Intuitive NUIs for Speech Editing of Structured Content (Work in Progress)

Marina Isaac[1], Eckhard Pflügel[2], Gordon Hunter[3], and James Denholm-Price[4]

[1] Faculty of Science, Engineering and Computing, Kingston University
M.Isaac@kingston.ac.uk
[2] Faculty of Science, Engineering and Computing, Kingston University
E.Pfluegel@kingston.ac.uk
[3] Faculty of Science, Engineering and Computing, Kingston University
G.Hunter@kingston.ac.uk
[4] Faculty of Science, Engineering and Computing, Kingston University
J.Denholm-Price@kingston.ac.uk

Abstract. Improvements in automatic speech recognition, along with the growing popularity of speech driven "assistants" in consumer electronics, indicate that this input modality will become increasingly relevant. Although good functionality is offered for word processing applications, this is not the case for highly structured content such as mathematical text or computer program code. In this paper we combine the principles of natural user interfaces with the concept of intuitive use, and adapt them for speech as the input modality in the context of editing content displayed on a screen. The resulting principles are used to inform design of the user interface of a specialist language editor for spoken mathematics.

Keywords: natural user interface, NUI, spoken mathematics, speech control

1 Introduction

Using speech as an input modality has been available since the 1980s but is not yet a mainstream form of human-computer interaction. Due to recent improvements in the capabilities of automatic speech recognition (ASR), and introduction of speech driven "assistants" in consumer electronics, this type of interface is likely to become increasingly relevant.

ASR products such as Nuance's Dragon[1] have existed for many years, and provide great functionality in the context of word processing, allowing the user to dictate content in a variety of widely spoken languages. Such developments have not been applied to specialised languages used to describe structured content such as mathematical text or computer program code. Because of their formatting and punctuation, these are not served well by standard document editing facilities, and so even experienced ASR users have great trouble working with such content.

The problem of enabling casual users (that is, those not expert in LaTeX or similar languages) to create properly formatted mathematics is well known, and we hope that spoken mathematics[2] will help in this area. The difficulty with the spoken approach is that the use of standard ASR software imposes a high enough cognitive load on the user (through having to recall the command language) to present its own challenge.

The concept of the natural user interface (NUI) has so far been applied mainly to input via touch and gestures (Wigdor & Wixon, 2011). In this paper we consolidate the general NUI guidelines with those pertaining to "intuitive use" (Naumann et al., 2007) of interfaces, to provide a list of intuitive NUI principles.

After considering what might be regarded as "feeling natural" in a speech-driven environment, our contribution is to adapt these principles specifically to a speech controlled environment, and consider how they may be used to enhance the speech based mathematical expression editor *TalkMaths* (Wigmore, Hunter, Pflügel, & Denholm-Price, 2009).

2 Natural User Interfaces

The concept of the natural user interface was first developed by Fjeld, Bichsel, and Rauterberg (1998, 1999) to describe a user environment with minimal discontinuity between the physical actions required to complete a task

[1] http://www.nuance.com/dragon/index.htm

[2] In this context, the term "spoken mathematics" refers to that which would be dictated by a native English speaker.

and the user's internal problem solving process. The idea is based on the activity cycle[3] in action regulation theory (Hacker (1994), as cited by Fjeld et al. (1999)), and the fact that for optimal task performance, users need to be able to perform epistemic as well as pragmatic actions (Fjeld et al., 1998)[4]. Epistemic actions are treated as fundamental by Fjeld et al. (1998), who put the ability to engage in these at the top of their original list of NUI design guidelines. To give users the confidence to behave in this exploratory way, the negative effect of making any mistakes needs to be minimised – the second guideline. The third guideline is to permit the user to employ as much of their body as possible as well as their voice in interactions (Fjeld et al., 1998). This last guideline requires system observation of the complete user environment, including interaction with artefacts such as visual projections (Rauterberg, 1999) and, as a somewhat ambitious requirement, this remains purely the subject of experimental UIs, no longer appearing in the list of design principles later presented by Fjeld et al. (1999).

