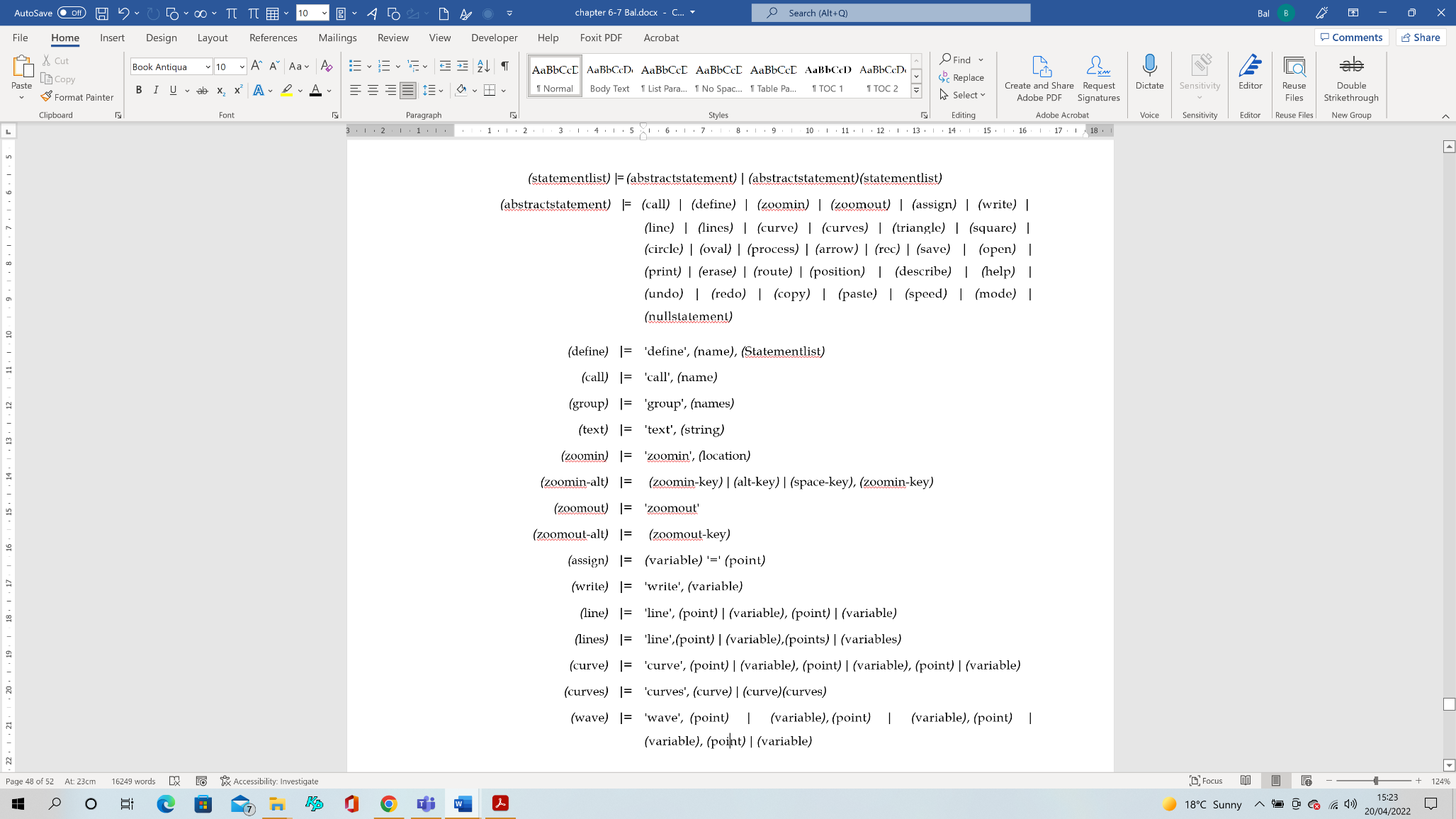
Description automatically generated

*Fig.2. Components of SETUP09 model.*

* 1. Formal specification of SETUP09

This section introduces the formal model used to develop a software language specifically intended for usage by BVI people. The language uses established modelling techniques and is based on a collection of mathematics notations addressing the requirement specification, analysis and verification. The characterization of the model described earlier was instrumental in the creation of the proposed language. Results of prior studies and Extended Backus-Naur format were used to determine the grammar to describe the space, shape, and usability. The proposed language also needed to be amenable for processing and validation by compilers and interpreters that translate source code into machine code.

