

Evaluating a Tool for Improving Accessibility to Complex Visual Objects

Leo Ferres
Dept. of Computer Science
Universidad de Concepción
Concepción, Chile
lferres@
udec.cl

Gitte Lindgaard
HOTLab
Carleton University
Ottawa, ON, Canada
gitte_lindgaard@
carleton.ca

Livia Sumegi
HOTLab
Carleton University
Ottawa, ON, Canada
lsumegi@
connect.carleton.ca

ABSTRACT

We discuss factors in the design and evaluation of natural language-driven (NL) assistive technologies that generate descriptions of, and allow interaction with, graphical representations of numerical data. In particular, we provide data in favor of screen-reading technologies as a usable, useful, and cost-effective means of interacting with graphs and data in favor of carrying out evaluation on the target communities, rather than potentially equivalent participants, with whom certain important subtleties of navigation and interaction may be lost.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing—*language generation, natural language interfaces*; H.5.2 [Information Interfaces and Presentation]: Natural Language

General Terms

Human Factors

Keywords

Accessibility (blind and visually-impaired), natural language generation and interaction, statistical graphs and diagrams

1. BACKGROUND AND MOTIVATION

Charts and graphs (line graphs, bar charts, stacked-bar charts or any of their combinations) are an extremely common means of representing numerical data for exploration and mass-consumption. Although the number of published graphs is impossible to estimate, for just one publication (“The Daily”¹), Statistics Canada, for example, publishes on average two graphs a day on a wide variety of topics (employment, GDP, etc.). This means about 500 graphs a year.

¹<http://www.statcan.gc.ca/dai-quo/index-eng.htm>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ASSETS '10 Orlando, Florida USA

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

This is for just one of the tens of publications of this agency. This figure is simply to put into perspective the staggering number of published graphs if we aggregate those appearing in newspapers, business reports, and science textbooks.

This stands in stark contrast with the fact that a majority of the almost 88,000 *reported* legally-blind people living in Canada in 2006² (993,766 in the US³, 2004) do not yet have cost-effective, straightforward access to visually-encoded information. This means that the blind community is effectively excluded from a vast repository of useful information, and from jobs, and that their chances at any area of education that deals with quantitative data is seriously restricted.

Fortunately, excellent research into non-visual interfaces to graphs and charts has been carried out in the past decade to help alleviate this problem. These non-visual interfaces to graphs generally fall into four loosely defined categories: sonification [11, 3, 4, 14, 2], haptic interfaces [20, 18, 16], hybrid systems [19, 13, 12] and NL interfaces [9, 1, 6] which use sound (linguistic and non-linguistic), touch or both together to communicate graphical information to the user.

However, evaluation of these technologies has been relatively scarce, but this is particularly noticeable in the NL group: Are NL interfaces to graphs useful and usable for the target community? This paper reports on a carefully designed user-based evaluation of **iGraph-LITE**, a novel Assistive Technology (AT) which has been designed to help blind and visually-impaired individuals interact with graphical representations of statistical data using natural language through key commands and a Text-To-Speech (TTS) engine. The purpose of this evaluation is three-fold. First, we aim to determine whether the information in line graphs is accessible to blind and visually-impaired individuals through the **iGraph-LITE** system, and, generalizing, through language and screen-reading technology. Second, we aim to identify *which* parameters within the system might aid or hinder interaction with graphs and to offer insight into the navigational strategies used by blind people during (line) graph comprehension. Finally, we believe that when conducting a study in which participants evaluate novel NL ATs, it is important that the participants are members of the community for whom the system is designed. Although researchers have used seemingly equivalent participants such as blind-folded sighted people to sit in as legally blind participants, it may be the case that there are some subtleties that are

²<http://www.statcan.gc.ca/pub/89-628-x/2009013/fs-fi/fs-fi-eng.htm>

³<http://www.center4research.org/blind0204.html>

evident when the actual target demographic is tested. We have therefore tried to shed some light into this matter.

This is, to the best of our knowledge, the most comprehensive empirical study of blind people interacting with graphs through natural language. We have thus made available the stimulus, materials (including the **iGraph-LITE** navigator that was current at the time of this studies) and the whole dataset in the hopes that some other researchers might find them useful for replication and extension⁴.

2. IGRAPH-LITE

In this section we provide a general description of the **iGraph-LITE** architecture and the first attempt during the iterative design of the navigational commands of the system.