Developments in touch screen and gesture technology have motivated researchers to investigate the resulting opportunities in user interface (UI) design (Wigdor, Fletcher, & Morrison, 2009). Wigdor and Wixon (2011) present a practical guide for designers, including ways in which many of the ideas of Fjeld et al. (1999) may be implemented.

Jetter, Reiterer, and Geyer (2014) also address the area of NUIs in their Blended Interaction framework. This uses conceptual blending as defined by Fauconnier and Turner (2008) along with the image schemas[5] described by Hurtienne and Israel (2007) in an effort to predict which metaphors will be easy for users to understand (Jetter et al., 2014). It is worth noting that the blends suggested by Jetter et al. (2014) incorporate the metaphors involved in image schemas, which themselves reflect the language used to describe relations and actions (Hurtienne & Israel, 2007). Although both areas of work relate to direct manipulation (typically using hands), the fact that language lies at the core of the ideas may prove useful to designers of speech UIs.

The original descriptions of NUIs refer to the property of an interface being "intuitive" (Fjeld et al., 1998, 1999). Blackler and Hurtienne (2007) describe intuitive use as taking advantage of the user's existing knowledge of comparable situations, to make aspects of an interface seem familiar to them; Naumann et al. (2007) add the condition that the user should almost be unconscious of the fact they are operating a UI. One of the benefits of this lack of awareness by users will be a lower cognitive load associated with using the interface (Naumann et al., 2007), specifically with carrying out the low level actions required to achieve a goal in the activity cycle.

The following general principles combine the ideas of Fjeld et al. (1999) with the other work on NUIs and the above definitions of intuitive use. They are grouped according to general objective of NUIs.

Encourage epistemic actions and exploratory behaviour. As proposed by Fjeld et al. (1998), doing so will help the user complete their task efficiently, exhibiting the exploratory behaviour that will help them progress to expertise in the application (Wigdor & Wixon, 2011, p.55).

1. Users with proficiency levels ranging from novice to expert should feel comfortable using the software (Wigdor & Wixon, 2011, p. 13).
2. The interface should provide alternative ways of invoking functionality for different classes of user, as well as employing other types of redundancy such as using both text and icons to describe controls (Blackler & Hurtienne, 2007).
3. Interaction with the system should feel robust to the user, so that they have the confidence to attempt new operations.
   - For major changes or destructive actions, the system should require confirmation from the user and provide previews where appropriate (Wigdor & Wixon, 2011, p. 55).
   - The system should minimise the impact of user errors (Fjeld et al., 1998) by allowing the user to reverse them easily (Fjeld et al., 1999).

The user should feel that their interaction with the system is *intuitive*. There is considerable overlap between the concepts of intuitive use and naturalness. The following principles pertain to intuitive use.

4. Where conventions have been established in the application area or medium (Wigdor & Wixon, 2011, p. 13), adhere to these, otherwise use appropriate metaphorical devices to represent objects or actions (Blackler & Hurtienne, 2007).
5. Take advantage (where appropriate) of users' existing skills, to make their experience feel more familiar (Wigdor & Wixon, 2011, p. 13).
6. Facilitate the planning aspect of the activity cycle by indicating the current state of the software and the available actions at all times (Wigdor & Wixon, 2011, p. 45) (Fjeld et al., 1999).
7. Show the results of all user actions (Fjeld et al., 1999), with feedback being immediate, appropriate (Blackler & Hurtienne, 2007) and informative. Non-trivial feedback (for example, system messages) should increase user understanding of the system and provide appropriate help where needed (Wigdor & Wixon, 2011, p. 56).

[3] The repeated steps of goal setting, action planning, performance and evaluation, taken by an individual to accomplish a task.

[4] *Pragmatic* actions are those that bring a task physically closer to completion, while *epistemic* actions are performed primarily to aid mental processing (Kirsh & Maglio, 1994).

[5] Abstractions that reflect the way humans relate objects in the real world.

