Preliminary Updates & Scientific Reports: *Multimedia Research Group. MMG – Intelligent User Interface*.
Lehrstuhl Algorithmen & Datenstrukturen. Institut für Informatik. Albert-Ludwigs-Universität Freiburg.

Adaptable, Adaptive and Perceptual Concepts for UI-On-Demand-and-Within-Reach Configurations

Khaireel A. Mohamed

27 December 2004

Preamble

The digital ink environment is unlike the desktop metaphor that we know and have grown used to. A digital board is more personal than a monitor, in a way that it allows us to directly touch it and see the immediate reaction through the digital pen. In classrooms where huge wall-mounted digital boards are utilised, it is impossible to have a bird's eye view of the whole environment when we are so near their proximity to impose the conventional desktop interfaces. We present an alternative solution that seeks to combine three classical UI-design concepts involving adaptable, adaptive and perceptual ideas for an on-demand-and-within-reach interface that monopolizes the single-stroke digital ink inputs. We show that by maintaining an active background tracking system on the temporal and spatial information of every input ink data, we can anticipate convincingly what users are trying to do to achieve their goals when working on the boards, and provide them with assistance within predictable contexts. Our methods maintain that users must always feel in control no matter how adaptive the assistance from the system may be.

1 Introduction

Over the years we have come to accept the fact that graphical user interfaces are not only visually helpful, but optimal in almost every way we can imagine for users of the desktop computers. The designs of which are tailored specifically for user-oriented goals such as completing that word document with style, enhancing the artwork for that poster, searching and getting information from the web, or even controlling that regional train tracks to follow schedule. Such diversity in the range of applications used for the desktops triggered sharp interests in both the IUI and HCI communities to continue enhancing and providing guidelines for future interfacing trends. Where it seems that the former community favours adaptivity concepts and the latter adaptability, we have also seen in the recent years the synthesis of the two [3].

Interfaces that are adaptable allow users to customise the application to suit their comfort needs and this is seen strongly in Shneiderman's stand for "direct manipulation" [16, 17]. By putting everything on screen in a certain orderly fashion and relying on the human hand-and-eye coordination, the interface design stresses on the convenience of users to achieve their goals through their own actions. The interface provides a top-down global approach environment for the users, and a lot of efforts are placed in this "art" of direct-manipulation on the designers to support complex information management tasks [9]. In addition, users can customise the default interface widgets for an overall screen outlook that they can be most

comfortable with. Adaptive interfaces on the other hand, model the individual user's interest, preferences and usage characteristics of the application to allow the interface to tailor itself to each other. The interface takes a proactive stance in assisting users and often involves background agents. It operates best when there is indeterminism about user actions and events within the environment and when there is no viable prospect to organise all data at hand. We agree with Maes [17] that agents are not an alternative for direct manipulation, but are the complimentary metaphors; very good application interfaces are needed for agents to be effective as they are not substitutes for the interfaces. But by combining adaptivity and adaptability, we do find added advantages for intermediate and advanced users as reported by Bunt *et al.* [3] in their user-analysis studies that involved the application of Fitt's and Hick's laws and the GOMS model. Particularly, their mention that "a reduced interface is bound to be more efficient" is what triggered our own research in developing an appropriate interface for the huge wall-mounted digital boards in classroom situations.

Our strongest constraint when working towards an appropriate interface for these huge boards is to maximise on a single input modal, namely the digital pen. Recently, the digital ink has received more attention as progress is made in optimising its data structure and storage methodologies [15]. A shift in the writing paradigm has led many to believe that we can leave the raw input ink as ink representing itself [1, 2]. As, in the first place, content readability would depend greatly on the user's ability to write clearly (and that very good handwriting recognition technology is vital to not misinterpret any freehand writings on the boards). What's more, because of this perceptual thinking, many have looked at the physical properties of the ink features and derive from them natural gesticulated commands [14]. Few works show the successful merging of the 'writing' mode and the 'gesture-command' mode on a single, non-segregated platform, without having escape sequences or using the button on the digital pen to switch between the two modes [7]. However, those works are not meant for the interface-interaction techniques with huge wall-mounted digital boards, and that the user interfacing issues of such have yet to be fully exploited. Here, it follows that Turk & Robertson's [18] perceptual user interface (PUI) approach seems to be the bridge we are looking for.