2.1 General architecture

iGraph-LITE comprises three subsystems: (1) a knowledge representation system that enriches a basic semantic representation of line, bar and combination graphs, (2) a natural language generation system that produces a static description of a graph, and, (3) an interface that allows users to navigate the full enriched representation of (1) above by means of keyboard combinations, much as JAWS⁵ or DOLPHIN⁶ do. **iGraph-LITE** can therefore be used to generate rich descriptions to publish alongside graphs through the `longdesc` tag in plain HTML, and, if a graph is published with **iGraph-LITE**'s full semantics embedded through `exif`⁷ or a similar tool, the **iGraph-LITE** navigator can be used to explore the graph at different representational levels.

iGraph-LITE works by first accessing the Object Model (henceforth, OM) for graphs and charts in common plotting and graphing applications such as MS Excel, GNUMERIC, GNUPlot or Lotus. **iGraph-LITE** extracts the data from the OMs and writes an eXtended Markup Language (i.e., XML⁸) and/or a Web Ontology Language (i.e., OWL⁹) representation of important graph information (including *geometrical properties* such as width and height and *visual properties*, such as colors or line styles). The advantage is that XML and OWL documents are application independent.

Since graphs are created primarily for visual consumption (i.e., published online as GIF or JPG files), the information present in the OMs of graph-drawing applications is not complete. For instance, in our corpus of approximately 500 graphs taken from Statistics Canada's "The Daily" publication, the title property of the OM is *never* used. Instead, graphs have *text boxes* that perform the title function implicitly. This is acceptable for visual inspection of the graph, but for accessibility applications, this means that there is only *some* text at *some* location in the graph; in other words, no function, no semantics to it. Thus, after retrieving the graph representation coming from graphing applications, **iGraph-LITE** has about a dozen and a half algorithms that help *curate* this first representation. For a more detailed account of graph representation and curation, please see [10, 7].

To enrich the messages to be conveyed by the graph, after information extraction and curation, **iGraph-LITE** has

⁴See <http://www.inf.udec.cl/~leo/assets2010.html>

⁵<http://www.freedomscientific.com>

⁶<http://www.yourdolphin.com/>

⁷<http://www.sno.phy.queensu.ca/~phil/exiftool/>

⁸<http://www.w3.org/XML/>

⁹<http://www.w3.org/2004/OWL/>

Key	Function
Graph interaction commands	
	general description of the graph (same as Fig. 1)
	where (am I in the graph?)
	move right one point
	move left one point
	help
 + digit	Skip <i>n</i> points
Speech handling commands	
	increase the speed of speech
	decrease the speed of speech
	stop the TTS engine

Table 1: Original command set for the **iGraph-LITE navigator**

a plug-in infrastructure that makes it simple to add more semantics to the curated representation of previous stages. This works by simply taking the XML or OWL representations as input, and applying specialized algorithms (either declaratively enhancing the XML or by SPARQL¹⁰ and other rule-based systems). Examples of enriching algorithms may be the presentation of *visual slopes* for lines and a qualification of whether these slopes are *small*, *moderate* or *sharp*, or naming segments using *slope event* predicates such as *advance*, *double* or *plummets*.

For the NLG subsystem, **iGraph-LITE** is based on a string templating engine¹¹ built in-house that allows complex messages to be transmitted to the user (for a theoretical discussion of NLG and templates, see [15, 5]). Depending on the type of graph, available information in the enriched semantics, and other considerations, the NLG system generates a detailed description of the graph.

Finally, from requirement specifications sessions with blind participants, it was clear that much more than a simple static description was needed to make graphs truly accessible [8]. We therefore designed the **iGraph-LITE** navigation tool, inspired by command-driven interfaces like JAWS. At its core, the **iGraph-LITE** navigator queries the XML representation in particular ways (see Section 2.2). This paper deals with an evaluation study of the navigator.

2.2 Implementation of the initial commands in **iGraph-LITE** navigator

iGraph-LITE's navigator is designed to be used 'in focus'. While this is so, the interaction is driven by key presses. Thus, when a recognizable keyboard event occurs, queries are executed in order to find the relevant data in the tree representation of the graph, enriched by the execution of the concepts. Afterwards, the string template engine discussed above generates the final output text or speech.

Based on previous studies, 9 commands were implemented [9]. Five of these were used for graph interaction and querying, and the other five for general application management such as increasing the speed of the speech output, stopping

¹⁰<http://www.w3.org/TR/rdf-sparql-query/>

¹¹See, for instance, <http://velocity.apache.org/>

The line graph has the title "operating profits up slightly in second quarter". Quarterly values from 1996 to 2006 are shown. The values are given in billions of dollars. The lowest yearly values occurred in the second quarter of 1996 (26.7 billion dollars), and the highest values in the fourth quarter of 2005 (57.6 billion dollars).