8. Clear affordance[6] in the design of controls will help the user identify both their functions and modes of use (Fjeld et al., 1999; Blackler & Hurtienne, 2007) (Wigdor & Wixon, 2011, p. 55).
9. The interface should be compatible with the user's mental model of the system (Blackler & Hurtienne, 2007).

Context of use should be taken into account. The design should reflect:

10. the nature of the user's task rather than the technology of the application (Blackler & Hurtienne, 2007), and also
11. the physical environment and social context in which the system is used (Wigdor & Wixon, 2011, p. 19).

3 What Feels "Natural" in a Speech Interface?

While natural language may seem to be the ideal choice for casual use of or novice support for an interface, not only is the current technology too immature for serious application, but speaking "wordy" sentences to describe repetitive actions is not desirable for most users. Research shows that brief commands are preferred to natural language sentences for such tasks (Elepfandt & Grund, 2012; Stedmon, Patel, Sharples, & Wilson, 2011), suggesting a language that is superficially simple but still allows for more complex commands may go a long way towards providing a natural feeling experience.

The approach of Wigdor and Wixon (2011) reflects a natural progression from manipulating objects on screen using keyboard and mouse (or other pointing device) to a subjective feeling of direct manipulation using the same extremities. Although it may be tempting to develop a "typing assistant", we believe this would be unsatisfactory because such a proxy could not give the user the feeling of manipulating the content themselves. If instead we ask how objects on screen may be manipulated using voice commands, the user should be less aware that they are having to go through an intermediary.

A major challenge for speech control is how to indicate which objects are to be manipulated, and how to transform them. One promising method for object selection is to use eye gaze as an adjunct to speech (Elepfandt & Grund, 2012). The findings of Kaur et al. (2003) and Maglio, Matlock, Campbell, Zhai, and Smith (2000) – that a user's gaze is naturally directed towards the object they wish to manipulate just before they issue a command – suggest that the use of this modality may contribute to the overall efficiency of such an interface. As Sibert and Jacob (2000) acknowledge, when gaze is used in this way (rather than as the main method of interaction), the interface is in fact making use of natural human behaviour and so partially meets the third original guideline for NUIs (Fjeld et al., 1998). In addition to its "naturalness", eye gaze has been shown to enable objects to be selected more quickly than by mouse (Sibert & Jacob, 2000), so might be expected to become a popular means of interaction when the technology matures. Until then, other means are needed to refer to objects on screen, for example the type of grids described by Wigmore (2011), Nuance's mouse grids, and the context-sensitive mouse grids described by Begel (2005). Rather than use numbers to index all non-word content, we propose meaningful labels, where possible, on the grid, to make selection easier. This may also help in frequently repeated sequences of actions, where using a label name will be easier than locating a label number, and may be particularly pertinent for "semantic grids" (Wigmore, 2011), where the labels would reduce the user's cognitive load by eliminating the need to recall terms such as "numerator". Use of such labels for semantic grids may also facilitate learning of mathematical concepts, an area investigated by Attanayake, Hunter, Denholm-Price, and Pfluegel (2013).

4 Redesigning *TalkMaths* to Reflect NUI Principles for Speech

The principles for intuitive NUIs are adapted for speech editing environments, and illustrated by application to *TalkMaths*[7] (Wigmore, Hunter, Pflügel, & Denholm-Price, 2009). This is a web based tool whose purpose is to allow users to enter and edit mathematical expressions using speech or keyboard input, and was created as the result of work initially carried out by Wigmore, Hunter, Pflügel, Denholm-Price, and Binelli (2009) when investigating the use of speech input for creating and editing structured documents. In this system, spoken commands are used to dictate mathematical expressions, including common mathematical symbols and operators, as well as to edit content. The numbering of the suggested modifications below follows that used in Section 2.

1. *Users with varying proficiency should feel comfortable using the software.* A mechanism should be in place to remind novice users of the basic commands, alongside a means of making them aware that more complex versions are available. This should help them in getting the feeling of gaining mastery of the command language (rather than continually having to resort to a help system), thus forming part of the "scaffolding" described by Wigdor and Wixon (2011, p.53). A command history screen area could show completed commands, which may also help novice users learn the language. Experienced users should be able to give more than one command in a single utterance. As with the command language, it may be desirable to hide the

[6] See item 8 in Section 4.