PUIs seek to combine an understanding of natural human capabilities with the actual device medium and the related machine perception and reasoning. The goal is to make the user interface more natural and compelling by taking the advantage of the ways people naturally interact with each other and with the world. In our case, we want the user to feel most comfortable using the electronic board as if it was a chalkboard. And then, without having to always search for a command from the pull-down menus (at some fixed screen location), indicate as naturally as possible by gesticulating anywhere on the board, of 'what we think the command might look like'. This is as opposed to forcing users to memorise 'what gestures are associated to which commands'. This manner of communicating with the hardware, according to the PUI standards, is expected to be transparent and passive, and the machine in turn should perceive this relevant human way of communicating, and react by generating graphical or visualization outputs that would be naturally understood by the user.

The abundant writing surface allowed by huge wall-mounted digital boards (see Figure 1) is harder to perceive for user-machine interactions as compared to the comfortably sized tablets and office whiteboards. Freehand writings are an entity on their own, containing the flair and style of expressions of the writer, which do not necessarily follow an ordered sequence or format when ample space is provided. To date we see a number of digital ink-based studies that concentrate on managing contents and editing 'informally' [1, 7], which came about as a result of this entity. Whether it is grouping a set of freehand words into a continuous flow of sentence, or combining a set of drawings to form the physics of represented objects, the programs rely on a lot of command buttons in toolboxes and pull-down menus to affect their causes. And since all of these applications are tailored for the desktop setup (including the tablets), the authors need not mention any proximity problems. Try putting these same appli-

cations on the huge wall-mounted digital boards and we can expect users to complain about its lack of suitable interface.

The point is, if we can get the system to anticipate correctly what its users are trying to do to achieve their goals, by providing assistance-on-demand and within-reach just by analysing the input ink elements, then we would be one step closer to realizing the most appropriate user interface that properly combines issues of adaptability, adaptivity, and those mentioned in the PUI standards, for our huge wall-mounted digital boards. As George Santayana (1863 - 1952), a Spanish-born philosopher once said, "Science is nothing but developed perception, interpreted intent, common sense rounded out and minutely articulated."

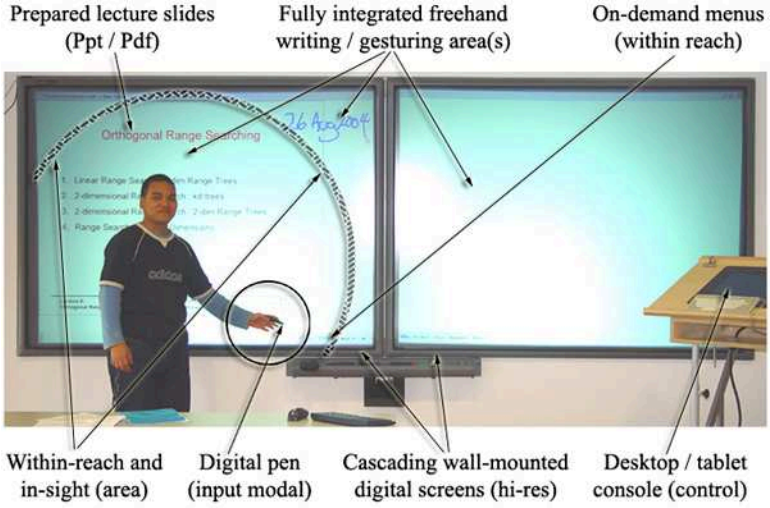

Figure 1 Wall-mounted digital boards and issues of user interfaces for the classroom.

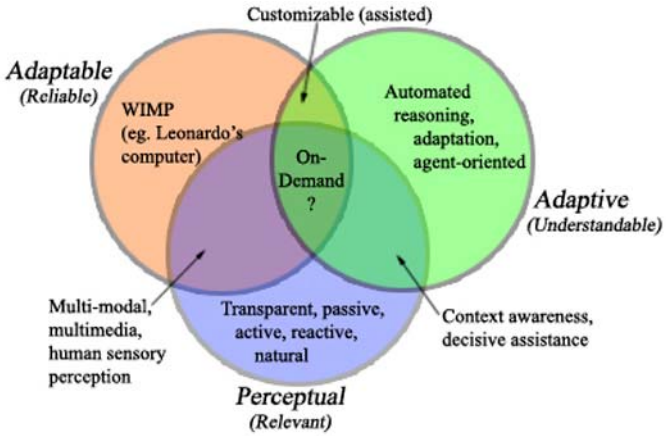

Figure 2 Relating adaptable, adaptive and perceptual concepts on user interfacing and its design.

