Making graph-based diagrams work in sound: the role of annotation

Andy Brown, Robert Stevens and Steve Pettifer
University of Manchester
Oxford Road
Manchester. M13 9PL
UK
[andrew.brown-3 | robert.stevens | steve.pettifer]@manchester.ac.uk

Abstract

Non-linear forms of diagrammatic presentation, such as node-arc graphs, are a powerful and elegant means of visual information presentation. While providing non-visual access is now routine for many forms of linear information, it becomes more difficult as the structure of the information becomes increasingly non-linear. An understanding of the ways in which graphs benefit sighted people, based on experiments and the literature, together with the difficulties encountered when exploring graphs non-visually, helps form a solution for non-visual access to graphs. This paper proposes that differing types of annotation offer a powerful and flexible technique for transferring the benefits of graph based diagrams, as well as for reducing disorientation while moving around the graph and for tackling some of the inherent disadvantages of using sound. Different forms of annotation that may address these problems are explored, classified and evaluated, including notes designed to summarise and to aid node differentiation. Graph annotation may be performed automatically, creating a graph that evaluation shows requires less mental effort to explore, and on which tasks can be achieved more effectively and more efficiently.

1 Introduction

This paper builds on the body of accessibility research by examining the problem of the accessibility of graphs. We propose that graphs may be automatically annotated with information that will improve the ease with which screen reader users will be able to use and understand them. Our approach is first to analyse and understand the benefits diagrams afford sighted users; here the literature from cognitive science, psychology, information management and computer science are collated, to identify these
benefits and understand the causes of difficulties encountered when browsing non-visually. This analysis suggests that many of the benefits could be recreated (and the difficulties reduced) in audio using relatively simple, algorithmically generated, annotations. In this paper we present an analysis of the problem, leading to a taxonomy of annotations that reflects the requirements of audio users. Finally, we present some examples of annotations, discussing how each could help the user.

If visually impaired people\(^1\) are to take a full role in society, they must have access to information. This is neither an insubstantial problem, nor applicable only to a tiny number of people: an estimated 450,000 people in the UK (Department of Health, 2003; RNIB, 2006) and 10 million people in the United States (American Foundation for the Blind, 2008) are visually impaired. An added impetus for enabling accessibility is legislation, such as the Disability Discrimination Act 1995 and the Special Educational Needs and Disability Act of 2001 in the UK, and section 508 of the Rehabilitation Act in the US. As a final note on the importance of accessibility, there are marked similarities between the difficulties encountered by visually impaired users and those encountered by sighted users in certain circumstances (e.g., Harper and Patel (2005) explored the use of Web link summaries to aid both small screen device users and visually impaired users).

The standard means of computer access for a blind user in the UK is the screen reader (Alliance for Technology Access, 2000; Kirkpatrick, Rutter, Heilmann, Thatcher, and Waddell, 2006; Chen, 2006; Ramen, 2008): software that reads text on the screen (including document text, button labels, menus and other software controls) and presents it as speech via a speech synthesiser (hardware or software — nowadays this is often incorporated into the screen-reading software). While giving a level of control over the flow of information, particularly with linearly organised information such as simple text documents, screen readers are far from perfect. One particular problem is that navigating non-linear information (such as tables or equations) is difficult.

The type of non-linear information that has received the most attention from accessibility researchers is hypertext (Petrie, Morley, McNally, O’Neill, and Majoe, 1997; Yesilada, Harper, Goble, and Stevens, 2004a), with the prize of efficient Web access as the main driver. These systems are largely text-based, in that the raw information is plain text, but the chunks of text are organised in a more complex way than a simple linear narrative. Giving visually impaired users an experience as rich and efficient as sighted users is a difficult task that, arguably, has not yet been achieved. In other

\(^1\)We use the UK terminology, where a person is considered blind if he or she has a visual acuity (as measured for example using a Snellen letter chart) of less than 0.05; they are considered to have low vision (i.e., be partially sighted) if their visual acuity is greater than 0.05 but less than 0.3 (International Council of Ophthalmology, 2002). Visually impaired is a term that covers people who are either blind or partially sighted.
domains, tools for accessing mathematical equations (Stevens, Edwards, and Harling, 1997) have been researched and developed; here the non-linearity is present at a much smaller granularity making this probably the most closely related to graphs. Tables (Brown, Brewster, Ramloll, Burton, and Riedel, 2003; Yesilada, Stevens, Goble, and Hussein, 2004b) and numerical graphs (Yu and Brewster, 2003; McGookin, Kildal, and Brewster, 2005) are also being investigated. In fact, most types of visual information have been the subject of research into making them accessible to the blind, including graphs (Petrie et al., 2002; Horstmann et al., 2004) and photographs (Roth, Richoz, Petrucci, and Pun, 2001).

The graph is, mathematically speaking, a set of nodes connected by arcs. Graphs form a common and important class of diagram that is particularly suitable for depicting relationships. For example, consider railway diagrams such as the London Underground map, flowcharts, hierarchies, trees (e.g., family trees) — all of these might be encountered in everyday life. In technical and educational environments graphs are even more common, almost ubiquitous. This research is limited to exploring accessibility of information that is (or can be) presented as a graph. We are concerned only with diagrams that are graphs in the strict mathematical sense, i.e., graphs where the positioning of the nodes is for presentation and does not affect the semantics.

Presentation of diagrams to blind people obviously requires the information to be conveyed either through touch or sound. Devices are available to present information through haptic interfaces (including both tactile and force-feedback interfaces), and these have been utilised by many research groups (for example, Kennel, 1996; Blenkhorn and Evans, 1998; Rotard, Knoedler, and Ertl, 2005; Wall and Brewster, 2006; Yu, Kuber, Murphy, Strain, and McAllister, 2006). The equipment required for these interfaces is, however, specialist, and often expensive. Even Braille and raised-paper diagrams are not ideal, requiring specialist hardware to generate them, and being unable to respond quickly to changes in the data. Braille also needs to be learnt, and is by no means universally known amongst blind people (estimates vary, but are thought to be well below 20% of visually impaired in the UK and US (Braille Institute of America, 2010; Schroeder, 1994)). For these reasons, this research concentrates on keyboard input and audio output. Many of the techniques developed for exploring graphs through such an interface may well be applied to other, more specialist interfaces.