Figure 1: A sample iGraph-LITE static description.

```
[igr]: I'm now looking at a line graph
entitled "Operating profits up slightly in
second quarter"
[usr]: d
[igr]: Ok, I'm describing the whole graph now.
[... as per the description shown above ...]
[usr]: w
[igr]: You're at the beginning of the graph.
The current point is the second quarter for
1996 and its value 26.7 billion dollars.
[usr]: →
[igr]: There is an increase between Q2 and Q3,
1996 from 26.7 to 27.9 billions of dollars.
[usr]: s 2 →
[igr]: Skipping 2 values. There is an
increase between Q3, 1996 and Q1, 1997 from
27.9 to 29.6 billions of dollars.
```

Figure 2: A sample iGraph-LITE interaction with the original set of commands.

the speech output, or quitting the application. Graph interaction may begin with the "description" (`[d]`) of the graph as shown in the following excerpt:

As can be seen in Figure 1, the description provides information about the title, the graph type and its purpose, as well as about both the axes and the scales. The fact that it is a single-line graph is readily inferred from the information. Users are able to navigate the graph using the commands summarized in Table 1. For instance, they can move back and forth one data point at a time by using the left and right arrow keys (`[←]` or `[→]`). Pressing `[s]` followed by a number, every subsequent `[←]` or `[→]` command will skip the specified number of points in the graph. Finally, if users have lost track of their current position, they can press `[w]` ("where am I"). In addition, a help command is provided. When the user presses `[h]`, iGraph-LITE reads out all other keyboard commands along with their functions. Figure 2 is a sample interaction with iGraph-LITE. Everything in lines starting with [igr];, the system says by TTS; everything in lines starting with [usr];, the user enters by keyboard.

3. FIRST FORMATIVE USABILITY STUDY

The purpose of Study 1 was to isolate and identify potential strengths and weaknesses of iGraph-LITE as well as to investigate the degree to which information contained in line graphs with a single line would be readily accessible for all possible users through the iGraph-LITE system.

3.1 Method

Participants.

Ten legally blind (4 congenital, 6 non-congenital) and ten sighted people matched to the extent possible in terms of age, gender, educational background and utilization of

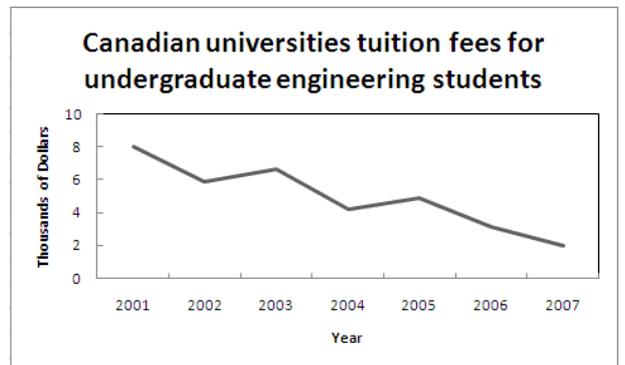


Figure 3: Actual graph stimulus from the evaluation study. Seven data points, downward trend.

graphs in their daily life, were recruited from existing contacts and through Carleton University's Paul Menton Centre for students with special needs. They were tested in the Carleton University Human Oriented Technology Lab (HOTLab), or in a quiet place convenient to the participant, in individual sessions lasting approximately one hour. Participants were offered \$20 for their participation. Throughout the experiment, key presses were automatically time-stamped via iGraph-LITE's key logging component on the experimenter's computer. Response time was calculated as the time difference between the start of a question and end of the answer to it. The number of commands used was analyzed from the key logging component, and performance was assessed by the number of correct answers to the experimental questions. The number of commands used to answer each question was also calculated.

Materials.

The study used a Dell Latitude D810 laptop computer supporting the iGraph-LITE software. Three questionnaires were prepared. The **graph familiarity questionnaire** sought information about the types of graphs participants were accustomed to work with, the frequency with which they used these, and if they were also asked to create graphs in their work. The **demographic questionnaire** sought information about education level, whether the participant had taken any mathematics or statistics courses, if they were legally blind, and, if so, if they were congenitally blind. The **post-test questionnaire** asked about ease of use and likeability of iGraph-LITE, as well as some open questions about the best/worst aspects of the software.

Five single-line stimulus graphs similar to single line graphs published in "The Daily", were created in Microsoft Excel (similar to Figure 3¹²). Of the four test graphs, two "simple" graphs contained three data points; two "complex" graphs contained six data points. The fifth "complex" line graph was used for pre-experimental practice to familiarize participants with the commands in iGraph-LITE. Five questions were asked for each of the four graphs. These covered global trend issues, patterns within the graphs, descriptive issues such as title, units of the axes, and requests for specific items of data (see Figure 2 for sample questions).

¹²The number of points for complex graphs differed for Study 1 (six) and Study 2 (seven). Otherwise, graphs were similar. The graph in Fig. 3 belongs to study 2.

Response times and executed keyboard commands were automatically recorded for later analysis. Participants' verbal answers were recorded manually.

Procedure.

All participants were given the same materials. Questions were read aloud; the researcher recorded responses manually. The participant was seated at the computer and presented with information about the practice graph upon pressing “h” (help), which broadcasts all nine available commands and the function of each (see Table 1). Participants were encouraged to familiarize themselves with all the commands, spending as much time, and repeating the commands as many times as they liked, until they felt comfortable with the system. Once they were ready, the four test graphs were presented, one at a time, and the five questions were asked for each graph, also one at a time. The graphs were presented in a Latin Square design, but questions were always given in the same sequence. Upon hearing a question, the participant attempted to answer it by navigating the graph using the keyboard commands generated by the **iGraph-LITE** text-to-speech engine (FreeTTS¹³). Upon completion of all 20 questions, the post-test questions were administered.