A set of recursive rules are the basis of context-free grammar (CFG) that generates patterns of strings describing a programming language. Because CFG allows rapid development of computer languages it was used to develop the proposed SETUP09 software. The context-free grammar in the proposed software includes application of Extended Backus-Naur protocol and notation in the [16]. Fig. 3 and 4 show the two CFG subset models of SETUP09.



*Fig.3. Context free grammar of SETUP09 model subset #1.*

Graphical user interface, text

Description automatically generated

*Fig.4. Context free grammar of SETUP09 model subset #2.*

* 1. Backend development of system SETUP09

The software is executed by the interpreter. The interpretation was carried out in several phases. In the first phase the interpreter performed lexical analysis on the input command/instruction including identifying names, numbers, space, and marks which are transformed into words or tokens. This is followed by the syntax and semantic analysis where the interpreter determines the meaning of the instruction. Here the input text is examined, checked, translated and assigned to the required format, which is done by determining the meaning of the token that is validated for correct assignment. During the parsing phase the interpreter combines the source programme phrase and corrects issues encountered with the input language, which is based on context-free grammar. There are numerous ways the parser uses to transform the grammar including LR(k) technique (Left-to-right parse, right most derivation, k-token lookahead). Tokens generated in the lexical analysis phase are used by the parser to determine patterns and executes appropriate actions defined by the syntax of the language. Once the code has been parsed and a clean abstract syntax tree (AST) has been generated the user’s input is checked to sure it makes sense. This requires performing validation to identify semantical errors. In the final phase the interpreter executes the input command/instruction. In the case here the output generated is a piece of drawing/artwork created by the BVI user. Text commands input by the BVI user were transferred to the abstract syntax tree. The text inputs were then verified with abstract syntax tree and rules of concrete syntax [17]. Checked here were the language’s structure and commands and the resulting output.

* 1. System SETUP09 frontend development

The development of the proposed SETUP09 system and its interface involved the software developer researchers (sighted personal), an undergraduate student who was blind as well as ten additional blind individuals of various ages. The blind student was involved over the development cycle of the system. Similar process was used by researchers in [8]. The steps involved in the system development included:

Step #1 *–* Auditory interface was developed to navigate the blind user on the screen. It enables the user to create, track and identify drawing objects and features. The usability and effectiveness of the interface was confirmed by the blind student who had experience in playing interactive computer games.

Step #2 *–* Features of the interface and their functionalities was created. Sufficient features were created to allow the blinder users to closely represent their drawing/artwork scenario. Involvement of the blind participant was essential to ensure the proposed system commands were suitable.

Step #3 *–* Informal discussion with the blind users helped the researchers understand and appreciate the issues encountered by the users. This was important to refine the system’s features and thereby improving its usability. In fact, the users helped develop the option to interact with the system including shortcut keys to save, copy, paste and open file.

Step #4 *–* The system’s usability and functionality were validated with the blind users who were given a verbal description an object and scene which they had to recreate using the features provided in the systems toolbox.

Step #5 - Formative evaluation of the proposed system was carried out by the system developers with the target end users. The system developers described a relatively complex scene giving step by step instructions to the users who had to recreate the scene using the proposed system. The users were able to obtain feedback at the end of each step.

Based on the user feedback on usability and system functionalities three versions of the prototype system were developed. The first version based on the initial concept divided the computer screen into nine navigational sectors. The second version used knowledge-based rules and commands for drawing. The last version included language interpreter.

* 1. The system SETUP09

In Fig. 5 shows a flowchart drawing created by a blind user. The flowchart begins with a start block followed by a process bock and concludes with an end block. Also shown are the braille letters corresponding to the block symbols. The flowchart image was created using a command language in section 2.4.

Diagram

Description automatically generated

*Fig.5. Flow Chart produced using SETUP09 system.*

Shown in Fig. 6 is a blind participant using the proposed system (SETUP09). Fig. 7 shows a screen shoot image created by a blind user. The very simple 2D image created is of a house with a door and window surrounded by a fence. The image also includes a stick figure and skyline. The drawing was constructed using basic function tools in the system of lines, curves and circles. The user commands, which are not shown here, are made visible at the bottom of the screen panel.