[7] We assume use of an ASR product such as those offered by Nuance or Microsoft, for recognition of English speech, and that the user is not severely visually impaired.

full size and complexity of the content language from novice users. For example, casual users will not want to be overwhelmed with the large number of mathematical symbols and operators expected by researchers. This requirement could be addressed either by showing only the most popular words by default, or by using a visual device to make the popular ones more prominent.

2. *Allow functionality to be invoked in different ways.* Because our discussion concentrates on a single modality, there is some overlap between this guideline and the previous one. While experts may give one or more entire commands in one utterance, novice users may need to build them up interactively. It should be possible to customise commands, perhaps changing specific words to ones less likely to be misrecognised given the usage environment, or create commands that replace a frequently used phrase with a single word (Fateman, 2013). Because a speech interface has words at its heart, rather than provide icons with text that is hidden by default, the reverse approach may be taken, using icons only where appropriate to guide the user's eye to the text reminder. Where command reminders are used, explanatory words may be included in these, that do not need to be spoken as part of the command, and which are ignored if included in an utterance.
3. *Users should not be afraid of making mistakes.* Bearing in mind users' preference for briefer utterances, previews should be shown at the same time as requests for confirmation. As well as confirmations and previews, it should be possible to use the command history as a means of undoing changes made. Syntax errors in user commands should be handled in a way that minimises the amount of further user input required.
4. *Follow established conventions, and use appropriate metaphors.* The system should respond to popular conventions in ASR software, for example, "What can I say?" or, "Scratch that". It may be possible to arrive at appropriate metaphors by considering the points made by Jetter et al. (2014) regarding conceptual blends[8] (Fauconnier & Turner, 2008) and image schemas (Hurtienne & Israel, 2007).
5. *Allow users to exercise existing skills.* In addition to helping the user learn to use the software, this may boost their confidence by giving them a feeling of prior familiarity with the command language. Permitting vocabulary customisation will help by allowing the user to employ terms belonging to their discipline, or use "shorthand" from their working environment.
6. *Always indicate system state and available actions.* Areas on the screen may be used to indicate the system's state, for example to distinguish between the task of providing missing information for a command issued but not executed, and that of editing a command recalled from history. Because all actions may be invoked using spoken commands, only those command reminders that are appropriate for the context should be displayed as being *sayable* (see principle 8).
7. *Give appropriate feedback for all user actions.* Because of the lack of haptic feedback provided with speech control, a mechanism should be used to indicate the fact that the user's input has been detected (Wigdor & Wixon, 2011, p. 45), to allow for any delay in the processing of this input. (This is to avoid the spoken equivalent of the phenomenon of clicking a button on a web page multiple times – not through impatience, but because the user thinks a click event may not have been registered.) The delay may be particularly noticeable when using non-incremental speech recognition[9]. In such cases it may be useful to show the progress of the input handling, perhaps to indicate completion of initial speech recognition, parsing, and processing of the command itself. This is in addition to the usual feedback one would expect.
8. *Clear affordances.* Here, there are two levels to the idea of affordance: (1) the traditional notion of an input control indicating its function, and (2) a control indicating how it may be used, for example by clicking or typing in text. The loss of "tooltip" text[10] in touch-screen GUIs has caused a resurgence in controls whose functionalities are a mystery until activated. This is one case where the addition of text to an icon would not only inform users of its meaning, but provide a cue for what to say to activate it. Given that in WIMP[11] interfaces, even text controls are selected by clicking, the equivalent of 'clickable' in the context of speech input would be '*sayable*'. Command buttons could be labelled with the text of the command (with perhaps an additional brief description), while other controls could be labelled with appropriate names, that may play a role similar to nouns in a command. This approach is already used by ASR software to enable the user to select, for example, fields within a form or follow a hyperlink, but a consistent style to indicate *sayable* may make it easier for the user to recognise sayable objects as such.
9. *Compatibility with the user's mental model.* The user should be able to understand the structure of the objects they are manipulating (at the task level). For example, if a user is provided with a choice of alternative parses of a spoken instruction (rather than a list resulting from probabilistic predictions), this fact should be made clear.
10. *Reflect the nature of the task rather than the technology.* The software needs to be compatible with an environment where the user may want to combine various input modalities, and provide different input styles – one mode might allow the user to build up an expression as they think about a mathematical problem, while another would be optimised for fast input of hand-written notes.

