



Using low cost devices to support non-visual interaction with diagrams & cross-modal collaboration

Metatla, Oussama; Martin, Fiore; Bryan-Kinns, Nick; Stockman, Tony

For additional information about this publication click this link.

<http://qmro.qmul.ac.uk/jspui/handle/123456789/5015>

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk

ISSN 2043-0167

Using low cost devices to support non-visual interaction with diagrams & cross-modal collaboration

Oussama Metatla, Fiore Martin, Nick Bryan-Kinns and Tony Stockman



EECSRR-12-03

June 2012

School of Electronic Engineering
and Computer Science

A glowing diagram of a network or system architecture with nodes and connecting lines, overlaid on a circular lens effect.

Computer
Science

A detailed view of a printed circuit board (PCB) with various components like a microchip, capacitors, and resistors, overlaid on a circular lens effect.

Electronic
Engineering

Using low cost devices to support non-visual interaction with diagrams & cross-modal collaboration

Oussama Metatla, Fiore Martin, Nick Bryan-Kinns and Tony Stockman

June 2012

1 Introduction

This report presents a series of explorations into the feasibility of using low cost devices to design support for non-visual interaction with diagrams. These explorations are a follow up to the Collaborative Cross-modal Interfaces project (CCmI)¹, which explored the potential of using multimodal input and output technologies (audio, haptics, graphics) to improve the accessibility of collaboration between visually-impaired and sighted individuals when using diagrams in the workplace. The main challenge addressed in CCmI was how to design support for collaboration where participants have differing access to modalities in order to increase workplace inclusion.

One of the main outcomes of CCmI was the development and release of an open source software for collaborative editing of nodes-and-links diagrams. The software augments a visual diagram editor with audio and haptic technology, combining hierarchical and spatial models of representation [1] to support simultaneous visual and non-visual interaction. Auditory interaction in this tool is supported by organising the content of a diagram into a hierarchical structure that users can interact with using the computer keyboard. The content of the diagrams is then displayed using a combination of speech and non-speech sounds. Haptic interaction is supported through the use of a PHANTOM Omni² haptic device, which allows a user to explore the spatial layout of a diagram by rendering its content accessible in a virtual haptic space using a mixture of haptic effects and auditory feedback. The tool, its design and evaluation are fully described in [2].

Whereas the integration of audio into the software tool is cost effective, the use of the PHANTOM Omni device - which at the time of writing this report costs over £1000 - could hinder the wide adoption of the developed software by user communities. The follow up project therefore aimed to:

- Explore the use of low cost alternatives to support non-visual spatial exploration of diagram content.

¹ <http://ccmi.eecs.qmul.ac.uk/>

² <http://www.sensable.com/haptic-phantom-omni.htm>

- Design and implement prototype interactive systems to exemplify the use of the alternative low cost devices.

Two alternative low cost devices were chosen for this project; a Wacom BambooTM tablet³ and a Novint Falcon[®] controller⁴. The choices were based on both the retail price range of these devices and on their potential to support spatial interaction with virtual content. The developed prototype designs for both devices addressed three aspects of non-visual interaction: 1) the representation of the spatial aspects of diagram content, 2) support for non-visual inspection of diagram content, and 3) support for active editing of diagram content. The following sections describe the design and implementation of the prototype interactive systems.

2 Wacom Bamboo Tablet

2.1 Device specification

The Wacom Bamboo device used in this project (Figure 1) is a small tablet which includes sensors for both pen and multi-touch input. The device is intended for use as a substitute for a mouse and as a sketching and markup tool. The tablet includes four customisable buttons on the tablet, and two on the pen, with dimensions of (WxDxH) 278mm x 176mm x 11mm and an active area of (WxD): 147mm x 92mm for pen input and (WxD): 125mm x 85mm for touch input⁵. At the time of writing, the device can be purchased at a price range of between £80 and £100.