*Fig.6. A blind participant using SETUP09 system.*

Diagram

Description automatically generated

*Fig.7. A scenery of a home produced using SETUP09 system.*

1. Proposed Hybrid Approach

In the context of human computer interaction system, the researchers in [22]-[24] have used user-centred design process where end-users are required to actively engage in the design process. However, in the context of computer-aided drawing systems for blind users there is no universally agreed design methodology that involves BVI users. Proposed here for the first time to the knowledge of the authors is a new computer aided drawing process for blind users that uses a formal approach. This section provides the steps that should be taken in the hybrid approach for CADB. The steps given are not limited in the development of SETUP09 system but are generic and can be applied in the implementation of any grammar-based system development.

* 1. Process flow of the hybrid approach

1. Guidelines to classify system requirements

• System requirements can be either user defined and/or research defined. Requirements that are user defined, are features the users need the software product to support. Whereas researcher defined features the software should support have been acquired from experimental studies. All these technical requirements must be first categorised in pre-defined guidelines that are clear and concise for the system developers. The features relevant to SETUP09 were usability, space and shape guidelines. These guidelines are important for software developers to optimize the code, verify and validate the application.

1. Translate system requirements into abstract grammar rules

• System requirements need to be converted to abstract grammar rules that describe a language. This was achieved with a formal approach that uses mathematical models to the build software. The formal approach provides a framework that is used to specify, develop, and verify the system in a systemic, rather than ad hoc manner. In this step the high-level system requirements are converted into mathematical notations. With this approach any repetition and inaccuracies are eliminated.

1. Grammar rule refinement

• Formal specification grammar rules are well-defined and do not stand in isolation. There are multiple methods in which to execute grammar. Some grammar rules refer to other grammar rules before a primary rule is executed. The grammar rules can be resolved in a complex or simple manner. Grammar rule refinement facilitates the elimination of complex grammar parsing, maintains accuracy of the specification that is verified using a parsing tree.

1. Grammar implementation using concrete syntax

• A high-level programming language is required in the implementation of specification grammar which is used to initiate user interaction with the system. The selected language data structures enable the execution of system actions instituted by the user. In addition, grammar implementation necessitates lexical analysis, semantic analysis and parsing which operate with implementation with concrete syntax.

1. Grammar check

• The proposed language should carry out meaningful actions by resolving multitude of the defined grammar rules. The system interface and predefined commands should be used to verify the language on the concrete system. This strategy allows verification of even more complex language command executions. An abstract syntax tree representing the constructs in the language and their subsequent rules will enable the identification of inconsistencies in the language.

1. Requirement test

• The system requirements and the proposed grammar needs to be verified with end-users for consistency and to identify any glitches and errors that will need to be rectified. This is essential to ascertain whether the system satisfies its specifications by comparing the actual results with the expected results. The grammar test is necessary to verify the accuracy of the mathematical and structural integrity. The purpose of these two tests is to ensure the system and grammar executes the commands of the user as specified. Ensuring quality through the development process ensures a robust system.

1. Implementation modification

• As a result of the end user testing a formal analysis of the system specification needs to be instituted to revise and update the grammar rules. This action is necessary for the proposed model to meet user requirements and ensure system operates consistently. Additional grammar (terminals, non-terminals, or tokens) and/or changes to the existing grammar can also be introduced at this stage. This stage can be repeated until the system designer is fully satisfied it functions as required. Also, at this stage the system and/or interface updates can be carried out.

* 1. Process flow diagram of the Hybrid Approach

Fig. 11 shows a new architectural model of the hybrid approach flow used in this study. The model provides a guideline on implementation, improvement, and integration for designing a command-driven computer-aided drawing system for blind users. The first stage of the process involves identifying user requirements, classifying them, and creating a refined set of abstract rules. In the second stage the grammar is implemented using abstract or concrete tools. In the third stage the system requirements are verified and if necessary modified.



*Fig.11. Process flow diagram of the Hybrid Approach.*

End user feedback was used to refine the grammar of the proposed SETUP09 model. In this study the final SETUP09 model incorporated new and revised Extended Backus-Naur grammar. The grammar changes were categorised into space, shape, and usability revisions. Not all changes were grammar-based as users paid particular attention to the ergonomics of the system interface as it played a role the user experience of the system.