3.2 Results and discussion

Results from the pre-experimental data are presented first, followed by analysis of the performance data as well as of the response-time data. Reflections on the general command usage are then presented, and finally, the list of commands, extended as a consequence of the results, is presented.

3.2.1 Pre-experimental questions

The graph familiarity questions revealed that all 20 participants had used line and bar graphs before. Seven of the 10 blind participants encountered graphs less than once a week, and the three others, who were familiar with a range of graphs, indicated they encounter graphs several times a week. Only two of the 10 sighted participants encountered graphs less than once a week; the remaining eight participants said they encountered graphs almost every day. Only five of the sighted participants created graphs in their work, generally line or bar charts, and usually in Microsoft Excel or SPSS. The graph familiarity data confirmed that all participants had at least a basic understanding of line graphs to be able to use **iGraph-LITE** for its intended purpose. Eight of the 10 participants in each group had taken at least one mathematics or statistics course at college or university. Five sighted and two blind participants had taken more than five such courses.

3.2.2 Performance- and response-time data analysis

Performance data were analyzed by response time (in seconds), accuracy, and number of commands used to answer each question. As expected, all participants took significantly longer to answer questions for the complex graphs ($M = 40.52, SD = 46.43$) than for the simple graphs ($M = 26.33, SD = 30.79$), $t(398) = 3.60, p < .001$, and they used significantly more commands to answer those questions ($M = 8.66, SD = 11.43$) than for simple graphs ($M = 4.03, SD = 5.39$), $t(398) = 5.18, p < .001$. However, graph complexity did not affect performance accuracy, $p > .05$.

Interestingly, blind participants used nearly twice as many commands ($M = 8.19, SD = 11.60$) as sighted participants ($M = 4.50, SD = 5.41$), $t(398) = 4.07, p < .001$, regardless of graph complexity. The blind participants used the **Esc** command to stop the speech much more frequently than the sighted participants, probably as a result of their experience and proficiency with screen readers. Highly experienced screen reader users are extremely adept at listening to high-speed speech, and to decide very quickly if what they hear is relevant to the information they are seeking. Despite this significant difference in number of commands used between the two groups, there was no difference in the groups' response times ($p > .05$). Blind participants answered more of the graph questions correctly ($M = .91, SD = .29$) than sighted participants ($M = .83, SD = .38$), $t(398) = 2.35, p < .05$. Blind participants found the system easier to use ($M = 7.5/10, SD = 1.18$) than sighted participants ($M = 6/10, SD = 1.76$), and they liked using **iGraph-LITE** slightly more ($M = 6.8/10, SD = 1.23$) than sighted participants ($M = 5.3/10, SD = 2.63$). This is not surprising, since the blind participants were able to discern information more accurately, probably because they are more accustomed to relying on hearing and auditory memory than sighted participants. The blind participants reported that **iGraph-LITE** was superior to the means they normally use for graph interpretation (tactile graphs, others interpreting graphs for them).

3.2.3 Reflections on command-usage

None of the participants used the minimum number of commands needed to answer the questions, which had been calculated *a priori* for each stimulus graph. It was noteworthy also that, whereas participants tended to listen to the graph description to answer the first graph question, the description was not repeated when a later question could also have been answered by it. It is possible that participants forgot about the contents of the description once they had answered some of the questions. If that is true, then it could be advantageous to provide a separate command for each component in the description as well as keeping the description command (**d**).

Surprisingly, the “skip” command was used very rarely for reasons that are unclear. Instead, participants were observed moving forwards and backwards through the graph several times using the less optimal arrow keys to find a particular answer. It is possible that they may have felt more comfortable progressing step-by-step through the graph, perhaps because they feared getting lost. Alternatively, it may be unclear just exactly how the command works. For instance, skipping one (**s** + **l**) actually moves the cursor *two* data points (i.e., it skips one point). It is possible that, by re-naming the skip command to “move”, that may eliminate any confusion about the distance moved. Thus, “move 2” would achieve the same as “skip 1”. Furthermore, since the number of data points was not reported in the description, participants would not have known the appropriate number of points to skip until they had moved through the graph at least once. This could also have prevented them from using the “skip” command. Participants often began a search for an answer with a backward skip, even when they already were at the beginning of the graph. If they were unsure of their current location in the graph, they could have used the **w** (“where am I”) command yielding precisely

¹³<http://freetts.sourceforge.net/>

Question	Scope
What is the unit of values in the vertical axis?	B
In what year was tuition closest to \$6000?	L
How many years are reported in the graph?	G
What were the tuition fees in 2004?	L
What type of university program is this graph showing?	B
Based on the trend in the graph, would you expect that 2008's tuition would be greater or less than it was in 2007?	G

Table 2: Sample questions for the usability studies, where B stands for background, G for global and L for local. These are questions for the sample graph in Fig. 3.

that information. Yet, the “where am I” command was used very rarely, suggesting that it was either forgotten, or it was not found to be very useful.