There have already been several research projects that have studied non-visual accessibility for graphs (for example, Metatla et al. (2008); Bennett (2002); Blenkhorn and Evans (1998); Kennel (1996)), and more that deal with a specific class of graph, such as the hierarchy (Brewster, Raty, and Kortekangas, 1996; Brewster, Capriotti, and Hall, 1998; Smith, Francioni, Anwar, Cook, Hossain, and Rahman, 2004). The approaches taken, and conclusions drawn, by some of these are discussed in Section 3. These solu-
tions are specific, typically describing a system allowing users to explore a particular form of graph, but graphs are a very common form of diagram, and appear in many different domains. What is needed is a generic solution: a set of techniques that can be applied to the wide range of graphs that people encounter. Such a solution necessarily starts with an understanding of how and why graph-based diagrams work: what is it that makes them better than other forms of presentation? Only with this understanding can the requirements for non-visual solutions be identified. Unfortunately, the design of tools so far appears to be largely based on intuition, with little analysis of the underlying cognitive science.

By using a sound basis of theory upon which to base our development, we aim to take a rigorous approach, understanding the benefits of diagrammatic representations and the problems associated with non-visual browsing, before developing techniques that should afford visually impaired users some of these benefits and alleviate the problems, and testing these experimentally. This paper proposes annotation as a powerful, flexible, and generic technique, and sets out to justify and test the following hypothesis:

Annotations can be designed to replace some of the benefits imparted by visual presentation of graph-based diagrams, including making implicit information explicit, grouping related items, interactivity and acting as an external memory, and to reduce disorientation while moving around the graph. A graph which is annotated in such a way requires less mental effort for a visually-impaired user to explore than one which is not. Tasks can be achieved more effectively, efficiently and with more satisfaction through use of annotation to replace features of a visual presentation.

Furthermore, we propose that these annotations need not be rich in order to help: even relatively simple notes that can be automatically generated have the potential to benefit users.

2 Theoretical Considerations

As an example of a graph, consider Figure 1, which shows the structure of the molecule ethanoic acid. In this type of graph the nodes represent atoms and the edges the bonds between atoms.

Finding an effective means of non-visual presentation of diagrams such as this (and particularly more complex graphs) requires a thorough understanding of the nature of the problem. We suggest that the problem can be characterised as finding answers to the following questions, and, from them, identifying a set of requirements.
2.1 The benefits of diagrammatic representation

The popular belief that diagrams are a very efficient way of presenting information was investigated by Larkin and Simon (1987), who found that the 2D indexing of the information in diagrams can support extremely useful and efficient computational processes. By comparing the computation required for problem solving using diagrammatic representations with that required when using equivalent sentential representations (sequential representations, like propositions in a text), they concluded that diagrams facilitated problem solving by easing search and recognition. Taking these important conclusions with those from other theoretical studies enables identification of the attributes of diagrams that make them useful that need to be taken into account when considering non-visual interaction:

1. Diagrams aid recognition of information; that which would be implicit in some representations often becomes explicit when presented as a diagram.

2. Diagrams facilitate searching by using 2D indexing, allowing related nodes to be easily identified as such.

3. The external diagrammatic representation facilitates external cognition — in particular memory demands are lower and error-making constrained (Scaife and Rogers, 1996).
4. Palmer’s (1977) model of perception suggests that building the data into a hierarchical structure might allow processes to perform in as similar a way as possible to visual perception.

5. Miller’s (1956) idea of chunking in short-term memory can be related to both hierarchies and recognition, and will be an important factor in memory-intensive problem solving.

6. Problem solving is thought to occur by manipulation of a mental model (Johnson-Laird, 2004); integrating information to build this model is often the most demanding part of the process (Goodwin and Johnson-Laird, 2005).

2.2 Presenting information aurally

Any technique that aims to enable effective non-visual accessibility of graphs needs to support the processes of mental model formation and manipulation. To do this aurally, however, requires an understanding of the differences between the presentation of information in the visual and aural modes. Perhaps the most striking of these is that all parts of the visual form are seemingly instantly accessible. Crucially, any part of the diagram may be visited or revisited in an instant (although note that finding the part of interest may not be trivial), enabling it to act as an external memory. This has three significant benefits for readers of diagrams:

1. It reduces demands on short-term memory;

2. It is possible to gain an overview quickly, and;

3. It facilitates external cognition.

Reduced short-term memory demands and external cognition help by freeing up mental capacity for problems more directly related to the task in hand. Similarly, preview summaries have been shown to be beneficial to various groups (Holmqvist, Holsanova, Barthelson, and Lundqvist, 2003; Graves, Cooke, and Laberge, 1983; Neuman, Burden, and Holden, 1990), since knowledge gained from a preview allows more processing capacity to be applied to obtaining information from other knowledge sources (Stanovich, 1980). While loss of these benefits will cause difficulties in reading the graph and building the mental model, they will be compounded if the benefits of visual diagrams are not also present. For example, the lack of external memory is particularly significant if the reader is unable to recognise and chunk features of the graph.
2.3 Mental representation of graphs

Reading the diagram can be considered a journey through information space, similar to the idea of navigating hypertext (Bernstein, 1988). Tasks require understanding one’s position (orientation), and moving from one location to another (navigation). While the applicability of this metaphor has been questioned (Boechler, 2001), it is still used within the hypertext community (e.g., Sorrows and Hirtle, 1999; Yesilada, Stevens, and Goble, 2003), and in information visualisation, where, for example Ingram and Benford (1996) applied the urban landscape work of Lynch (1960) to improve graphical data visualisations. We contend that it can be useful here, sitting neatly with the mental model explanation of reasoning: together these suggest that the key requirement is to facilitate model building (map making).