[8] The integration of two or more apparently unrelated or incompatible notions to form a new idea that draws parts of its meaning from both.

[9] That which requires the entire utterance to be complete before it is interpreted.

[10] Brief help text displayed when the pointer is moved over a control.

[11] Windows, Icons, Menus and Pointers.

11. *Work within the environment of the user.* The software may be used in a noisy or public environment, in which case a user who is able to do so may want to use all modalities except speech input or output. Appearance and vocabulary of the interface may need to be adapted to the user's social environment – for example, professional mathematicians may want a very different interface from students who occasionally need to include mathematical text in an electronic document.

4.1 Additional Requirements Specific to Speech-based Specialist Language Editors

There are a number of other issues that need to be considered when designing the interface of editors for content described by a specialist language, that we summarise very briefly here.

Cursor replacement An alternative means is needed to specify the insertion point for new content, or for selection of content for editing. The fact that we are working with what could be viewed as a "random access" modality gives us the opportunity to allow a richer specification for navigating through code, using its natural structure as well as exploiting eye gaze tracking technology when this becomes feasible.

Handle incomplete commands It would be useful to handle errors either in the commands or recognition of content by allowing incomplete content to be specified. This will also allow users to give just vague descriptions to some parts of a structure, as suits their way of working.

Deal with ambiguous commands The system should have a strategy in cases where alternative parses may be obtained for what the user has said.

Concatenability The interface should permit the user to include more than one command in a single utterance.

Permit multiple utterance commands The system should allow commands that are too long to be said in one breath to be broken into several utterances.

Restriction of vocabulary It should be possible to limit the vocabulary for the ASR to words that are appropriate within the context.

5 Conclusion and Further Work

We have compiled a list of general principles for natural user interfaces optimised for intuitive use, and adapted them for the modality of speech control in the context of developing an editor for content described by a formal language. These have enabled us to suggest a number of modifications to the interface of our example editor *TalkMaths*. This system uses an operator precedence grammar that includes mixfix[12] operators, so that the command and content languages may be described in the same way (Attanayake, 2014). We hope that by describing their language using such a grammar, that many types of structured document could potentially be handled by future versions of the editor, including computer programs.

Our next step in the process is to implement the proposed design changes in the *TalkMaths* system and test it for usability by comparing user experiences of the original and new interfaces.

References

Attanayake, D. R. (2014). *Statistical language modelling and novel parsing techniques for enhanced creation and editing of mathematical e-content using spoken input* (Doctoral dissertation, Kingston University, Kingston-upon-Thames, UK). Retrieved from `http://eprints.kingston.ac.uk/29880/` (Accessed on 30/05/2015.)

Attanayake, D. R., Hunter, G., Denholm-Price, J., & Pfluegel, E. (2013). Novel multi-modal tools to enhance disabled and distance learners' experience of mathematics. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, *6*(1), 26-36.

Begel, A. (2005). Programming by voice: A domain-specific application of speech recognition. In *AVIOS speech technology symposium–SpeechTek West.*

Blackler, A. L., & Hurtienne, J. (2007). Towards a unified view of intuitive interaction: definitions, models and tools across the world. *MMI-interaktiv*, *13*(2007), 36–54.

Elepfandt, M., & Grund, M. (2012). Move it there, or not?: The design of voice commands for gaze with speech. In *Proceedings of the 4th workshop on eye gaze in intelligent human machine interaction* (pp. 12:1–12:3). New York, NY, USA: ACM. doi: 10.1145/2401836.2401848

[12] Operators involving more than one symbol or word, e.g. function notation.