Because of the tendency to use the description only when seeking the answer to the first question for a graph, it was decided to increase the number of commands to allow future participants to issue exactly the command that fits their current question. The description was therefore broken into single commands, while also leaving the description as is. The directional commands of left and right arrows were broken down to allow the user to move in either direction point by point but with the ability to choose which level of granularity they would like the information provided (i.e., full point description such as “increases from 3.2 percent in 2002 to 5.4 percent in 2003” versus just the point value “5.4 percent” versus just the slope direction “up”). Additional commands were created to increase efficiency in the graph navigation and accommodate expressed desires from the participants. These included commands to bring the user to the beginning/end of the graph as well as commands that will provide the highest/lowest points in the graph. The `[Esc]` command to stop speech was replaced with `[Ctrl]` to be consistent with most screen readers which typically use this key for that function, as indicated by the blind users in the study. Command selection was more ergonomically-based, relying on the spatial organization of the keyboard relative to finger placement and proximity, rather than selected based on mnemonic association alone (like those of `VIM`¹⁴); however, commands that did not inherently have some kind of spatial relevance, were assigned a mnemonic aid for the command (i.e., `[W]` was still “where am I”). Therefore, directional commands to move left were assigned keys left of the index finger on the right hand and those to move right were assigned to letters spatially located to the right of the index finger of the right hand. The extended command set shown in Table 3 was therefore implemented in preparation for the formal usability test.

4. SECOND FORMATIVE USABILITY STUDY

The purpose of Study 2 was to test the next iteration of the `iGraph-LITE` design, taking into consideration the

¹⁴<http://www.vim.org/>

Key	Function
Graph interaction commands	
<code>[x]</code>	help
<code>[d]</code>	general description of the graph
<code>[r]</code>	title of graph and axes titles with respective units of measurement and scales
<code>[a]</code>	go to the first point in the graph
<code>[f]</code>	go to the last point in the graph
<code>[h]</code>	move left one point (full point information)
<code>[l]</code>	move right one point (full point information)
<code>[j]</code>	move left one point (only point value)
<code>[k]</code>	move right one point (only point value)
<code>[n]</code>	Move left one point (only slope direction)
<code>[m]</code>	move right one point (only slope direction)
<code>[t]</code>	go to the lowest point in the graph
<code>[u]</code>	go to the highest point in the graph
<code>[w]</code>	where (am I in the graph?)
<code>[s]</code> + digit	Skip <i>n</i> points
Speech handling commands	
<code>[Ctrl]</code>	stop the TTS engine
<code>[p]</code>	increase the speed of speech
<code>[o]</code>	decrease the speed of speech

Table 3: Improved command set for the `iGraph-LITE` navigator

lessons learned from Study 1. Thus, once the extended command set had been implemented, the second formative usability test was conducted.

4.1 Method

Participants.

A new sample of 10 legally blind (6 congenital, 4 non-congenital) and 10 sighted people matched as in the first formative study from the same sources was recruited. They were tested individually in the HOTLab in two sessions, the first lasting approximately 10 minutes and the second session taking approximately 30 minutes. Participants were offered \$20 for their participation in the entire study and were paid at the end of the second session. Throughout the experiment, key presses were automatically time-stamped via `iGraph-LITE`'s key logging component on the experimenter's computer. Response time was calculated as the time difference between the start of a question and end of the answer to it. The number of commands used was analyzed from the key logging component, and performance was assessed by the number of correct answers to the experimental questions. The number of commands used to answer each question was also calculated.

Materials.

Four new single-line test graphs containing either four or seven data points were prepared in MS Excel. As before, the number of data points was used to indicate graph complexity: the two “simple” graphs contained four data points, and

the two “complex” graphs contained seven data points. For each graph, six questions were generated covering three main areas: global trend-related questions, local questions requiring information about individual data points, and background questions related to the topic of the graph. A fifth graph was prepared for pre-experimental practice as before.

Procedure.

Cued recall was tested in session 1, and session 2 was a free recall test as well as offering an opportunity to use the commands to answer questions about a new set of line graphs. In preparation for session 1, participants had learned the functions and commands of the entire command set in their own time. This was done to reduce the likelihood that less-than-optimal command usage in session 2 could be attributed to participants having forgotten some of the commands.