Mental models of spatial environments are acquired via three phases (Siegel and White, 1975): landmark recognition, landmark coordination, and survey knowledge formation — although these do not necessarily occur in series (Hirtle and Hudson, 1991). Taylor and Tversky (1992) found that the structure of a model formed from a description does not appear to be influenced by the perspective of that description. Considering the detail of these phases, two key features can be identified from Allen’s (1999) framework for wayfinding: the importance of object recognition, and the use of landmarks in all tasks. Landmarks act as familiar points that mark a route, either as confirmation or indicating a change of direction (Michon and Denis, 2001). Distant landmarks can also provide orientation and directional clues (Raubal and Winter, 2002). Sorrows and Hirtle (1999) suggested that landmarks could be prominent visually, cognitively, or structurally, and that the strongest landmarks in an environment will be so in all three categories. Landmarks may also be classified as local or global, and while experiments performed in a virtual environment demonstrated that although people had preferences for which they used, either would suffice if necessary (Steck and Mallot, 2000).

If reading graphs non-Visually is considered to be a browsing activity, then any tool facilitating it must enable users to build a cognitive model of the graph via steps analogous to those involved in acquiring spatial mental representations of environments. Considering this process with the understanding that travel in a node-arc environment is constrained by the requirement to move only along arcs (this muddles the meaning of distance, and is the basis of Boechler’s (2001) criticisms), we can identify the stages involved in understanding a graph, and suggest some difficulties that are likely to be encountered.

1. Landmark recognition can be considered a phase of exploration where the user gains an understanding of the types of nodes present. The essence is *discovery*, and it is typified by wandering around the graph.
with no particular direction in mind, rather an intention to see what there is. In this phase difficulties could include:

- Understanding the scope of the graph, e.g., knowing when all nodes have been visited.
- Distinguishing nodes from one another.
- Visiting all nodes, i.e., finding unvisited nodes.

2. Landmark co-ordination is a phase where movement generally has an intended destination, even if the exact location of the destination is unknown. For example, the user wishes to know the arcs and nodes between two particular nodes. The most significant problems likely to be encountered in this phase are memory-related:

- Remembering and recognising (differentiating) nodes.
- Remembering and identifying arcs.

3. The graph exploration analogy of survey knowledge formation is probably linking (relating) sections of graphs rather than combining routes into a survey, although it is essentially the phase where an overall understanding of the graph is developed. This is like relating nodes but at a larger scale, suggesting that the difficulties will include:

- Recognising the presence of coherent groups of nodes (chunks).
- Identifying chunks during navigation.
- Remembering relationships between chunks (or alternatively remembering the relationships between large numbers of individual nodes).

The phases given above are for general exploration; browsing for a particular task might not require development of a full mental model of the graph. For example the user might only need to know from a graph whether a certain node type exists. Similarly, many tasks might only demand a good model of a subgraph; in this instance the rest of the graph need only be explored to the level of landmark identification (or sufficient to recognise the area of interest). It is in these cases that the importance of search comes to the fore. While Larkin and Simon identified that the ability to easily search was one of the key attributes of a diagram, this was essentially a local search: nodes that are closely related are represented nearer (in the 2D presentation space) than those which are not, whereas the linear form of a sentential representation might mean that two connected nodes are described in distant parts of the text. This benefit may be at least partly recreated in an audio implementation by enabling connection-based browsing, where users move from node to node along the arcs. The search that is important for users in
the cases identified above, however, is a global search. The ability to search the graph for a node, arc (or chunk) with a particular characteristic, then move there for exploration, will significantly reduce the effort required for these tasks, when compared with having to explore the graph manually.

Finally, an important feature of real-world navigation, both before and after mental map formation, is the use of signposts (see Passini, 1984). These aid decision making during travel, reduce the demands on memory, and reassure users. If the spatial metaphor is valid for non-visual graph exploration, as we believe it is, consideration should be given to generating an audio implementation of signposts.

2.4 Requirements

This review of the literature has explored the difficulties associated with presenting information aurally, rather than visually, and considered the particular benefits that graphs offer. Thus, it is now possible to understand the requirements for any non-visual graph browsing system. It must:

1. Facilitate recognition: make the implicit explicit.
2. Allow searching, both globally and locally.
3. Assist with node differentiation (landmarks).
4. Give summary information, both for previewing and as context.
5. Enable interaction with the graph. E.g., allow users to add notes to help solve problems.

Before attempting to find a solution that can fulfil these requirements, we survey the previous work in this field. By doing this, we hope to understand how the problem has been approached in the past, and to explore how the successes and failures of those approaches may be explained by the fulfilment, or otherwise, of the requirements.

3 Related Work

This section examines some previous research into presenting graphs to visually impaired users, with emphasis on evaluated tools. Each tool is described with emphasis on how it has, or has not, tackled these requirements. Since this paper concentrates on audio solutions this review is similarly biased.

Mathtalk, by Stevens, Edwards, and Harling (1997), considered a complex algebraic equation to have a hierarchical structure where a sub-expression is itself composed of sub-expressions. This audio tool fulfilled some of the requirements above. While features were not recognised and presented explicitly, the tool did identify sub-units of an equation so that it could be
presented hierarchically. In addition, this information was used to inform the prosody used when reading the equation. Of particular note was the use of Earcons to provide summary information: the authors stressed the need for summarisation as a method for the reader to keep in mind his ‘location’ in the equation and estimate the complexity of an expression — this was facilitated by the hierarchical view. Further themes emerging from this research were the need for the reader to be active in his reading (not just passively have the equation read to him), and the need to overcome lack of an external memory.

