The influence of design techniques on user interfaces: the DigiStrips experiment for air traffic control

Christophe Mertz (+°), Stéphane Chatty (+), Jean-Luc Vinot (*)

(+) CENA
7 avenue Edouard Belin
31055 Toulouse, France

(*) CENA
1, rue Champagne
91200 Athis-Mons, France

(°) CR2A-DI
32, rue des Cosmonautes
31029 Toulouse, France

{mertz,chatty,vinot}@cena.fr

ABSTRACT

Graphical user interfaces have limitations in terms of the information bandwidth they provide between users and systems. This can impede the redesign of systems previously based on more physical media: information may be less appropriately displayed, and shared cognition between users can be reduced. However, in parallel with research on new user interaction techniques, a more systematic use of visual design techniques can relieve those limitations. This article explores some of those techniques and how they can be applied, through a design experiment. Virtuosi and DigiStrips are two user interface prototypes developed within a research program on air traffic control workstations, which make use of touch screens and served as a basis for research on the use of graphical design techniques in user interfaces. This paper describes the lessons learnt in that experience and argues that techniques such as animation, font design, careful use of graphical design techniques can augment the possibilities of user interface design and improve the usability of systems. We finally analyse the possible enhancement brought in the communication between system and users as well as between the users.

Keywords:

Interaction bandwidth, touch-screen, animation, graphical design, feedbacks, gesture recognition, mutual awareness, air traffic control.

INTRODUCTION

More than ten years ago, many air traffic operators have set as a goal the evolution of air traffic control from the current tools to fully computerised environments. But apart from the inevitable technical difficulties, the process has been considerably slowed down by the difficulty to manage the transition in terms of tools and working methods. This is especially true in France, where the current paper strip boards have allowed controllers to build strategies that are hard to match with other systems.

Assuming that this transition is a suitable goal (which is not the topic of this article), there are two kinds of difficulties. The most apparent is to design a set of computer-based tools that allow controllers to detect and monitor conflicts between aircraft. The mere transposition of strip boards has failed to support efficient working methods because of the constrained interaction styles that the use of computers imposed. This forced researchers to imagine new tools to support new methods: Erato [leroux98] is a major achievement of that approach. The other difficulty is to reproduce the flows of information that are supported by the current system. First information flows from the users to the system: they need to manipulate tools and to take notes. Information also flows from the system to the users: notification of events, coding of aircraft parameters, etc.; the information that has to be coded even increases with the new systems, which generally introduce new data and events. And finally, information flows between users: distributed cognition [hutchins98] between controllers plays an important role in the working methods, and the current system supports it in many subtle ways. Designing user interfaces that support those flows is a challenge. The input bandwidth of graphical user interfaces is physically restrained by the devices used: see for instance [accot98]. Such interfaces also usually restrain the output bandwidth to that provided by computer displays, which in turn is often restrained by a limited use of the graphical capabilities of computers.

There have been attempts to improve those bandwidths by using new technologies, whether in general or specifically in the field of air traffic control: research on input techniques [mertz97] or on visual techniques such as animated alarms [athenes00]. In this paper, we relate another approach, that is a series of design experiments aimed at exploring how far the bandwidth limit can be pushed using available technologies, and how this can be achieved. We use those experiments to propose a set of design guidelines that can help building interfaces.

In this article, we first describe the context of this research: the Toccata demonstrator developed at CENA. After mentioning related work in the related fields, we present the four technologies we used, and elicit associated guidelines or properties: visual design, animation, touch screens, and gestures recognition. We categorize the associated gains according to four kinds of communication: input, output, between users through the system, and between users outside the system.

Finally we present some implications for the design of air traffic control workstations.

CONTEXT: THE TOCCATA PROJECT

Most work done in the 1990s on air traffic control workstations was grounded in the belief that copying a few features from desktop computers would be enough to ensure success. Those features can be summarised by the WIMP acronym: Windows, Icons, Menus and Pointing. Most designs tended to gather all displays on a very large vertical screen, and to use a mouse and menus for interaction on that screen. Contrasting with that approach, CENA's research program Toccata gathers a series of research projects aimed at understanding more deeply the human-computer interaction issues in workstation design, while exploring a larger range of technologies. Those projects are federated in the Toccata workstation demonstrator.