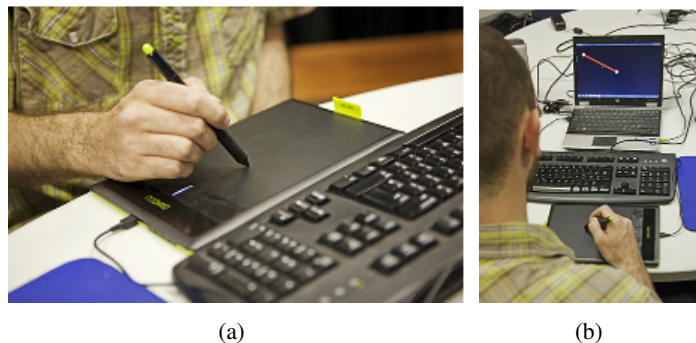


Figure 1: The Wacom Bamboo multi-touch tablet.

³ <http://www.wacom.com/en/Products/Bamboo/BambooTablets.aspx>

⁴ <http://www.novint.com/index.php/novintfalcon>

⁵ The full specification of this device can be found here: <http://www.wacom.eu/index2.asp?pid=294&spid=1&lang=en>

2.2 Design

We explored the potential of using pen interaction within the active area of the tablet to support non-visual spatial exploration of diagram content as follows:

2.2.1 Spatial representation

To capture its spatial layout, we transform a given diagram into a simplified representation that captures the essence of its content and arrangements, which we then reproduced on a new window (Figure 2 left). In this representation, diagram nodes are represented as dots and links are represented as connecting lines with varying styles. Since the active area of the tablet uses absolute positioning when in pen input mode, the coordinates of the captured content on the new window are reproduced on the tablet's active area. For ideal interaction, the new window is used in full screen in order for it to be calibrated to match the dimensions of the active area of the tablet.

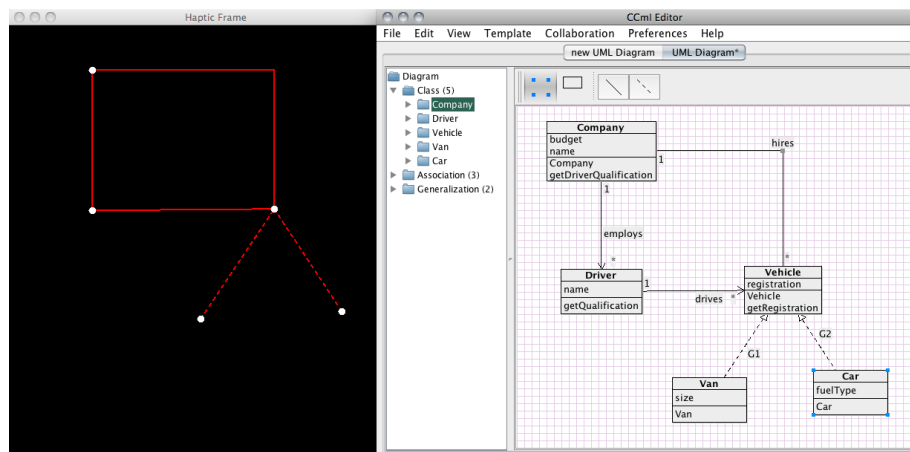


Figure 2: The graphical view of the diagram editor (right), a simplified representation of the diagram to support haptic interaction (left).

2.2.2 Non-visual inspection

Non-visual interaction with diagram content is then achieved by dragging the pen around the tablet's active area, a combination of speech and non-speech sounds is displayed when the position of the pen encounters that of items on the diagram. The auditory display techniques used in the CCmI editor (described in [1, 2]) were reused in this design to support non-visual auditory inspection of diagrams. For instance, contact with nodes is highlighted using a non-speech sound with a distinctive timbre that reflects the shape of the node. A node with a circular shape for example is displayed using the sound of bursting bubble, whereas that of a square shape is displayed using

a wooden sound. This is accompanied by a speech output displaying the label of the encountered node. A similar technique is used for the auditory display of diagram links.