The importance of summary information has been recognised in the world of information visualisation, and is perhaps best expressed in the Visual Information Seeking mantra: “Overview first, zoom and filter, then details-on-demand” (Shneiderman, 1996). Perhaps as a result of this, it has been given greater prominence by those researching audio access to numerical data. Zhao, Plaisant, Shneiderman, and Duraiswami (2004) re-phrased this mantra for audio: “Gist, Navigate, Filter, Details-on-demand”; the parallels with the stages identified in section 2.3 are clear. This has been applied to numerical georeferenced data in a system that provides three highly coordinated data views: a table (each row representing a region of the map and the corresponding data values), a map, and a scatter plot (Zhao, Plaisant, and Shneiderman, 2005; Zhao, Plaisant, Shneiderman, and Lazar, 2008). In this, gist is provided by sonifying rows or columns of the tables, or by transforming the map into a table and providing a row by row sonification for a single variable. Kildal and Brewster (2005) examined how to summarise tables of numerical data; they stated: “A natural way of examining a data table (or any other data structure) that is encountered for the first time always starts with a browsing stage to obtain overview information about it and to judge on the relevance of further analysis, as well as where or how to carry it out.”. They mapped the numerical value of a table cell to pitch, then created a complex sound to represent an entire row or column.

The partners in the TeDUB project (“Technical Drawings Understanding for the Blind”) developed a system for understanding digital circuit diagrams, which presented information in a hierarchical structure (intended to chunk the information to reduce the demands on short-term memory and facilitate overviews), while also allowing connection-based movement (Horstmann et al., 2004). A degree of recognition was used, with the intention of chunking information to reduce the demands on short-term memory and facilitate overviews, for example a group of logic gates might be grouped to form a half-adder. Summarisation was given by the hierarchical method of navigation. Users could annotate nodes. Nodes of the same type (e.g., AND gates) were differentiated using numbers (AND gate 1, AND gate 2, etc.).

The approach taken by Blenkhorn and Evans (1998) was to concentrate on the connections between nodes. Their system, known as ‘Kevin’, used
a tactile pad in combination with audio output to allow non-visual reading and editing of a type of data-flow diagram used by software developers. The tactile pad was composed of two regions, with the output area split into a $N \times N$ grid, where $N$ was the number of nodes in the graph. The leading diagonal of the grid gave access to the nodes and their attributes, while the remainder was used to give access to information about the connections. This approach clearly enabled connections to be identified quickly, but did not facilitate recognition. Since Kevin allowed editing of diagrams, some level of external cognition was enabled, although there did not appear to be any formal method for annotation. Node differentiation was provided implicitly: since input was via a tactile pad, nodes necessarily occupied different spatial locations. Kevin did not appear to offer summaries.

Bennett (2002) looked at the limitations of the Kevin system and proposed that presenting information as a hierarchy would afford the benefits Larkin and Simon associated with grouping. He investigated how the nature of the task influenced whether diagrams were better presented with this hierarchical structure, or with a connection based structure, as with Kevin. He conducted some experiments using central heating schematics, and demonstrated that hierarchically presented information facilitated hierarchical tasks, but that if the tasks were navigational the information was best presented with an emphasis on connections. A brief summary was provided during loading, which gave the diagram name, the number of elements within it and the maximum depth of the hierarchical view. This was intended to give an overall view of complexity. Elements were given exclusive numbers.

Bennett also presenting location information, since he felt that previous work ‘suggests that position information is part of the reason why diagrams are successful representations’; he argued that not knowing the location of the components in the original diagram creates an informational inequivalence. He therefore also investigated if musical ‘earcons’ (the audio equivalent of graphical icons) presenting coordinate information would ease the tasks, but found no evidence to support this hypothesis. It is debatable if lacking coordinate information destroys informational equivalence; it is arguable that in graph-based diagrams the inequivalence is purely computational. While Kevin used a tactile pad to present a transformation of the graph, a direct rendering of the graph on a tactile pad, such as Audigraf Kennel (1996), gives location information implicitly, in a way that also allows differentiation of nodes. Kildal and Brewster (2006a,b) also used physical space in their table visualisation tool TableVis, providing their row and column summaries directly from graphics tablet.

Brown, Pettifer, and Stevens (2004) evaluated a system (Kekulé) that enabled non-visual browsing of molecular structure diagrams. This system allowed some of the theoretical ideas described above to be examined in practice. The main conclusions of the evaluation were as follows.
Effectively enabling both connection- and hierarchy-oriented browsing will allow a range of tasks to be accomplished.

Techniques for making the implicit accessible proved important.

Annotation by the reader offers possibilities for enabling computational offloading.

Orientation is difficult but might be tackled with techniques used for non-visual Web browsing.

Use of the spatial metaphor is supported by the language used by participants when commenting on difficulties (e.g., ‘where am I?’). Further evidence is provided by the observation that many of the difficulties could be attributed to disorientation. The applicability of Larkin and Simon’s conclusion about recognition was also demonstrated by the ease with which one participant was able to understand the structure of a complex molecule when its components were made explicit as large, meaningful, chunks rather than individual atoms.

Metatla, Bryan-Kinns, and Stockman (2007, 2008) attributed some of the issues encountered by Kekulé to lack of any prior knowledge about how the node grouping was going to occur. Accordingly, they examined how a fixed hierarchy might be used to structure the data more predictably. The system they proposed for browsing Unified Modelling Language (UML) diagrams (Metatla, Bryan-Kinns, and Stockman, 2007) used a hierarchy with a fixed upper structure: the three top-level categories being objects (nodes), associations (arcs), and generalisations. At the second level of the hierarchy were the individual nodes and arcs, while the third and fourth levels allowed users to determine which nodes an arc connected (and its direction) and to which other nodes a node were connected (and by which arcs). The design of the user-interface was such that it was simple to shift between node and arc-based views of the graph. Although they claimed that ‘we push down the dynamic components, which are specific to a particular diagram, to a deeper level in the hierarchy’, it is not clear how this worked, and it appears that their system did not allow for recognition and presentation of chunks of a diagram. Thus, the hierarchy described acted as a means of access to information about the nodes and arcs of the diagram, rather than a means of organising them into groups. Of note are the comments of visually impaired participants in their evaluation that a bookmarking feature would be desirable. Although a graph editing tool (Metatla, Bryan-Kinns, and Stockman (2008), which used a hierarchy based only on nodes and arcs, with no use of the generalisations category) has been implemented and tested, annotation of graphs did not appear possible.