The current version of the demonstrator is based on a more traditional architecture than the WIMP workstations: a large screen for displays, and a horizontal area for work aids. This implies the use of touch screens, and part of our work in Toccata is aimed at finding how such screens can be best used: what kind of information to manipulate, what relationships with the information displays, and what interaction styles? This article focuses on the last issue of that list, and is illustrated by two recent prototypes which address that issue: Virtuosi is an alternative to menus for data input and note taking, and DigiStrips is a revisiting of the old problem of electronic stripping.

RELATED WORK

There is little literature on designing for touch-screens. [shneiderman93] contains good descriptions of the differences between touch screens and WIMP interfaces. [meyer95] describes issues raised by pen computing, which is often closely related to touch screens. [chatty96] has explored the use of pen computing on touch screens and gesture recognition for air traffic control.

In the more general field of graphical interface design, there are three approaches to the problems of interaction efficiency. The most fundamental approach is that of performance evaluation, modelling and prediction. Models like GOMS, Fitts' law, or the steering law [accot98] provide solid foundations for understanding the mechanisms of interaction, as well as hints for choosing designs. A more practical approach is that of ergonomical guidelines and rules [vanderdonckt99], which are derived from experience but often complex. Finally, there is a body of knowledge that has been present for decades, especially for information display: that of visual design techniques. Even though user interface design is more and more seen as a design discipline and there is more an more evidence of that in products, there is little literature on how design techniques can be applied. An excellent exception is [mullet95]. We take that approach, and in the rest of this article, we report about the lessons learnt and the design guidelines we derived from our experience, starting with visual design.

GRAPHICAL DESIGN AND ANIMATIONS

Appropriate fonts and good graphical design can increase displayable information:

By using dedicated and carefully designed fonts and by precisely composing the graphic components, it is possible to display more information while making it more legible. For instance, one of the findings of DigiStrips is that it is possible to display almost all the printed information of French paper strips for up to 30 strips on a 20" screen.

Figure 1: Current selected instructions displayed in Virtuosi

Texture or color gradation can code information:

Interfaces usually only used infills, lines and texts and they did not consider using textures or color gradation. One reason may be the limited number of colors (usually 256) available on most 28" screens, though this is now outperformed even on home computers. [graham97] states "that colors may reduce the ability to build a traffic picture". This seems very strong a statement and in fact it is probably influenced by the use of large infills of saturated red as conflict coding. Similarly, [cardosi99] claims that "the number of colors assigned a different meaning should be limited to six". But such a statement is often interpreted in an abusive way, limiting the total number of colors available on the screen. Colors can be used in many other ways than coding, though.

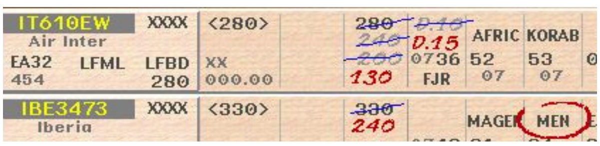

Figure 2: Strips with hand written font for 'annotations'

For example, in Virtuosi (figure 1), selected current instructions appear clearly, on the foreground, thanks to color gradations. This color-coding is relatively self-explanatory and should require little training and little cognitive effort to remember. Similarly, in DigiStrips a different texture codes the zone where the user can handle the strip and move it (approximately the left third of the strip, see figure 2). This subtle coding is enough to remind the user the limits of the grip zone.

Different fonts can convey information:

ATC system designers are often reluctant to use fonts to code information. For example it may be a bad idea to code the "assume control" state of a flight with two fonts or with their slant or bold attributes. This type of coding may indeed be difficult to memorize and even to recognize. However, it is possible to distinguish system-computed data and user input data through the font. For the former, we used "computer fonts" and for the latter we use legible "hand-written fonts". Such a coding appears easy to perceive, understand, and remember. In figure 2 for instance, it is easy to detect with a glance which flights have been given clearances.

Similarly, [marais99] mentioned that a value named Transfer Flight Level (TFL) should be distinguished when it is a standard pre-computed value from when a controller modified it after a negotiation with the following sector. Coding it with a "script-like" font seems a good choice. Other levels of coding can probably be imagined, as fonts can convey a number of subtle yet well perceived nuances. For instance, Downlinked Aircraft Parameters could be distinguished from radar data on a radar display by using a “digital-watch” font.