Upon agreeing to participate, the list of the 18 commands shown in Table 3 was emailed to participants together with explanations of the functions. Participants were instructed to learn these in their own time. Once they felt confident that they would be able to recall all commands and functions, they contacted the experimenter to arrange a phone session. For one half of the participants, the experimenter cited the commands in random order, asking the participant to explain the function of each. This was reversed for the other half, for whom the experimenter provided the explanation, requiring the participant to cite the relevant command. This process continued until all commands had been matched correctly with the explanations. The purpose was to identify any commands/functions that may be difficult to comprehend or recall. Participants were not introduced to **iGraph-LITE** in Session 1.

Session 2, designed to assess the degree to which **iGraph-LITE** would be used optimally was conducted approximately one week after Session 1. It was the actual usability test.

Using a free recall paradigm, participants first cited the commands and functions. If recall was less than 100%, the experimenter provided the explanation for the missing commands. These were then recalled again until the complete set had been recalled correctly.

Participants were then given the practice graph, spending as much time as they wanted to familiarize themselves with using the commands and listening to the synthetic voice in **iGraph-LITE**. They were encouraged to try all the commands. Although participants had memorized all the functions of all the commands, it was important for participants to experience practical application of these before commencing the experiment. When they were ready, they were given the first test graph which, as before, could not be seen. They used the keyboard for command entry and listened to the voice response. Commands were logged automatically to the same computer as in Section 4.

A Latin Square design was again adopted for presentation of the four test graphs, and questions were always given in the same sequence to each participant. When all questions had been attempted, the same post-experimental questions were administered as before, and participants were invited to make any comments before being debriefed.

4.2 Results and discussion

As before, results of the pre-experimental questionnaires are presented first, followed by the performance- and response-time analyses. Participants’ navigation strategies and re-

flection on participants’ comments are then presented and discussed.

4.2.1 Pre-experimental questions

Results from the graph familiarity questionnaire revealed that all 10 sighted participants and seven of the 10 blind participants had worked with line graphs before. Although six of the 10 blind and six of the 10 sighted participants indicated that they encountered graphs less than once a week, the graphs they did encounter were most often line- and bar graphs. This was also true for the remaining participants who reported encountering graphs more frequently during a typical week. Only three of the sighted participants said they were required to create graphs on a monthly basis.

The demographic questionnaire data revealed that the eight of the blind and only three of the sighted participants had taken at least one course in mathematics or statistics at college or university. Blind participants said they relied on screen readers or on others to describe graphs they encountered, or else they used tactile graphs. However, several participants reported that they know of no tool that can assist in graph accessibility.

4.2.2 Performance- and response-time data analysis

As graph complexity did not affect response times, the number of commands used, or accuracy, $p > .05$, data from all graphs were pooled in the following analyses using independent samples t -tests. As before, there was no differences between the two participant-groups for response time ($p > .05$) or for accuracy ($p > .05$) (blind participants: $M = .84, SD = .37$; sighted participants: $M = .85, SD = .35$), in contrast to the previous study in which the blind participants outperformed the sighted participants on accuracy. Both groups answered approximately 85% of the questions correctly, thereby clearly demonstrating their ability to explore and extract information on a global, local, and background level in **iGraph-LITE**.

As before, the blind participants used significantly more commands to answer the questions ($M = 7.13, SD = 11.03$) than the sighted participants ($M = 4.74, SD = 4.94$), $t(478) = 3.06, p < .01$. This difference was mainly due to the number of times the blind participants issued a redundant direction command when they were at either end of a graph ($n = 189$ times) compared with such commands issued by sighted participants ($n = 94$ times). Apparently, blind participants needed to reassure themselves of their current location more than sighted participants. The data are insufficient to show if blind participants lacking awareness of their current location in the graph, or if they lacked confidence in same. We are more inclined to interpret this redundant use of commands as confirming participants’ need to confirm their location rather than it representing a lack of awareness of their current location. This is because blind participants’ performance was equally accurate, and they took no longer than sighted participants to complete the tasks. If the blind participants were unsure of their current location in the graph, then we would have expected that they require have taken longer than the sighted people, or that their level of response accuracy would have suffered. None of these occurred. Because Apparently, blind people need to confirm their location in a graph compared to sighted people who were less inclined to “double-check” their location. This observation is consistent with studies showing that blind people need ref-

erence points for spatial orientation when exploring objects in space [17]. Many software application designed for visually impaired people provide haptic or sonification cues to convey landmarks, for example major grid lines, start and end lines, and so on (e.g. MultiVis¹⁵, or Phantom¹⁶). The observation suggests that **iGraph-LITE** should provide cues on current location at either end of a graph.

The post-experimental questions showed that both groups found **iGraph-LITE** easy to use (blind $M = 7.8/10$, $SD = 1.32$; sighted $M = 7.7/10$, $SD = 1.16$) and that they liked using **iGraph-LITE** (blind $M = 7.5/10$, $SD = 1.72$; sighted $M = 7.4/10$, $SD = 1.84$). Both these sets of ratings represented an increase compared with the first formative usability study, suggesting that the amendments to **iGraph-LITE** improved the system. However, recall that participants here were already familiar with the commands, in contrast to participants in Study 2. Blind users all agreed that **iGraph-LITE** was “much better” than the tools they normally use.