In summary, it can be seen that the requirements enumerated above are rarely all met. Additionally, the justifications given by the designers
Bennett reported a comparison between hierarchical and connection-based browsing, but did not report a system that combined the two.

Via the hierarchy.

A version did allow creation and editing of graphs, but apparently not annotation.

In the UML diagrams studied, entities are necessarily differentiated by class names.

Spatially, using a tactile pad.

A numbering system was used, but was considered poor.

of these tools are rarely reported. While the TeDUB system fulfilled all of the requirements, the design decisions were not rationalised in terms of the benefits of diagrams and the differences between visual and aural representation. For example, the hierarchical presentation of the logic circuits was justified in terms of chunking (to reduce memory demands), rather than facilitating recognition. Without understanding why the different features of TeDUB benefited users, it is difficult to apply them more widely. Figure 2 summarises which requirements were satisfied by the systems introduced above.

### 4 Annotation

In a simple experiment designed to discover the vocabulary used to describe diagrams and to give some insight into the type of strategies people used to describe, explore and understand graphs, four pairs of volunteers were asked to take turns describing abstract graphs (with between 6 and 23 nodes) to each other. The participants were asked to describe the graph so that their partner could later recreate it. Taking notes was not allowed, although the process involved conversation, not just a one-way description. Participants only needed to recreate the node-arc structure of the graph, not its precise layout. The vocabulary used was generally very simple, usually relying on up, down, left, right, top and bottom to describe the location of nodes or groups of nodes. Graphs were often described starting with a central feature.
before progressively describing other sets of nodes and arcs attached to it. Despite only needing to consider connectivity, by far the most common method of describing the graphs was to use the shapes formed by closed loops of nodes and arcs, such as squares, triangles and hexagons. Most pairs used some form of labelling when describing their graphs; although it was often implicit, explicit labelling, in the form of numbers, was applied in some cases. In general, performance appeared limited by memory. Labelling was generally applied to groups of nodes; presumably this chunking was an attempt to reduce the number of items it was necessary to remember. Those graphs where there were large numbers of nodes or arcs, and where chunking was not possible, proved the most difficult to communicate.

This experiment, and the evaluation of Kekulé, are both supportive of the expectations that were derived from theoretical considerations. Difficulties with memory capacity were considerable, but were eased by identifying groups of nodes that formed recognisable features. The language participants used to describe their experiences gave weight to the validity of the spatial metaphor, suggesting that supporting orientation and navigation could prove beneficial. It may also be noted that many of the strategies used by participants to perform their tasks were strongly reminiscent of forms of annotation. This prompts the following question: is annotation as a technique sufficiently generalisable to form the basis for supporting non-visual browsing of graph-based diagrams? In other words, can annotation be used to replace certain of the benefits imparted by visual presentation of graph-based diagrams, including making implicit information explicit, grouping related items, interactivity and acting as an external memory, and to reduce disorientation while moving around the graph?

The definition of annotation (Trumble and Stevenson, 2002) is simple: ‘A note by way of explanation or comment,’ but the concept is powerful. Supplementing the primary source of information with notes can be used (amongst other things) to clarify, highlight, correct, or reference, and has been shown to benefit readers (Marshall, 1997, 1998). If annotated text books can be useful to future readers, why not diagrams also? And if annotation augments diagrams, can suitable notes be added to graphs that will ease some of the problems associated with exploring them non-visually?

4.1 Annotation in Interfaces

Annotation is a powerful method for conveying information, and has been applied to many types of information from traditional documents (Wolfe, 2000), through hypertext (Marshall, 1998) to the ‘Semantic Web’ (Berners-Lee, 1999), in addition to data such as protein sequences (Stein, 2001). It has also been used for applications more closely related to non-visual graph browsing. Several groups have looked at how Web site accessibility can be improved by identifying and annotating certain features (Asakawa and
Takagi, 2000; Takagi, Asakawa, Fukuda, and Maeda, 2002; Yesilada, 2005). Yesilada, Stevens, and Goble (2003) applied the travel metaphor to Web browsing, and identified and classified travel objects on a page; these are ‘environmental elements that are used during a journey’. For example, a menu was classified as a decision point, as it is a point where alternative paths are possible. The Danté project (Yesilada, 2005) continued this work, examining how travel objects could be identified and annotated, then using these annotations to transform the page in order to improve its accessibility. Evaluation demonstrated that this approach could improve the experience of visually impaired Web travellers.

Another use of annotation in accessibility is Wall and Brewster’s use of ‘beacons’ for people exploring bar charts through a haptic force feedback device (Wall and Brewster, 2005). Their system allowed people to add multimodal beacons on a bar chart to mark points of interest, acting as an external memory. Qualitative studies suggested it was most useful for comparing two distant bars on a chart.

In addition to using annotation to improve accessibility, it has also been used to improve visual representations of data. Of particular relevance in this context is the work of Ingram and Benford (1996). Ingram and Benford applied the work of urban planner Kevin Lynch, who considered how one might develop a cognitive map of an urban landscape (Lynch, 1960). Lynch identified five key features: paths, edges, districts, nodes and landmarks (Lynch, 1960, chap. 3). Ingram and Benford (1996) took these attributes and developed algorithms to generate these features in information visualisations. They were applied as a ‘legibility layer’, but may be seen as annotations to the 3D visualisation. For example, nodes (which essentially represent important points within the city, such as junctions) were presented as larger than normal data points, while districts (regions with a coherent, distinct, character) were presented with different colours. Their evaluation was limited, but suggested that annotating the visualisations with these features helped people to find a previously visited known point within the dataset. Some of these features will be discussed in more detail below.

4.2 Annotating Graphs

If annotation has proven useful in the variety of applications noted above, what types of note can augment a graph so that the benefits of visual presentation are recreated and the problems of aural presentation are minimised? Working in the audio domain means that annotations cannot take their traditional form of marginal text, and the information must be presented as sound. This might be seen as a benefit — freed from the constraints of being represented on paper, non-visual annotations may take the form of

\textsuperscript{2}Note that these terms are used differently by Lynch than when applied to graphs.
speech or non-speech sounds, or some combination. Without a margin, however, annotations (particularly speech-based ones) must take up a primary position in the information stream, perhaps temporarily preventing presentation of the main information and potentially acquiring a significance on a par with it. In this section, we consider some possible forms of annotation, concentrating on how they might ease some of the problems associated with non-visual graph browsing.