Animations

Like graphical design helps displaying more information, animations in interfaces are useful in expliciting state changes. They can be used to explicit the result of an action as on the Macintosh desktop when the user sends a document into the trashcan. They can reveal events like the arrival of an email in your mailbox. They can even complete an action or give an indication of job advancement. Animations can also be used for alarms [athenes00] (we will not discuss use of animation for alarms in this paper). Animations can be very useful in an ATC interfaces, and we will give some animation examples from DigiStrips and Virtuosi.

Animation can facilitate transitions:

The user can move, push or shift DigiStrips flight strips. When a strip is moved between two others, they move altogether to clearly display the result through animation (see figure 3).

Users thus clearly perceive the result of the action instead of reading, memorizing, reading again, and comparing call signs of electronic strips, which is what they need to do without animations. Such animations allow users to quickly detect what is wrong in case of minor interaction errors. The observations we did during the performance evaluation (described later in this paper) demonstrated it. Similar animations can also be used for managing messages or flight lists.

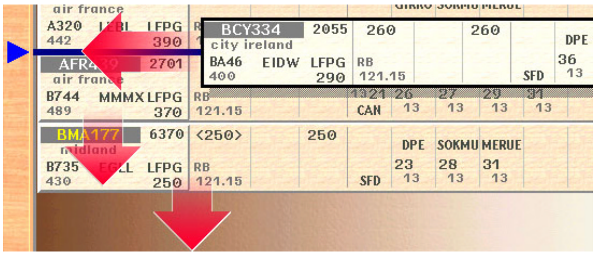

Figure 3: Animation following the insertion of a strip

Animation improved menu opening or closing:

[marais99] reports that users miss feedback at the end of input with a menu. Users felt sometimes unsecure about which flight in the radar display the menu applied to. In DigiStrips we tried some short and simple animations when menus open (figure 4) or close (figure 5). This has the following advantages:

- the menu-opening animation re-inforces the feedback on which flight the input will apply: opening begins in the strip, and the menu grows from this position;
- a menu can hardly open unnoticed (due to an involuntary "click" on the touch screen). The user has every chance of seeing it with his or her peripheral vision;
- at closing, the menu shrinks toward the strip on which the input applies. This feedback helps the user perceiving he or she did not mistakenly input data on a wrong flight;
- at closing, when the menu shrinks, the colored selected value move (during the animation) towards the modified field. This helps confirming to the user both the input value and the information;
- if the user cancels the input (by touching the screen outside the menu) the menu "explodes" to clearly state that no input has been made. This avoids ambiguity about whether there has been a mistake or not.

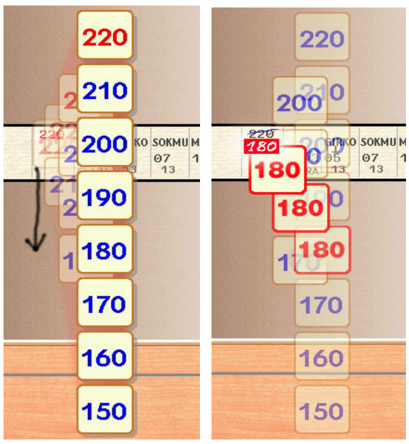

Figure 4 and 5: Menu opening and closing in DigiStrips

Animations can enhance scrolling menus:

In most WIMP interfaces, menus used to enter data are scrollable. For example the Cleared Flight Level opens on the current flight level and the controller can scroll up or down if the value he wants to input is not displayed. Scrolling is instantaneous. We used animations in a similar case in Virtuosi, where the horizontal bars are like big, horizontal, scrollable menus. The left and right arrows let the user scroll around if necessary, and the values move right-bound or left-bound. They accelerate and then decelerate to show a new set of values. The user can even interrupt the scrolling animation if he sees the desired value moving. We also use a bouncing animation to notify the user when the scrolling bar reaches its end.

Animations can notify events:

Currently with paper strips, the controller is notified by the slight noise of the printer. In WIMP interfaces, new strips, new messages or new elements in flight lists are barely notified to the user. In DigiStrips, we animate the new strip when it arrives. It moves from right to left at the bottom of the touch screen (figure 6). The user notices it seamless.