1. Systems Testing: SETUP09 Hybrid Approach System versus IC2D User Centred Approach System

Pertinent questions that were ask about the proposed SETUP09 software application included: “Will it satisfy the requirements of BVI users?”; “To what extent will users find it easy and joyful to use?”; and “Will target users be able to easily create high quality drawings/artwork? These questions were evaluated with various groups of blind and visually impaired groups of end users. In previous studies [14], the efficiency and the effectiveness of the language use to navigate SETUP09 was evaluated.

Fig. 8 shows the breakdown of the main areas involved in the evaluation of cognition-of-navigation and cognition-of-drawing using SETUP09 by BVI and sighted users.

Diagram

Description automatically generated

*Fig.8. Main areas to evaluate SETUP09 tests.*

* 1. Why is system (IC2D) compared with SETUP09?

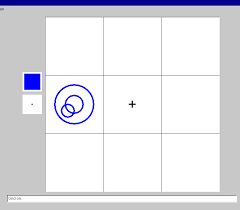
IC2D system was the closest equivalent system to SETUP09 even though it was introduced some time ago. IC2D has the following features: (1) Uses grid-based screen navigation, and (2) designed and develop for blind users for digital drawings on the computer. IC2D was therefore used in the benchmark comparison of SETUP09. The data of IC2D was extracted from the results reported in [18]. It should be noted that there is a difference in 2D shape placement between IC2D and SETUP09. IC2D has a palette of 2D objects for users to select shapes such as circles, squares and triangles. In IC2D, the objects are placed at the centre point of the grid whereas in SETUP09 grid outline is used. Also, SETUP09 has object placement functionality for grid outlines and centre point. In one task, the participants had to reproduce the square inside the gridline, and another task involved the placement of a square on top of gridlines. SETUP90 has more commands for finding grid points inside the cell than on the gridlines.

* 1. Comparison of SETUP09 hybrid-based system development with similar non-hybrid-based systems (IC2D)

The usefulness of the 2D drawing (IC2D) product [17] was originally studied with mixed participants including blind and sighted. These participants were given tasks to draw an arbitrary object under instructed and non-instructed conditions. The participants had to create two dimensional and three-dimensional images. The drawing capability of SETUP09 was compared with the well-established IC2D technology under identical experimental conditions. The participants were given two tasks, namely:

**Task 1 –** In this task the participants were instructed to draw circles inside any grid location on the SETUP09 screen. An example of this is shown in Fig. 9. For each attempt the participants had to explain the grid location of where on the screen they were attempting to draw the circle. This was for the purpose of establishing the participant’s original intention. This information was later used to compare with the actual location of their attempt. The aim of this task was to establish whether the participants were able to articulate their individual non-instructed drawings.

**Task 2 –** In this task the participants were instructed to (i) draw a small triangle in the bottom righthand grid of the screen, (ii) draw a rectangular shape inside the top left grid of the screen, and (iii) draw a straight line that connects the top vertex of the triangle to the top left vertex of the rectangle, as shown in Fig. 10. The aim of this task was to establish whether the participant was able to carry out instructed drawing with accuracy.



*Fig.9. Task #1 drawing.*

Chart, line chart

Description automatically generated

*Fig.10. Task #2 drawing.*

*Table 1. Information on Blind and Sighted Participants*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Sex** | **Age** | **Reason** | **Residual Vision** | **Age at Blindness** | **Education** | **Computer Literacy** |
| 1 | M | 71 | Microphthalmia | None | Birth | Graduate | High |
| 2 | F | 66 | Coloboma | None | Birth | Graduate | High |
| 3 | F | 38 | Microphthalmia | None | Birth | Manager | High |
| 4 | F | 57 | Optic Nerve Damage | None | Birth | Graduate | Moderate |
| 5 | F | 37 | - | Sighted | - | Graduate | High |
| 6 | F | 36 | - | Sighted | - | Graduate | High |
| 7 | F | 52 | - | Sighted | - | Graduate | Low |
| 8 | M | 56 | - | Sighted | - | Graduate | High |

* 1. Procedure

The experiment in this study was designed and conducted during the social isolation period in 2020 due to Covid-19 global pandemic. This was done with the approval of both the university’s research ethics committee and the participants. To minimise the risk of catching Covid-19, the experiment was conducted on an individual basis with the permission of the participants at their residence. The participant observed social distancing, wore personal protection equipment (face mask), used hand sanitiser whenever appropriate, and all surfaces we were in contact with were cleaned with antibacterial wipes including the computer/laptop. At the location of the experiment the only people present were the participant and the observer.