2.2.3 Active editing

Additionally, the user can alter the spatial positions of diagram elements by dragging a node from one location to another, or bending a link at a given point on the line. This is achieved by using the available pen buttons, where clicking once on the front button picks up the item (or a point on a link) and clicking it again drops it at a new location. Moving diagram elements in space was augmented with auditory feedback in a similar way to that which was used in the original CCmI prototype as follows: Three distinct auditory icons are used to highlight that an item has been successfully picked up (a short sucking sound), that it is being successfully dragged in space (a continuous chain-like sound) and that it has been successfully dropped in the new location (the sound of a dart hitting a dartboard). This is described in more details in [2].

2.3 Notes on interaction design & implementation

2.3.1 Interaction notes

While the tablet supports both pen and multi-touch interaction, pen interaction should be used for the prototype design described above. This is because the design is based on the ability to capture the absolute positioning of diagram elements on the tablets active area as mapped from the onscreen window. The absolute positioning mode is not supported when multi-touch interaction is enabled.

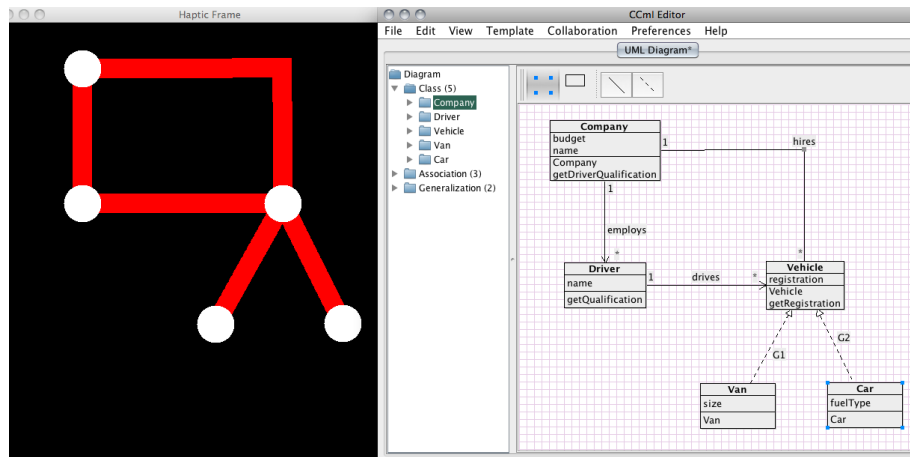


Figure 3: A magnified version of the simplified representation of the diagram to improve orientation.

One of the main issues that we encountered when using pen interaction in this manner is orientation. In the CCmI tool, the PHANTOM Omni device was used to generate haptic feedback using magnetic fields to stimulate attraction to diagram elements. That is, the user is automatically

attracted to a given element when they move the device's stylus in its proximity. This meant that a user, for instance, is able to easily trace a line without deviating from it. Since there is no tactual or haptic feedback that could be readily stimulated with the tablet pen, the user must rely on the auditory feedback they receive to be able to trace a connection between two nodes. For instance, by dragging the pen around the active area of the tablet to scan for diagram elements.

A number of strategies could be used to successfully inspect diagram content in this manner. For example, scanning the active area can be done in a zigzag fashion or following a spiral pattern to discover content or trace lines. This proved difficult, however, particularly since the size of the nodes was small and connecting lines were thin. We designed a simple alternative approach to work around this issue where diagram elements were magnified to produce graphically thicker objects (from 2 to 26 for links, and from 10 to 50 for nodes⁶) thus increasing the reactionary area where auditory feedback would be triggered when employing different search strategies (see Figure 3). Initial tests showed that orientation issues persist, however, and an alternative design is required in order to address this issue. An example of this is the use of auditory beacons to guide the user towards diagram elements by using an amplitude mapping where the sound of a given element increases as the pen moves closer to its position. An alternative design could use ambient sounds that alternate in display to emphasise whether the pen is on or off a diagram element.

2.3.2 Implementation notes

There are a number of configurations specific to using the stylus pen on the Bamboo tablet that need to be taken into account when implementing the prototype design described above. The tablet has two buttons to simulate the left and right click of a mouse. However, touching the tablet with the stylus is also equivalent to a left click. Hovering on the tablet without actually touching it is equivalent to moving the mouse, while moving the stylus around while touching the tablet is equivalent to moving the mouse around whilst holding its left button. Touching the tablet can thus be detected in Java by using a `MouseEvent` listener and implementing the `mouseDragged()` method, while hovering can be detected by implementing the `mouseMoved()` method.