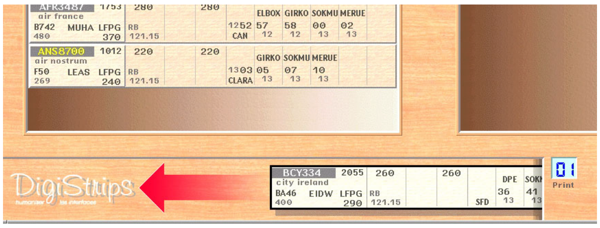

Figure 6: Arrival of a new strip in DigiStrips

Animations can make explicit the effect of an unfinished action:

In DigiStrips we used animations to show what will happen if the user releases a strip over the trashcan. As soon as the strip is close enough from the trashcan, slight collapsing rectangles show that the strip will be trashed if released.

Animations can help when system reaction time is too long:

With complex computerized tools, it is not always possible for the interface to give immediate feedback of success or failure of a request. Another CENA team in charge of the development of the interface of such a tool added animation (lasting approximately 2-3 seconds) when the system answered, usually some seconds after a request. If the user was not looking at the interface she had a chance of noticing this animation and the result in her peripheral vision.

Animation can enhance mutual awareness:

Controllers usually work by two or three (when there is a student controller). There are even sometimes four or more controllers working together (for instance during stormy weather). The importance of the system's ability to support this collaborative work is often underestimated. Animations can help by making more explicit to the other ones what a controller is doing. Though we do not claim that DigiStrips has the same properties as paper strips regarding mutual awareness, animation does make a difference with other designs.

DESIGNING FOR TOUCH SCREENS

Designing for WIMP interfaces or for touch screens is different in many aspects. First the size of the finger pulp does not allow the same precision than a mouse. Then you are limited to the equivalent of a single button. But to balance that, you gain the ability to use gestures and you avoid too small graphical buttons, costly even with the mouse [mertz97]. A good guideline when designing for touch screen is to have in mind the objectives of pointing with any hand and using the application even when standing up. We will now give some advantages and some rules usable when designing touch screen based interfaces.

Why using touch screen

Touch screens favor gestures:

Touch screens are direct interaction devices [baber98]: you point directly on the graphical object, on the screen, not via an indirect peripheral like a mouse or a trackball. This means that the user may be able to interact with less visual attention in a semi-blind mode. He has to look at the screen where the target is located, and then he can point his finger on this target without tremendous visual attention. This is just impossible with a mouse.

Touch screens favor mutual awareness:

Because you see what your colleague is doing with his hand on a touch screen you get many cues on his activity. A controller can also point on the screen to show something to his college. He does not necessary need a dedicated "show" function, which was sometimes implemented in WIMP interfaces.

Touch screens can be shared between users:

If you worked together with a colleague in front of your desktop, you must have noticed how difficult it is to share a mouse. With touch screens this becomes possible. Current technology does not allow simultaneous interaction on the same touch screen, but it is possible to share it and interact alternatively.

Designing for touch screen interfaces

Consider take-off strategy:

When designing for touch screens, even graphical buttons should behave differently than they do in the "classical" UI toolkits. Potter in [shneiderman93, p.161] proposes a new touch strategy called take-off, which is more precise (but may be slower) than 'traditional' clicks. The take-off strategy allows the user to drag a finger on the screen and the button is selected when the finger is lifted from the touch screen (in UI toolkits the button selected is the one touched at first contact, but if the finger moves out no button is selected). We used this take-off strategy for the selection of a value in Virtuosi value bars and in DigiStrips menus.

Use very simple gesture recognition:

In the past we studied pen computing and gesture recognition applied to ATC [chatty96]. But designing letter-like or digit-like strokes that are easy to draw by every user, easy to remember and easy to recognize is difficult if not impossible with current technologies. However it is possible to implement simple and efficient stroke recognizers if the strokes are simple enough like those used in marking-menus [tapia95]. This can dramatically improve touch screen based interfaces.

Different gestures can open different menus:

Data input was obtained in DigiStrips by combining simple gesture recognition (similar to markup menus) to open different menus. For example, to open a Cleared Flight Level menu, we use vertical gestures drawn on the flight level strip (figure 4). With gestures it is even easy to selected lower level values (downward straight stroke) and higher levels values (upward straight stroke). To input headings, we use horizontal strokes. A leftbound stroke opens a turn-to-left menu and a rightbound stroke opens a turn-to-right-menu. This dialog is quick and easy to remember.