Kamel in [18] tested the IC2D system with blind and sight participants; hence the proposed SETUP09 software was tested with the same number of people that included four blind and four sighted people. The experiment involved the participant in drawing a simple piece of artwork/diagram on a computer/laptop using SETUP09 with the given instructions. The observer took several HCI measurements for analysis. The parameters recorded during the experiment included:

* Task time
* Participants confidence
* Performance rating

Blind participants for the study were recruited through the Beyond Sight Loss Charity, UK. The blind participants were predominantly diagnosed with Microphthalmia, Coloboma and Optic nerve damage. All participants (blind and sighted) had an average age of 50 years and standard deviation of 15.04. Their ages ranged between 30 to 71 years as given in Table 1. Instructions for the experiment were circulated prior to the study so that participants were familiar with what was required from them. Participants were explained how to take control of their computer/laptop screen, and the keyboard commands they would be using in the study. Prior to commencing the study each participant was given roughly 30 minutes to practice using the keyboard commands of SETUP09 and get familiar with the software. In the study the eight participants were given at least two trials. There were in total 16 trials recorded. At the end of the experiment, participants’ feedback was taken for further analysis. Tasks 1 and 2 illustrated in Fig. 9 and 10 were carried out by the participants using object (shape) placement technique in SETUP09. This involved the placing the objects on top of gridlines. In task 1, participants were asked to draw three circles in three different grid positions of their own choosing and then draw the smallest triangle possible in the bottom right cell of the grid. They were then asked to draw a rectangle inside the top left cell, and finally draw a line connecting the top vertex of the triangle with the top left vertex of the rectangle in Task 2.

To draw the image in Fig. 9, the SETUP09 system needs input commands from text/speech or braille as follows: **[*W, circle NW, C, circle, NW, zoomout, W, SE, circle NE*]**. To draw the image in Fig. 10, the SETUP09 system needs input commands from text/speech or braille as follows: **[*SE, C, C, triangle W N E, assign 1 N, zoomout, zoomout, zoomout, NW, line 1 NW, square NW NE SE SW*]**. In IC2D system participants could only use menu driven options.

* + 1. *Experiment results*

*Table 2. Quantitative Study Results*

|  |  |  |
| --- | --- | --- |
| Measurements | SETUP09  System | IC2D  System |
| Task completion time | 3.35 (m:ss) | 7.91 (m:ss) |
| Confidence level (1-10) | 7.42 | 7.48 |
| Performance rate (1-10) | 9.4 | 8.2 |

*Table 3. Qualitative Study Results*

|  |  |
| --- | --- |
| Type of participants | Feedback |
| VI | 1. I enjoy the actual metal challenge.  2. Easy and clear, able to visualise. I forgot what keys to navigate.  3. You have to carefully listen to system commands.  4. The more I use, the more I know.  5. Can the system auto correct my commands. |
| Sighted | 1. Sighted custom voice is needed.  2. Can system tell, "your drawing is completed."  3. Last one was hard; you need to know and remember the commands. |

• The amount of time taken by the group of BVI and sighted participants to complete both tasks using SETUP09 was compared with a commercial drawing package for visually impaired users called IC2D [17]. The average time to complete the two tasks using SETUP09 was 3 minutes and 35 seconds. However, with IC2D the average time taken by the same participants was 8 minutes and 32 seconds. These results are very encouraging as the proposed drawing system took the users almost a third of the time as IC2D. This is mainly ascribed to two reasons. Firstly, the proposed SETUP09 system is command-driven whereas IC2D has a menu-driven input. It was observed that once the users got familiar with the commands, they could execute the drawing with just a few keystrokes. Secondly, unlike IC2D the SETUP09 system uses multi-point reference grid lines that allow the user to place drawing elements on the screen by aligning them with the grid lines or nearest intersection of the grid lines. Also, the grid lines are generated dynamically as the users zoom in or out. It was found that the multi-point reference grid permits the users to use few navigation steps.