Fateman, R. (2013). *How can we speak math?* Retrieved from `http://http.cs.berkeley.edu/~fateman/papers/speakmath.pdf` (Accessed on 21/05/2013.)

Fauconnier, G., & Turner, M. (2008). *The way we think: Conceptual blending and the mind's hidden complexities.* New York: Basic Books.

Fjeld, M., Bichsel, M., & Rauterberg, M. (1998). Build-it: an intuitive design tool based on direct object manipulation. In *Gesture and sign language in human-computer interaction* (pp. 297–308). Berlin & Heidelberg: Springer.

Fjeld, M., Bichsel, M., & Rauterberg, M. (1999). Build-it: a brick-based tool for direct interaction. *Engineering Psychology and Cognitive Ergonomics (EPCE)*, *4*, 205–212.

Hacker, W. (1994). Action regulation theory and occupational psychology: Review of German empirical research since 1987. *German Journal of Psychology*, *18*(2), 91–120.

Hurtienne, J., & Israel, J. H. (2007). Image schemas and their metaphorical extensions: intuitive patterns for tangible interaction. In *Proceedings of the 1st international conference on tangible and embedded interaction* (pp. 127–134). ACM Press.

Jetter, H.-C., Reiterer, H., & Geyer, F. (2014). Blended interaction: understanding natural human-computer interaction in post-WIMP interactive spaces. *Personal and Ubiquitous Computing*, *18*(5), 1139–1158. doi: 10.1007/s00779-013-0725-4

Kaur, M., Tremaine, M., Huang, N., Wilder, J., Gacovski, Z., Flippo, F., & Mantravadi, C. S. (2003). Where is it? Event synchronization in gaze-speech input systems. In *Proceedings of the 5th international conference on multimodal interfaces* (pp. 151–158).

Kirsh, D., & Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, *18*(4), 513–549.

Maglio, P. P., Matlock, T., Campbell, C. S., Zhai, S., & Smith, B. A. (2000). Gaze and speech in attentive user interfaces. In *Advances in multimodal interfaces – ICMI 2000* (pp. 1–7). Springer.

Naumann, A., Hurtienne, J., Israel, J. H., Mohs, C., Kindsmüller, M. C., Meyer, H. A., & Hußlein, S. (2007). Intuitive use of user interfaces: defining a vague concept. In *Engineering psychology and cognitive ergonomics* (pp. 128–136). Springer.

Rauterberg, M. (1999). From gesture to action: Natural user interfaces. *Mens-Machine Interactive: Diesrede 1999*, 15–25.

Sibert, L. E., & Jacob, R. J. (2000). Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 281–288).

Stedmon, A. W., Patel, H., Sharples, S. C., & Wilson, J. R. (2011). Developing speech input for virtual reality applications: A reality based interaction approach. *International Journal of Human-Computer Studies*, *69*(1), 3–8.

Wigdor, D., Fletcher, J., & Morrison, G. (2009). Designing user interfaces for multi-touch and gesture devices. In *CHI '09 extended abstracts on human factors in computing systems* (pp. 2755–2758). New York, NY, USA: ACM. doi: 10.1145/1520340.1520399

Wigdor, D., & Wixon, D. (2011). *Brave NUI world: designing natural user interfaces for touch and gesture.* London: Elsevier Science Inc.

Wigmore, A. M. (2011). *Speech-based creation and editing of mathematical content* (Unpublished doctoral dissertation). Kingston University, Kingston-upon-Thames, UK.

Wigmore, A. M., Hunter, G., Pflügel, E., Denholm-Price, J., & Binelli, V. (2009). Using automatic speech recognition to dictate mathematical expressions: The development of the "Talkmaths" application at Kingston University. *Journal of Computers in Mathematics and Science Teaching*, *28*(2), 177–189.

Wigmore, A. M., Hunter, G. J., Pflügel, E., & Denholm-Price, J. (2009, September). TalkMaths: A speech user interface for dictating mathematical expressions into electronic documents. In *2nd ISCA workshop of speech and language technology in education (SLaTE 2009)* (p. P3.4). International Speech Communication Association (ISCA).