Left-clicking the pen turns the mouse motion from moving to dragging. In the prototype design described above, the labels of diagram elements are displayed when the mouse is on the location of that element and the left click is pressed. We used `mouseDragged()` to make a text-to-speech engine utter the labels.

We use the right click for picking up and dropping objects as described above. However, a right click on the tablet disables its left click. In other words, moving the pen when touching the tablet is equivalent to moving the mouse while holding the left button thus triggering the `mouseDragged()` event, but issuing a right click nullifies the left click which was held down. This had two consequences of the implementation of the prototype: First, after issuing a right click, the method called by the Java window toolkit is no longer a `mouseDragged` but a `mouseMoved`, and so the spoken

⁶These numbers are the float values passed as arguments to the `BasicStroke` constructor which is defined as follows: The line width is the thickness of the line measured perpendicular to its trajectory. The line width is specified as a float value in user coordinate units, which are roughly equivalent to 1/72 of an inch when the default transformation is used.

label of diagram elements cannot be uttered after pressing the right button. Second, the right button nullifies the left click on the pen when clicked and so it is not interpreted as an actual right click. The first right click therefore will not pick up the diagram element as desired and another click is required in order to pick up the element in question.

We used the following workaround in order to address the former issue. An instance of `java.awt.Robot` was used to automatically issue a left click each time a right click occurs. The method `robot.mousePress(LEFT BUTTON)` is in fact called in the `mouseMove()` method so that once the user right-clicks on a diagram element to either pick it up or drop it, then a left click is pressed automatically when they move away from that element. The `robot.mousePress(LEFT BUTTON)` is called in the `mouseMove` rather than in the `mousePressed(RIGHT BUTTON)` method in order to prevent the label of the item to be displayed each time the user grabs it or drops it.

Note that this workaround introduces another, though less serious problem. After picking up/dropping objects, un-touching the tablet will not trigger any `mouseReleased()` call. This will in turn cause the text-to-speech engine to display the label of an item even when hovering over the tablet without touching it. This effect is cancelled when the tablet is touched and untouched again. Thus, the following occurs when an item is picked up and/or dropped: 1) the right click issued to pick up the item nullifies the left click, 2) the robot re-clicks the left button as soon as the cursor moves away from the current location, and 3) because a left click is performed by the robot, un-touching the pen is no longer considered as a mouse release, and so hovering over the tablet (without touching it) triggers the text-to-speech output.

3 Novint Falcon

3.1 Specification

The Novint Falcon device is a game controller designed to replace the mouse or the joystick. The device consists of a robotic handle with six degrees of freedom and incorporating a number of motors to allow it to simulate various haptic effects such as textures, shape, weights and dynamics. The user interacts with the device by holding its grip (see Figure 4), which contains a number of buttons, and by moving it in a three-dimensional space to experience virtual haptic effects⁷. The device is, for instance, typically used for first person shooter games where users can feel the recoil of shooting. At the time of writing, the retail price range of this device is between £150 and £200.

3.2 Design

We followed a similar approach to the one described for the Wacom Bamboo tablet to support spatial exploration of diagram content using this device as follows:

⁷ The full specification of this device can be found here: <http://www.novint.com/index.php/novintxio/41>

3.2.1 Spatial representation

A simplified representation of a given diagram is reproduced on a separate window as shown in Figure 2 where diagrams nodes are represented as dots and its links as connecting lines. Since the Falcon device can stimulate a variety of haptic effects, the diagrams nodes and links are rendered with an attraction force that allows the user to feel the nodes and trace their connections in space. Diagram elements which are encountered through the Falcon cursors are also augmented with a combination of speech and non-speech sounds following a similar approach as described in Section 2.2.2 above.



Figure 4: The Novint Falcon game controller device.