• It was important to ascertain the confidence level of the participants for using the proposed and commercial drawing packages. The participants were asked to score between 0-10 the answers to a set of questions where a score of 10 represented the highest confidence level. The average confidence level score for SETUP09 was 7.42 and with IC2D was 7.48. This result was contrary to we were anticipating. We were expecting a higher score for SETUP09. This was because unlike IC2D with SETUP09 the participants had to remember and recall the user commands. This had a direct impact on the confidence score. Nevertheless, the critique from the participants was useful. In fact, the participants said that with more usage of SETUP09 they would get much more familiar with the user commands that would have significant impact on the confidence level.

• The accuracy of performing the two tasks by the participants was scored between 0-10. From the two tasks the more challenging task was the second. It was observed that although a few participants made less errors in carrying out the second task however their drawing was less accurate. The empirical results revealed that for the two tasks the participants performed the drawings more accurately with SETUP09 than with IC2D. In fact, the average accuracy of carrying out the tasks with SETUP09 was 9.4 and with IC2D it was 8.2.

• The empirical results demonstrate that with IC2D system the BVI participants accomplished the tasks marginally better than with SETUP09 while participants who were sighted achieved better results with SETUP09. This does not imply that IC2D is more suited for BVI users than SETUP09 because statistical analysis shows there is no significant disparity in the outcomes of BVI and sighted participants. In fact, the outcome is determined on how the experiment is carried out as reported in studies in [19] and [20] where sighted people took less time than blind people to carry out tasks involving image scanning and spatial identification. Sighted users in [17] were blindfolded when using IC2D. Under this condition some users would have been disoriented and found using the keyboard challenging. This would impact on the speed of keyboard usage. The keyboard was concealed from the sighted user in the SETUP09 study. Fuller analysis of the study is presented in [14].

* + 1. *Experimental limitations*

• The empirical study was carried out at a time when there was a Covid-19 global pandemic, and the size of the sample (participants) was restricted by the university’s research ethics committee. The size of the sample in the SETUP09 study was therefore smaller than that reported for IC2D in [17]. Nonetheless the comparison analysis clearly discloses the trend in the results.

• Details of the original study using IC2D is reported in [17]. The conditions in which the study using IC2D was conducted are different to nowadays because of the lower processing speed of the computers then, and the varying levels of computer and IT literacy/competency skills of the participants. These factors should be considered when comparing the results in [17].

• The observations made in the study conducted here shows that the ability by sighted people to recognise geometrical shapes did not give them competitive advantage over partially sighted, early and late blind people. Even though the sample size in the study here was relatively small than that in [17] nonetheless the outcome of the results from SETUP09 are reassuring and substantiate that SETUP09 permits people afflicted with BVI conditions to create drawings with confidence as sighted individuals. Not evaluated in the study the ability of the participants on identifying geometrical shapes, their memory and computer literacy skills.

• In this study on SETUP09 the experimental tasks were not carried out by the same individuals. This discounts an exact comparison. However, application of the tools in this study was carried out under similar experimental conditions (insofar as was realistic) as that in [17]. Comparison of the SETUP09 results with those reported in [17] may result in some discrepancy due to factors such as the computer and IT literacy of the participants and their acquaintance with equipment used. This may affect the measurements of completion time of tasks, errors encountered, confidence level, and satisfaction rate. The effect of such factors on such studies was also observed in [21].

1. Conclusion and Future Work

Validated in the study is a SETUP09 System built based on hybrid approach for software development which is pertinent in the field of HCI. The hybrid approach presented here is a combination of the formal and user centric approaches. The proposed hybrid approach was implemented, tested and compared with existing IC2D approach. The proposed approach was employed in the design and development of SETUP09 model which is based on a set of formal grammar rules that was obtained from empirical studies. According to the authors’ knowledge this approach has not been done before in the assistive technology domain. From the experimental findings and comparison with IC2D, it is concluded that even though the SETUP09 prototype is not refined, it is nonetheless capable of effectively navigating the BVI user on the computer screen and enabling them to create 2D drawings/artworks with no intervention from an assistant. It should be noted that the formal approach is not by itself a complete solution to the main issue concerned here with design and implementation of a computer-aided drawing system, but it adds significant value. The proposed hybrid approach has the advantage of (1) modality independence where different forms of input and output are not restricted by the solution and can be coupled and used with a formal grammar, and (2) content and presentation independence where the grammar allows independent creation of system design pathways and independent access to those pathways.

1. References

[1] Hornbæk, K. (2011). Some Whys and Hows of Experiments in Human–Computer Interaction. Human–Computer Interaction, 5(4):299–373.