Touch screen permits writing:

Many different touch screen technologies allow the use of a stylus (either passive or active depending on the technology). This allows an easy implementation of free writing input, without any recognition. The controller can write any special information and associate it to a flight, either to support his memory, or to benefit the changeover or to send it through the system to the following sector. We currently demonstrate this function in Virtuosi (see top right of figure 1)

Combining graphics and animations with touch screen:

We also think that animations and graphical design is important for touch screen based interfaces. Many informal users of Virtuosi and DigiStrips expressed a strong feeling of touching something more real, less abstract, and more concrete. We are now convinced that merging these techniques dramatically enhance interfaces, but how is it possible to prove it through experiment in the ATC field?

Performance evaluation

We have compared the performance of users executing a manipulation scenario with DigiStrips on a (Cathodic Ray Tube) touch screen and with a mouse [mertz99]. We did a performance (time-to-complete) and manipulation error typology analysis (not detailed in this paper). The experiment showed a significant ($p<0.0001$) performance difference between mouse and touch screen. The latter performed 10% to 14% quicker than the mouse, even if subjects were not used to drag objects around a touch screen (but they were all "expert mouse users"). All but one (out of 8 subjects) were significantly quicker with the touch screen. The compared times include some manipulation errors corrections. Such errors were slightly more numerous with the touch screen (90 vs. 75 with the mouse) but even so, touch screens were quicker. This probably means that better trained touch screen users would probably perform even better. Finally, some touch screen manipulation errors can easily be reduced, like parallax errors (14 out of the 90 manipulation errors), with the use of flat screens.

ENHANCING COMMUNICATION

We just described the gain expected by using animation, graphical design, touch screen and gesture recognition. Most of these gains are related to communication bandwidth. The following table summarizes the advantages they provide in terms of communication. The first two columns are input and output communication between system and one user. The two last columns are information flows between multiple users of a system, either through the system or directly between the users. For example, animations can help controllers perceiving and understanding seamless what their colleague is doing. Animations can also help controllers perceive system notification primarily send to one of their colleague.

	System input	system output	output to other users	between users
graphic design		X		
animations		X	X	X
touch screen	X			X
gestures	X			X

IMPLICATIONS FOR ATC WORKSTATIONS

Beside the positive feedback obtained from air traffic controllers during the qualitative evaluations, Virtuosi and especially DigiStrips have triggered enthusiasm in the French community. However, ATC has a long history of research prototypes acclaimed by the users who evaluate it, which turn to be not so good when brought to the real world. This is probably due to the relative simplicity of the evaluations that are usually carried compared to the many levels at which a system has to be good (interaction level: usability, group work level: flexibility, task level: completeness, scalability to degraded conditions, etc.). We thus have to be very careful when analyzing the consequences of this research.

Design

The clearest lesson learnt is about the design methods and techniques used. The DigiStrips experience confirms that careful graphical design has a positive influence on the usability and acceptability [khaslavsky99] of a system. And secondly, our informal evaluations confirm the efficiency of animations for improving feedback, and thus the usability of an interface and safety of the associated human-computer system. Though most of these findings will be hard to prove formally, we consider the current evidence sufficient for generalizing these techniques.

Workstation and interaction techniques

The very positive feedback on DigiStrips and Virtuosi hint that the architecture we chose for Toccata and the interaction techniques we proposed for air traffic control are worth studying further. In particular, we are confident about the high potential of touch screens for adding future work aids to ATC workstations. However, we are very conscious that such preliminary evaluation results are not enough for us to recommend the implementation of Virtuosi or DigiStrips as they are. These applications are still incomplete, and the chosen design might prove inadequate when adding the functions that are currently missing in the prototypes. In addition, though strips in DigiStrips look much like paper strips, it would be inconsiderate to believe that the working methods of controllers would remain the same, and that DigiStrips is thus the ideal replacement for paper strips [mertz00].