3.2.2 Non-visual inspection

Since haptics effects can be readily simulated using Falcon device, we were able to reproduce the same design of the non-visual interaction as implemented in the CCmI editor (described in [2]) as follows: Diagram elements are rendered as magnetic points on the virtual plane such that a user manipulating the grip of the Falcon device in proximity of an item is attracted to it. This is then augmented with an auditory feedback which is triggered upon contact of the virtual cursor and the item. The user can thus trace the grip across a line without deviating away to other parts of the plane. Contact with links is also accompanied by earcons with distinct timbres, and the labels of encountered nodes and links are also displayed in synthesised speech upon contact.

In addition to representing diagram content using various haptic effects, we were also able to reproduce the implementation of two modes of interaction used in the original CCmI editor and which we refer to as *sticky* and *loose*. In a sticky mode of interaction, the magnetic attraction forces of the diagram elements are increased to make it harder for the user to snap away from them. This simulates an impression of being “stuck” to the diagram content and thus one can trace its content by following the connections from point to point. In a loose mode of interaction on the other hand, the attraction forces are decreased such that a user can freely move around the virtual space without necessarily having to be in contact with any diagram content.

3.2.3 Active editing

The user has the ability to move nodes and bend links in the virtual space. Again, this can be achieved by locating an item or a point on a link on the virtual plane, clicking on the central button of the Falcon grip to pick it up, dragging the grip to another point on the plane, then dropping it in a new desired location with a second button click. We also followed the same approach to augment the drag and drop feature with auditory and haptic feedback as is the case in the original CCmI editor. First, three distinct auditory icons are used to highlight that an item has been successfully picked up (a short sucking sound), that it is being successfully dragged in space (a continuous chain-like sound) and that it has been successfully dropped in the new location (a dart hitting a dartboard sound). Second, a haptic spring effect is applied, linking the current position of the grip to the original position of where the item was picked up from. This allows the user to easily relocate the item to its original position without losing orientation on the plane. Once an item is picked up, the user is automatically switched to the loose mode of interaction to allow for free movement while still able to inspect encountered items.

Finally, we also reproduced the implementation of the synchronisation mechanism that allows the user to switch between the haptic and auditory hierarchical views of a given diagram. That is, the user can locate an item on the hierarchy then issue a command on the keyboard which would cause the Falcon device to move the grip and locate that item on the haptic plane. If the user is holding the grip, they are then essentially guided to that location. Similarly, the user can locate an item on the virtual haptic plane then issue a command on the keyboard to locate it on the hierarchy.

3.3 Notes on interaction design & implementation

3.3.1 Interaction notes

One of the main difference between the design of the haptic interaction with diagrams when using the PHANTOM Omni and the Falcon device is in the virtual projection plane where diagram content is arranged. In the original design, the diagram content is displayed on a virtual vertical plane to match the exact orientation of the graphical view of a diagram on a computer screen. This was not the case in the prototype design using the Falcon device, where the user can freely move the grip of the device in a three-dimensional space without being restricted to a vertical plane. This also means that the combination of auditory and haptic effects that are triggered when the cursor of the grip encounters diagram elements are triggered regardless of where the grip is located on the Z axis of the virtual space so long as its X and Y coordinates matches those of the encountered item.

Another difference between the two devices is related to the haptic effects used to represent the links of a given diagram. In the original CCmI design, the line styles of a links were represented using a friction effect which simulates a different texture for solid, dotted and dashed lines. That is, the user is able to feel the roughness of a line when tracing it using the PHANTOM Omni device, which increases from smooth for solid lines to medium and very rough for dotted and dashed lines respectively. The friction effect was not available in the Falcon device, which meant that such detailed haptic representation of diagrams links could not be reproduced. While different auditory

icons are used to reflect the style of a given line and accompany the haptic effect, the lack of the friction effect might have an impact of the ability of the users to differentiate between different styles of lines.

In relation to synchronisation mechanism which was implemented to guide the user from one location to another, we observed that adding an auditory cue to signal that the movement of the grip is about to occur could increase the usability of this feature. In particular, the addition of an auditory cue could act as a reminder to the user to alter the force by which they are holding the grip of the Falcon, thus making it easier to follow it to the new location. Additionally, switching the interaction mode from sticky to loose, then to sticky again upon landing on the new location can help avoid the possibility of bouncing away from it.