[2] Preece, J., Sharp, H., and Rogers, Y. (2015). Interaction Design: Beyond Human-Computer Interaction. John Wiley and Sons Ltd.

[3] Ait-Ameur, Y. and Baron, M. (2006). Formal and experimental validation approaches in HCI systems design based on a shared event B model. International Journal of Software Tools Technology Transfer, 8(547–563).

[4] Blackwell, A. (2009-2011). Human Computer Interaction. Technical report, University of Cambridge Computer Laboratory, Computer Laboratory, Williams Gates Building, J J, Thomson Avenue, Cambridge CB3 0FD.

[5] Sears, A. and Hanson, V. (2011). Representing users in accessibility research. In ACM, editor, CHI, vol.978.

[6] Blenkhorn, P. and Evans, D. G. (1998). Using speech and touch to enable blind people to access schematic diagrams. Journal of Network and Computer Applications, 21(1):17–29.

[7] Mattheiss, E., Regal, G., Sellitsch, D., and Tscheligi, M. (2017). User-centred design with visually impaired pupils: A case study of a game editor for orientation and mobility training. International Journal of Child-Computer Interaction, (11):12–18.

[8] Sahib, N. G., Stockman, T., Tombros, A., and Metatla, O. (2013). Participatory design with blind users: A scenario-based approach. IFIP International Federation for Information Processing, part I (8117): 685–701.

[9] Chase, S. C. (2002). A model for user interaction in grammar-based design systems. Automation in Construction, 11:161–172.

[10] D’Ulizia, A., Ferri, F., and Grifoni, P. (2007). A Hybrid Grammar-Based Approach to Multimodal Languages Specification. In OTM Confederated International Conferences on the Move to Meaningful Internet Systems, 4805: 367–376.

[11] Dabek, F. and Caban, J. J. (2017). A grammar-based approach for modelling user interactions and Generating Suggestions During the Data Exploration Process. IEEE Transaction on Visualization and Computer Graphics, 23(1).

[12] Amin, P. W. O. (2019). Models of Generative Grammar. Technical report, Ibn Rushd College of Education University of Baghdad.

[13] Bowen, J. and Reeves, S. (2007). Using formal models to design user interfaces a case study. In People and Computers XXI – HCI. British Computer Society.

[14][Fernando, Sandra](https://research.gold.ac.uk/view/goldsmiths/Fernando=3ASandra=3A=3A.html). 2021. [A Formal Approach to Computer Aided 2D Graphical Design for Blind People.](https://research.gold.ac.uk/id/eprint/30382/)Doctoral thesis, Goldsmiths, University of London [Thesis]

[15] Cook, J. (1992). Formal methods: What? why? and when? The Computer Journal, 35(5).

[16] Norvell, T. (2002). A short introduction to regular expressions and context-free grammars. http://www.engr.mun.ca/theo/Courses/fm/pub/context-free.pdf.

[17] Appel, A. W. (2002). Modern Compiler Implementation in Java. Cambridge University, Press, New York, NY, USA, 2nd edition.

# [18] Kamel, H. M. and Landay, J. A. (2002). Sketching images eyes-free: a grid-based dynamic drawing tool for the blind. [Proceedings of the fifth international ACM conference on Assistive](https://dl.acm.org/doi/proceedings/10.1145/638249). 33-40.

[19] Kerr, N. H. (1983). The role of vision in “visual imagery” experiments: Evidence from the congenitally blind. Journal of Experimental Psychology: General, 112(265-277).

[20] Hill, E. W., Rieser, J. J., Hill, M.-M., and et al, M. H. (1993). How persons with visual impairments explore novel spaces: Strategies of good and poor performers. Journal of Visual Impairment and Blindness, (87):295–301.

[21] Bornschein, J., Bornschein, D., and Weber, G. Comparing computerbased drawing methods for blind people with real-time tactile feedback. In CHI.

[22] Mitchell, V., Ross, T., Sims, R., and Parker, C. J. (2015). Empirical investigation of the impact of using co-design methods when generating proposals for sustainable travel solutions. CoDesign, 12(4):205–220.

[23] Wilkinson, C. R. (2011). Evaluating the Role of Prior Experience in Inclusive Design. PhD thesis, Cambridge University Engineering Department.

[24] Luck, R. (2003). Design studies. Elsevier, 24:523–535.