CONCLUSION

In this article, we have described the Virtuosi and DigiStrips prototypes of user interfaces for air traffic control. We have used those descriptions to support our arguments about the importance of professional visual design in user interfaces for ATC, and about the possibilities of touch-screen based interaction. We have proposed a set of guidelines for design such interfaces. Finally, we have discussed the impact of these techniques on communication and the potential consequences of this research. In the future, we plan to work on devising more formal evaluations of the techniques we described, as well as on exploring more of the questions we have raised about the design of alternative workstations for air traffic control.

ACKNOWLEDGEMENTS

Daniel Etienne developed Virtuosi and had bright ideas on visual design too. He also recoded and enhanced DigiStrips. Many people gave their time and ideas and inspired DigiStrips.

BIBLIOGRAPHY

[accot99] J. Accot, S. Zhai. Performance evaluation of input devices in trajectory-based tasks: An application of the steering law. In *Proceedings of CHI'99*, Addison Wesley, 1999. 466-472

[athenes00] S. Athènes, S. Chatty, A. Bustico. Human factors in ATC alarms and notifications design: an experimental evaluation. *ATM'2000 conference*. http://atm-seminar-2000.eurocontrol.fr/

[baber97] C. Baber. Beyond the desktop, designing and using interaction devices. Academic Press, 1997.

[cardosi99] K. Cardosi, D. Hannon. Guidelines for the Use of Color in ATC Displays. DOT/FAA/AR-99/52. June 1999. National Technical Information Service, Springfield, Virginia 22161 USA.

[chatty96] S. Chatty, P. Lecoanet, C. Mertz. Toward pen-based interaction with ATC tools. ATC Quarterly, 1995 3(4), 205-228.

[graham97] B. Graham. Harmonisation of Human Machine Interface, the intuitive approach. In *Proceedings of the ATM'97 seminar*, 1997.

[hutchins98] E. Hutchins. The Cognitive Consequences of Patterns of Information Flow. In *Proceedings of IHO-Aéro 1998*. p.3-6. Montréal.

[khaslavsky99] J. Khaslavsky, N. Shedroff. Understanding the Seductive Experience. In *Communications of the ACM*, May 1999, 42(5).

[leroux98] M. Leroux. An in-depth evaluation of Erato tools. In *Proceedings of the ATM'98 seminar*, 1998. http://atm-seminar-98.eurocontrol.fr/

[marais99] F. Marais, M. Torrent-Güell, E. Tremblay. Contraintes ERATO V1. Internal CENA report NT99775.

[maury99] S. Maury, S. Athènes, S. Chatty. Rhythmic menus: toward interaction based on rhythm. In *Proceedings of CHI'99, ext. abstracts*. 254-255.

[mertz97] C. Mertz. Faut-il tuer la souris? Une comparaison des charges cognitives induites par 6 procédures de saisie. In *Actes d'IHM'97*. Cépaduès-Editions. 1997. 79-87

[mertz99] C. Mertz, J.L. Vinot. Touch Input Screens and Animations: more Efficient and Humanized Computer Interactions for ATC(O). In *Proceedings 10th International Symposium On Aviation Psychology*. Columbus, Ohio. May 1999. 615-621.

[mertz00] C. Mertz, S. Chatty, J.L. Vinot. Pushing the limits of ATC user interface design beyond S&M interaction. *ATM2000 R&D seminar.* Napoli june 2000. http://atm-seminar-2000.eurocontrol.fr/

[meyer95] A. Meyer. Pen Computing. A technology overview and a vision. In SIGCHI Bulletin, 27(3), 1995, 46-90.

[mullet95] K. Mullet, D. Sano. *Designing visual interfaces*, Prentice Hall, 1995.

[shneiderman93] *Sparks of Innovation in Human-Computer Interaction*, B. Shneiderman editor, Ablex Publ, NJ (1993), 2nd edition.

[tapia] M. Tapia , G. Kurtenbach. Some design refinements and principles on the appearance and behavior of marking menus. *Proceedings of UIST'95*, Addison Wesley, 1995, 189-195.

[vanderdonckt99] J. Vanderdonckt. Development Milestones towards a Tool for Working with Guidelines, Interacting with Computers, 12(2), 1999, 81-118

[wellner93] P. Wellner, W. Mackay, R. Gold. Computer Augmented Environments. *Special Issue of Communications of the ACM*, 23(7), July 1999.