The annotations proposed below are all suitable for inclusion in a non-visual graph-reading tool. These annotations do not need a third party to create them, but are generated either when the user opens a graph or during exploration. Some are calculated from the structure of the graph, for example, algorithmic deduction of the presence of higher-level features so that they may be made explicit. Others are generated as part of the exploration process, for example, notes identifying previous visits to a node. These distinctions are clarified in the taxonomy (section 4.4). While generation of some of these annotations is limited in the case of a truly generic graph, the use of class-based rules mean that, crucially, no manual annotation is required before a user explores a particular graph. Some examples of class-specific differences in the implementation of these annotations are given below.

Chunking

Larkin and Simon (1987) concluded that one of the main reasons diagrams were effective was that information that might otherwise be implicit becomes explicit when presented in a diagram. If we are to allow visually impaired users to use diagrams effectively, then implicit information must be made explicit to them. Recognition of implicit features necessarily involves groups of nodes; if the nodes belonging to a group are annotated as such, the user can be made aware when moving onto a node belonging to a chunk, or be told what groups are present in the graph. The former should help as it will let the reader know what connections there are (assuming they are familiar with the group). The latter will allow the mental model to be constructed more easily, with larger chunks (hence fewer components). A graph annotated for chunking can also naturally lead to hierarchical browsing, which the evaluation of Kekulé demonstrated to be effective for achieving many tasks. Ingram and Benford (1996) found that their implementation of districts was most useful when used viewing the entire space, supporting the idea that this information is particularly helpful in forming an overview.

This type of annotation is a good example of how algorithmic annotation can very powerful for some classes of graph. A particularly striking example is found with molecular structure diagrams, where a well-defined set of rules may be applied to chunk atoms into units known in chemistry as ‘functional groups’ — if a reader is familiar with the form of such a functional group
Figure 3: The benefits of chunking for molecular structure diagrams. The original graph (a) has 9 nodes and 9 arcs, but grouping allows these to be presented (b) as 2 nodes and 1 arc. Experienced chemists will understand either representation equally.

there is no need to explore the detail. Figure 3 gives an example of this, showing how chunking can (for someone familiar with the domain) simplify a 9 node graph into a 2 node one. Even in the most generic case of an unknown node-arc diagram, the presence of loops (a potential cause of disorientation during exploration) and chains of nodes can be detected and presented to the user. Algorithms such as the Minimal Spanning Tree algorithm for clustering (Zahn, 1971) applied by Ingram and Benford (1996) may also be applicable (given a measure, presumably domain-specific, of edge distance), to identify and group regions of the graph that have a distinct character. In family trees it is possible to group nodes (people) into families of blood relatives, or those who share (or have shared) a surname (see Figure 4). In logic circuits, nodes may be clustered according to which output they influence (some gates can only affect the value of certain outputs), as shown in Figure 5.
Figure 4: Examples of annotation in a graph. This family tree is annotated with a home node, relationships, chunks (family groupings), numbering, and a summary.
Figure 5: Examples of annotation in a graph. This logic circuit is annotated with chunks (grouping gates according to the outputs they affect), numbering, and a summary.
Home Node

The formative evaluation of Kekulé (Brown, Pettifer, and Stevens, 2004) showed how important it was to allow users to return easily to a familiar node if they became disoriented. This behaviour could be replicated by enabling any node to be nominated and annotated as the ‘home’ node, and providing suitable functionality in the user interface to ‘return home’.

Relationship

Relationships are simply depictions of the relationships between entities, as represented by arcs and nodes respectively. While the direct relationships between nodes are always explicit, relationships between more distant nodes are often explicit in (or easily deduced from) a visual presentation, but implicit and more difficult to deduce non-Visually. Discovering them requires detailed exploration of the graph to find and both nodes and the path(s) between them — a task that becomes increasingly taxing for more distant relationships. Having a node annotated as ‘home’ allows these relationships to be made explicit by annotating all nodes with their relationship to the home node (and thus making the home node a global landmark).

This type of annotation can be used to further exemplify the power of class-based rules for automatic annotation. Sighted users benefit from a roughly standard layout when determining the relationship between two people in a family tree, but calculating this non-Visually requires traversal of the tree (or a sub-graph) — a non-trivial task prone to navigation and orientation errors and demanding for the short-term memory. Once a home node has been designated, however, the remaining nodes on the tree can be annotated with their relationship to that node. This can be done using a fairly simple set of rules, but can reduce the task to a trivial one.

Figure 4 gives an example of a family tree annotated with relationships. In this example the node ‘Andrew Brown’ is designated as home (this might have been done by the user), and the remaining nodes have been automatically annotated with their relationship to this person. Thus, wherever the user is while exploring the tree, their location relative to the home node is explicit. Even relatively common relationships (such as cousin, not shown in the example) are simply deduced, despite being 4 arcs distant.

User Notes

The existence of an external representation of the graph eases the memory demands, but also allows interaction: manipulation of the model can be performed externally to help in various tasks. For example in a large graph, even a simple task such as counting the number of nodes can be eased by marking off each node as it is counted. Figure 6 shows how a logic circuit
may be annotated to determine its output value; this method is less error-prone than calculating mentally, particularly for larger diagrams. Allowing users to give nodes labels, and to attach more lengthy notes, could facilitate such tasks, and allow certain nodes to become more salient as landmarks, and be searchable, as bookmarks.

![Logic Circuit Diagram](image)

Figure 6: Task-based annotation by the user. In order to calculate the output at C, given the input values at A and B, the user has annotated this logic circuit with the values that pass to each gate, reducing the risks of error associated with mental calculation.