4.2.3 Navigation strategies

Analysis of navigation paths revealed that blind participants were three times more likely to move left data point by data point from any current location than their sighted counterparts when first attempting a question. This was true even if they were already located at the leftmost point. By contrast, sighted participants tended to move directly to the beginning of the graph, then move right point by point through the graph and directly back to the beginning again if they needed more information. This occurred twice as frequently as for the blind participants. As was found in Study 1 in which the blind participants found it difficult to construct a mental model of a graph when given the most recent data first, the sighted participants visualized the graph from left to right in that same fashion. Here, however, the blind participants typically adopted a step-by-step strategy instead of moving straight to the leftmost, oldest data point, suggesting that they may visualize a graph differently when they control the commands than when they request information from another person. The option of moving to the beginning of a graph was not available in Study 2, making a comparison of this command use impossible. Finally, blind participants were found to use the directional commands yielding the value of individual data points approximately three times more often than sighted participants. This confirms the intended benefit of the decision to break down the graph description into single elements.

5. GENERAL CONCLUSIONS AND FUTURE WORK

In this paper we discussed two user-based evaluations during the iterative design process of **iGraph-LITE**, an Assistive Technology (AT) that allows access to visual representations of numerical data using a natural language (NL) interface based on screen-reading software, albeit with more semantic and linguistic processing.

Perhaps most importantly, the studies presented in this paper showed quite conclusively (i.e., with good statistical power) that NL interfaces to graphs and charts are *usable* by legally-blind people, who could accurately answer com-

plex questions that a static textual description of a graph (and possibly also a sonified version) would not be able to provide. They also show that the blind community *want* to use these technologies and that people *feel* that technologies such as **iGraph-LITE** help provide access to graphical information. This paves the road for a profound impact in, for example, government implementation of accessibility to inherently visual information by, for instance, adding plug-ins such as the **iGraph-LITE** navigation tool to both commercial and free screen-reading software to access the proprietary graphs, and other types of diagrams.

Our studies also provide some hints about the appropriateness of recruiting sighted participants to take part in evaluation/usability studies for tools targeting communities with special needs. Due to the difficulty in recruiting blind participants, it is commonplace for experimenters to run pilot studies using blindfolded sighted people and then subsequently test a smaller group of blind participants to validate their initial findings. Although this approach is a popular one, it is certainly not an infallible strategy. For example, it was clear after analyzing our data that different strategies were implemented by blind participants as compared with sighted participants who could not see the visual objects, in our case, graphs. Blind participants used almost twice as many keyboard commands as sighted participants who were essentially “blindfolded” to the target visual object. Further, strategies employed by each group throughout the study differed considerably. For instance, blind participants seemed to navigate through graphs more conscientiously than sighted participants, confirming almost twice as many times their position in the target graph before proceeding. This means that even though they knew that they were at the beginning of the graph, for instance, they would still hit the move left command to double-check. If only sighted people had been tested, subtle issues like these could have been missed. This effectively means that if we added an additional piece of information telling them that they are either at the beginning or end of a graph at certain times during the interaction, the number of commands and task time would have been lower. Another important issue has to do with the evaluation of the system. If only sighted people were asked about the likeability/helpfulness of a given AT, they could give a misleading rating to the system. In our case, blind people gave the **iGraph-LITE** navigator a much higher rating on both indices (likeability and helpfulness) than sighted participants, and one may be able to speculate that the opposite is also true: it is possible that a certain AT can “make sense” to “blindfolded” sighted participants because of their mental models, while being useless to blind people. In summary, although using “blindfolded” sighted participants facilitates the evaluation phase of any given AT, results can be potentially misleading, and/or important information may be lost.

In terms of future work, several of the blind participants commented that they would prefer commands that are similar to other common speech programs. However, the tasks people are able to perform using conventional screen readers are so different from those carried out here that a perfect match would be impossible to achieve. Still, there is room for improvement to the mapping of **iGraph-LITE** commands to functions. Before attempting to improve this, it will be of interest to address another important issue: **iGraph-LITE** tasks performed thus far were designed specifically for the

¹⁵<http://www.dcs.gla.ac.uk/~johank/MultiVis-web/default.html>

¹⁶<http://www.sensable.com/>

usability tests even though the graphs did resemble graphs typically encountered in publications such as “The Daily”. From the above results, however, it is impossible to know how closely our questions resemble the kinds of questions that actual blind users might want to ask of graphs in their daily work. Even if they are similar, it is possible, for example, that the six directional commands in the command set tested should be reduced to a smaller number.

A final thought is that we believe that screen-readers are an under-utilized and under-researched technology in the sense that they may be able to work with much richer semantics and, hence, linguistic messages, and that these deep-knowledge interactive language technologies could provide knowledge of the navigational and, in general, cognitive strategies people use to access those pieces of information which they have more trouble with.