3.3.2 Implementation notes

As is the case for the PHANTOM Omni device, the implementation of the prototype design for the Falcon device was done via a native code (.dll), in this case using the HAPI library. This is a free library that can be used to manage a number of haptic devices - including the PHANTOM Omni device. The graphic representation is implemented in OpenGL. The haptic rendering of diagram content was realised through the HapticShapeConstraint effect, which is an effect that constrains the grip of the Falcon device in an area that is bound by a shape. The grip of the Falcon device differs from the stylus of the PHANTOM Omni in a number of ways:

1. The contact of the grip with the nodes and lines is less precise, e.g. it exhibits a jiggly behaviour when on a given item that is not present in the PHANTOM Omni device.
2. In general, the attraction forces used to represent nodes and links is also less precise. For instance, it is possible to specify a magnetic value that can apply directly on point and which can then be felt with the PHANTOM Omni stylus when it is in its proximity. Increasing this value will only increase the magnetic force at the specified location in the virtual space without affecting the ability of the user to freely move around the surrounding space. On the other hand, specifying a higher value for the attraction force on the Falcon device seems to also affect the surrounding space of the specified coordinates. Thus, applying a low magnetic value on a given point impacts the “sticky” mode of interaction, while a higher value renders the exploration in the “loose” mode awkward.
3. In order to allow the user to browse the lines smoothly, it was necessary to render the magnetic force of the points weak. This was done in order to avoid the situation where the magnetic force of points dominates the space, making it difficult to snap away from them. This in turns means that in the absence of lines, the haptic effects of nodes within a given diagram are hard to perceive.

Another important issue to note when implementing interactive haptics with the Falcon device using the HAPI library is that it does not include a “touch callback system”. Touch callbacks makes it possible to register a function to be called upon touching/ un-touching a given object. A simple

work around can be implemented by exploiting the looping nature of the thread used for rendering haptics (i.e. OpenGL animation runs continuously on a loop to render each frame). This is achieved by querying the position of the Falcon grip at each frame from the HAPI library and calculating the distance between the grip and each point and line. If the distance is less than a constant threshold, then it can be assumed that the grip is in contact with an object. The touching/un-touching system is particularly crucial for managing the auditory cues that augment the drag and drop feature of the non-visual interaction. It is also important to note that there a number of features of the Falcon devices which were not exploited in the prototype design described above, and which could be provide potential avenues for future extensions. An example of this is the *viscosity effect* and the *time function effect* which can control the continuity of a given force and can thus be use to improve the usability of the synchronisation mechanism.

4 Summary of feasibility

Wacom Bamboo tablet. Our implementation of the prototype for this device shows that it is feasible to use the Wacom Bamboo tablet device to support non-visual interaction with diagrams. A number of modifications from the original CCmI tool are necessary in order to accommodate the particularities of this device, however. Namely, issues related to non-visual orientation within the virtual space and the management of the pen input mode. While the latter can be accommodated through implementation workarounds, the former might prove difficult to address, and thus the issue of orientation would need careful design and testing to ensure that it is addressed properly.

Novint Falcon. The fact that the Falcon device provides the ability to stimulate haptic effects makes it a more suitable low-cost device to support non-visual spatial interaction with diagrams. A number of modification would still need to be applied in order to make up for the lack of certain features, such as the use of explicit auditory feedback to make up for the missing friction effect and the use of auditory cues to manage the synchronisation mechanisms between the different views of a diagram.

References

- [1] O. Metatla, N. Bryan-Kinns, and T. Stockman, “Interactive hierarchy-based auditory displays for accessing and manipulating relational diagrams,” *Journal on Multimodal User Interfaces*, 2011.
- [2] O. Metatla, N. Bryan-Kinns, T. Stockman, and F. Martin, “Cross-modal collaborative interaction between visually-impaired and sighted users in the workplace,” in *Proceedings of the 18th International Conference on Auditory Display (ICAD2012)*. Georgia Institute of Technology, Atlanta, GA, USA, 2012.