**Location**

The evidence as to whether knowing the location of a node in the original representation aids understanding when browsing non-visually is unclear. Bennett (2002) found that presenting coordinates as pairs of notes was not useful to his participants, but it is possible that other methods may prove useful. Some participants in the Kekulé evaluation (Brown, Pettifer, and Stevens, 2004) appeared to find the position (presented as ‘top left’, ‘middle’, etc.) useful. If available in the original representation, annotating each node with its location would allow a user interface to take this information and present it in any form (e.g., Cartesian or polar coordinates, presented as spoken numbers or with tones, descriptions such as ‘top left’, or even 3D audio).

**Visit Histories**

As an addition to jumping back to a previously visited node such as the home node, it is important that the path of exploration may be retraced in smaller steps. While there are several methods by which this may be
achieved, some form of annotation could certainly enable this functionality. This type of annotation could also be used to warn users if they are exploring a new part of the graph, or conversely if they have visited a node previously — potentially useful information when trying to ascertain the extent of a graph.

Summaries

In addition to annotating individual nodes or arcs (e.g., with relationships or locations), the entire graph, or groups of nodes, may also be annotated. For example there are several features of a graph that may benefit readers if made explicit. These include the number or nodes and arcs, the number of different types of node, and the complexity of the graph. Information about the graph as a whole is essentially summary information. The design and evaluation of a non-speech audio glance at abstract graphs is described in Brown, Stevens, and Pettifer (2006). These audio glances were generated by passing through the graph along the arcs, playing a short sound each time a node was encountered. Where more than one arc left a node, all arcs were followed simultaneously, although the node sounds were temporally separated. User evaluations found that these Earcons could be used to convey the gist of a graph.

Speech-based summaries can also be provided, for example giving the total number of nodes and the number of each type of node, if appropriate. For example, a logic circuit might be summarised with the total number of nodes, plus the number of each type of node (see figure 5). A family tree might be summarised with the number of people, the number of generations, and the number of people with each surname (see figure 4). A user-interface designed to make use of these ‘whole graph’ annotations in an interactive way can make these summaries a potentially powerful way of exploring the graph. For example, when the summary gives the number of nodes of a certain type, one might be able to access a list of those specific nodes, from which one could access the nodes directly.

Node Identification

In visually presented graphs it is relatively easy to distinguish nodes as they are differentiated by their location. Further differentiation may be provided by other information on the graph, for example names or labels next to the nodes, the way in which a node is drawn. Differentiating nodes in a non-visual environment is more challenging. Location is not immediately apparent, and the other information can only be given with a significant time (and attention) cost. The constraints of audio presentation mean that each node must be distinguishable with a short sound (speech or otherwise); numbers are an obvious candidate.
4.3 Examples

To exemplify some of these types of annotation, figures 4 and 5 give graphs from two domains which have had limited annotation performed. Figure 4 shows an annotated family tree. In this example, a home node is selected, and nodes are annotated with their number and relationship (to home). Nodes are also annotated with group membership — they are chunked into families. The graph as a whole is annotated with some summary information. Note that if this were drawn strictly as a graph the relationships would be between individuals, e.g., a child would have a relationship to each parent, rather than to the marriage. The graph is shown in the standard form for simplicity and clarity.

Some of the benefits of these annotations are clear, even when reading the graph visually. Although the layout makes relationships reasonably simple to deduce, it is easier when the annotation makes them explicit. The numbering can distinguish between people of the same name — not uncommon — and give a clue as to whereabouts on the tree the person is (very roughly, a lower number indicates they are nearer the top). A further, domain specific, annotation that could help in this regard is to annotate each person with the number of their generation (not shown in Figure 4).

Figure 5 is a logic circuit that describes a ‘half-adder’. This uses numbering to differentiate nodes that have the same name (where the name is the gate type: AND, OR, etc.), has a summary with a similar form to that used on the family tree, and has chunks automatically identified according to which nodes are on paths to each output (this is a directed graph).

4.4 A Taxonomy of Annotation

The forms, function and origins of the annotations introduced above were wide and varied, from annotations with sounds that summarise the entire graph, to labels generated by the user, to temporary annotations enabling cost-free exploration. A system for classifying annotations should help the design of tasks for their evaluation. This section introduces two ways in which annotations may be classified. The first classification considers the quality of the information, the second the difficulties the annotation are intended to ease.

Provenance

If information is being presented to the user, it is important that they know whether that information is true, i.e., explicit in the original representation, inferred, or estimated. This type of categorisation is useful, since it strongly reflects the method by which the annotation was generated. We may fit annotations into four classes of provenance.
• Original information: That which is explicit in the base representation (i.e., the description of the graph provided to the annotation/reading system).

• Inferred information: That which is inferred from the graph, e.g., detected features. This can be further broken down into subjective or objective. This category also includes some information which may be considered fairly artificial or arbitrary, such as node numbers, or the initial home node.

• User-given: Information that is provided directly by the user.

• Recorded: Information that is collected about the user’s behaviour, e.g., breadcrumb trails.

Although these classes can be placed on a scale running from information that can be called true with some certainty (where ‘true’ is used in the sense of accurately reflecting the base information) to information with which the original author may not agree, this should not imply that this correlates with usefulness. Some of the less ‘truthful’ annotations are simply one of a set of choices (e.g., numbering systems) that, even if not optimum, should help exploration.

Application

An alternative, or complementary, method of classifying annotations is to examine how they are supposed to benefit readers; this should give a clearer picture of whether annotations are capable of helping with the range of difficulties users may encounter. Of course, some forms of annotation can be beneficial for several different reasons; the classes used are therefore not exclusive. Broadly speaking, the reader may encounter three main areas of difficulty, which become apparent when the problem is viewed using the spatial analogy. These are: summarising, orientating, and relating. A further class is for annotations that help user tasks.

• Summarising: Notes that give the reader an idea of the scope of the graph.

• Orientating: “The action or process of ascertaining one’s bearings or relative position, or of taking up a known bearing or position” (Trumble and Stevenson, 2002). At any point in the exploration it is important that users have a clear understanding of their location (i.e., they are where they think they are). This class of annotation includes notes that help prevent readers from becoming disorientated.
Figure 7: A taxonomy of annotations.