6. ACKNOWLEDGMENTS

We would like to thank Statistics Canada, in particular M. Lachance, L. Boucher and S. Bleeks for their unwavering support in time, resources and enthusiasm of this research. This work was supported by an IBM UIMA Innovation Award and a Proyecto de Insercion Postdoctoral PSD 57 del Programa Bicentenario de Ciencia y Tecnologia, CONICYT, Chile, to L. Ferres.

7. REFERENCES

- [1] I. Abu Doush, E. Pontelli, D. Simon, T. C. Son, and O. Ma. Making microsoft excel: multimodal presentation of charts. In *Assets '09: Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, pages 147–154, New York, NY, USA, 2009. ACM.
- [2] O. Ben-Tal, J. Berger, B. Cook, M. Daniels, G. Scavone, and P. Cook. Sonart: The sonification application research toolbox. In *Proc. ICAD, Kyoto, Japan*, 2002.
- [3] L. Brown and S. Brewster. Drawing by ear: Interpreting sonified line graphs. In *Proceedings of the 2003 International Conference on Auditory Display. ICAD, 2003*.
- [4] R. F. Cohen, R. Yu, A. Meacham, and J. Skaff. Plumb: displaying graphs to the blind using an active auditory interface. In *Assets '05: Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, pages 182–183, New York, NY, USA, 2005. ACM Press.
- [5] K. V. Deemter, E. Krahmer, and M. Theune. Real versus template-based natural language generation: A false opposition? *Comput. Linguist.*, 31(1):15–24, 2005.
- [6] S. Demir, D. Oliver, E. Schwartz, S. Elzer, S. Carberry, and K. F. McCoy. Interactive sight into information graphics. In *Proceedings of W4A 2010*, 2010.
- [7] M. Dumontier, L. Ferres, and N. Villanueva-Rosales. Modeling and querying graphical representations of statistical data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, In Press, Corrected Proof, 2010.
- [8] L. Ferres, A. Parush, S. Roberts, and G. Lindgaard. Helping people with visual impairments gain access to graphical information through natural language: The iGraph system. In *Proceedings of the 10th International Conference on Computers Helping People with Special Needs*, Lecture Notes in Computer Science. Springer-Verlag, 2006.
- [9] L. Ferres, P. Verkhogliad, G. Lindgaard, L. Boucher, A. Chretien, and M. Lachance. Improving accessibility to statistical graphs: the igrph-lite system. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 67–74, New York, NY, USA, 2007. ACM.
- [10] L. Ferres, P. Verkhogliad, L. Sumegi, L. Boucher, M. Lachance, and G. Lindgaard. A syntactic analysis of accessibility to a corpus of statistical graphs. In *W4A '08: Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*, pages 37–44, New York, NY, USA, 2008. ACM.
- [11] J. H. Flowers, D. C. Buhman, and K. D. Turnage. Data sonification from the desktop: Should sound be part of standard data analysis software? *ACM Trans. Appl. Percept.*, 2(4):467–472, 2005.
- [12] N. Grabowski and K. E. Barner. Data visualization methods for the blind using force feedback and sonification. In *SPIE proceedings series*, pages 131–139, 1998.
- [13] A. R. Kennel. Audiograf: a diagram-reader for the blind. In *Assets '96: Proceedings of the second annual ACM conference on Assistive technologies*, pages 51–56, New York, NY, USA, 1996. ACM.
- [14] D. K. McGookin and S. A. Brewster. Soundbar: exploiting multiple views in multimodal graph browsing. In *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction*, pages 145–154, New York, NY, USA, 2006. ACM.
- [15] E. Reiter. NLG vs. templates. In *Proceedings of the 5th European Workshop in Natural Language Generation*, pages 95–105, Leiden, NL, May 1995.
- [16] B. Riedel. Perception of gradient in haptic graphs: a comparison of virtual and hysical stimuli. In *Conference Proceedings to Eurohaptics*, 2001.
- [17] N. Sribunruangrit, C. Marque, C. Lenay, and O. Gapenne. Improving blind people’s spatial ability by bimodal-perception assistive device for accessing graphic information. In *In Proceedings of the 7th biannual Conference for the Advancement of Assistive Technology in Europe 2003*, pages 476–480, 2003.
- [18] W. Yu and S. Brewster. Comparing two haptic interfaces for multimodal graph rendering. In *HAPTICS '02: Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, page 3, Washington, DC, USA, 2002. IEEE Computer Society.
- [19] W. Yu and S. Brewster. Evaluation of multimodal graphs for blind people. *Universal Access in the Information Society* 2(2), 2:105–124, 2003.
- [20] W. Yu, R. Ramloll, and S. A. Brewster. Haptic graphs for blind computer users. In *Proceedings of the First International Workshop on Haptic Human-Computer Interaction*, pages 41–51, London, UK, 2001. Springer-Verlag.