- **Relating**: The major problem in moving around this type of space is more one of building up a mental model of how the nodes are related than one of traditional navigation. These notes, therefore, are ones which are intended to assist explorers determine how nodes are related.

- **User task**: Notes which may directly help a user perform a task with the graph, rather than helping them read the graph.

### Classification

The different types of annotation discussed may now be classified using the two schemes described above. Figure 7 presents this classification. This classification makes more explicit how each type of annotation is expected to help, and should therefore inform both the design of any evaluation, and assessment of whether the annotations were beneficial.

### 5 Discussion

This paper has explored and characterised the problem of non-visual graph browsing. The most significant issues encountered are associated with the lack of an external memory, leading to large demands on the reader’s internal memory, and the need to make implicit features of the graph explicit. It has been argued that there are sufficient similarities between non-visual graph exploration and real-world exploration of spatial environments to make this a useful analogy for identifying and alleviating problems. Of particular importance is the need to help users retain their orientation. A collection of annotations have been developed, both have been presented and classified; these are intended to augment graphs such that some of the difficulties identified above are reduced. The various forms of annotation explored here have
been combined in a tool to allow non-visual exploration of logic circuits and family trees. This tool loads a graph and automatically generates annotations appropriate for its type (further annotation occurs during exploration, but none is required beforehand).

This is an approach that suffers from some limitations. The most obvious is that tailoring is required for each different class of graph. In particular, the ways in which nodes are grouped into chunks is heavily domain-dependant. Although a generic solution is possible, based purely on graph topology, it is not clear how beneficial such an approach would be. Other customisation is applied to tailor other annotations (and their presentation), such as summaries and descriptions of relationships. A second limitation is that the effectiveness of the use of annotations will be critically dependent on the way in which they are used and presented — the user interface. Suggestions have been given as to how different annotations may be used for the domains of family trees and logic circuits; the user interface design must facilitate the particular role of each type of annotation in helping the user.

Despite these limitations, this solution is one that can be applied widely. This is because the approach is based an understanding of the fundamental difficulties of non-visual graph browsing, and the benefits that graphs offer sighted users. The main benefits of diagrams as a means of presenting information, and the ways in which annotations can replicate them are as follows:

Diagrams facilitate recognition — features that would otherwise be implicit become explicit when presented diagrammatically. Automatically identifying these features and annotating the graph to highlight their presence can simplify exploration, especially if used to build a hierarchical view of the graph.

Diagrams provide an external memory that readers may access very quickly. The transient nature of sound, however, radically changes this, making graph reading more like spatial exploration; annotations can be designed to ease the problems associated with exploration (see below). Diagrams may also be interactive, with readers able to modify and add notes of their own. Allowing similar user interaction in the form of notes and labels can enable non-visual graphs to be used in a similar way. For example participants in the evaluation annotated a gate in a logic circuit with its value or function so they could follow the logic from input to output.

Diagrams automatically summarise — due to the way they act as an external memory, and the rapidity of eye-movements, much information about the overall nature of a graph may be discerned from a glance, potentially informing readers of strategies for reading and using the information. Annotating the graph with summaries can allow quick access to similar information.

Diagrams also benefit readers by grouping related items. This benefit can be replicated through both annotation, indicating when nodes are part
of a group, and user-interface design, allowing groups to be explored hier-
archically and enabling connection-based browsing. In addition, one of the
key roles of graphs is to represent relationships. Augmenting a graph with
annotations that make near and distant relationships between nodes explicit
proved beneficial.

Considering next the ways in which understanding a graph non-visually
compares with exploration, the following difficulties may be expected, and
counteracted with annotation:

One of the most significant causes of disorientation is not recognising
where one is, due to an inability to distinguish locations. Differentiating
nodes using numbers was shown to be a valuable means of helping users re-
tain orientation. Another means of differentiating nodes, particularly during
the early phases of exploration, is by identifying if one is visiting a new lo-
cation or returning to one that has been previously visited.

Even with various annotations described above, disorientation is still
a distinct possibility. One particular problem for disoriented travellers is
returning to a known location in order to re-orient. Having a node designated
a ‘home’ node (and allowing this designation to be applied to any one node in
the graph) can allow people exploring non-visual graphs a means for quickly
jumping back to a known location.

The concept of a node annotated as a ‘home’ node is also crucial to
the idea of making relationships explicit. When a node is designated as
‘home’, the relationship between it and all other nodes may be calculated,
and these nodes annotated with the relationship. A user anywhere in the
graph may then be able to rapidly relate her current position to the known
location of home. The fundamental importance of relationships in graphs
was demonstrated by the value of this type of annotation in the evaluation.

Graphs are the source of much important information; allowing non-
visual access to them is crucial to allow people with disabilities to participate
fully in society. This paper has shown that a rigorous approach, gaining a
deep understanding of the benefits of graphs and the problems of non-visual
interaction can give insight into how these difficulties may be reduced, and
allow effective generic solutions to be designed. This work has demonstrated
that annotation is a powerful, and very flexible, technique for overcoming
many of these difficulties, and should therefore provide a good foundation
for continued improvement in this important field.

Notes

Background. This article is based on the Ph.D. thesis of the first
author.

Support. This research was undertaken while funded by an Engineering
and Physical Sciences Research Council (EPSRC) studentship.
Authors’ Present Addresses. Andrew Brown, School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK. Email: andrew.brown-3@manchester.ac.uk. Robert Stevens, School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK. Email: robert.stevens@manchester.ac.uk. Steve Pettifer, School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK. Email: steve.pettifer@manchester.ac.uk

HCI Editorial Record. First manuscript received May 15, 2009. Revision received November 5, 2010. Accepted by Rob Jacob. Final manuscript received August 12, 2011. – Editor

References


Bennett, D.J. (2002) Effects of navigation and position on task when presenting diagrams to blind people using sound. Lecture Notes in Artificial Intelligence, 2317, 161–175.