2 UI-Design Concepts

Figure 2 shows the Venn diagram representing the three main UI-design concepts we highlighted earlier and their relations to each other.

- Adaptable interfaces can be considered *reliable* if it gives a faithful picture of what is happening within the application programs, if it is comprehensive (nothing material has been left out of the picture), and if it is verifiable (the interface carries out the commands as hinted by the content-widgets).
- Adaptive interfaces are *understandable* when they are not designed for the benefit of experts alone, and when their proactive assistances still leave the impression that the users are always in control.
- The *relevance* of perceptual interfaces has the potential to affect a user's belief: to confirm his or her beliefs from past actions, or to shape his or her beliefs for future interactivities with the system. We add here that amassing relevant information is usually timely, and sometimes the delay reduces utility.

When we overlap these distinctive characteristics onto each other, we obtain corollaries that are generally useful for domain specific applications. For instance, merging part of adaptable and adaptive concepts results in better (assisted) management of customisable menus that has been proven to expedite (manual) repetitive tasks. Secondly, negative aspects of adaptive user interfaces on "ease of use" can be overcome by adding a perceptual support component of user-mental-models based on contextual information. And finally, visualizing the various human sensory perceptions on the WIMPs as they display each multi-modal element perceived from the environment, reassures the user that the system has registered all inputs and is doing something to evaluate them.

Our approach to developing a suitable interface for our case lies in the center of the merges. For in there, we are able to address the problems of providing for huge writing surfaces without compromising on critical issues known to each UI-design concept.

3 The Board Environment

Chalkboards are by far the most widely used medium for lecture presentations and they are very effective. Not only do they allow instructors to communicate all or part of the course material in natural handwriting, but they also take away the attention to details for their use. But as our knowledge providers turn to technology, Chong & Sakauchi [5] observed that when presentation slides are available (either as overhead projected slides, or PowerPoint slides from a beamer), the use of the chalkboard is significantly reduced during the course of the lecture. However, when it comes to facilitating questions and answers, the chalkboard is conveniently used again to demonstrate facts, confirm equations, and diagrammatically illustrate flowcharts.

Still, making a direct reference from the chalkboard to one of the slides can be a hassle, and sometimes the instructors resort to annotating on those slides in whatever little space is available, and may even compromise on legibility. Removing this awkwardness is similar to what Golovchinsky and Denoue [6] described as information triage, where they found people printing out heaps of documents to annotate on them only to return to the computer screen to fine tune their search, based on what was read on the printouts.

3.1 Digital Tablets

The Wacom tablets and the Tablet PCs place the desktop computer screens literally on the desktops and allowed for one extra mode of input; the digital pen. This hardware innovation solves the problem of referencing between separated slides and physical writing areas by placing both (windows) adjacent to each other. Annotating on the slides or documents is

made more accessible as described clearly by Bargeron and Moscovich [2]. Further more, by using appropriate software, all presentation data can be saved and retrieved for viewing at a later time.

3.2 Digital Boards

The last couple of years have seen a growing number of wall-mounted digital boards utilised in the education and corporate sectors. Almost of them when connected to an operating system impose the 'desktop' interface to their users for all board-application programs. Others have no clear-cut user interfacing mechanisms and rely strictly on the actual hardware devices to simulate a typical whiteboard. We can parallel arguments of adaptability and adaptivity of the interfaces to the current literature because all these wall-mounted boards are being handled from another console; usually a desktop setup. When this happens, we feel that the wall-mounted boards are just the more expensive versions of the beamer-projectors, and it seems wasteful that we do not fully utilise the functions available on the digital screens. As a matter of fact, the screens themselves are able to receive direct inputs from users albeit their awkwardness in size. We attributed this earlier to the lack of proper interfaces for those screens, or if a good one exists, then it is not completely dependable on just the digital pen input, which may be most convenient for users of this device.

Unlike desktop computers, wall-mounted digital boards are meant to be an active presentation medium for a congregation of audience in a large room. While there is always a desktop console that can control the wall-mounted boards, through the mouse and keyboard, we want to free up this mode of interfacing and let the user (who is usually the instructor) communicate directly with the screen using just a digital pen. Judging by the dominant metaphor that what we have today is a mismatch for the computer environment we are dealing with tomorrow, it is difficult to place this huge screen scenario (alone) into any pronounced categories of user interface debates on concepts of adaptivity and adaptability. For instance, the sphere of influence of the hand-and-eye coordination needs to be enlarged, and perhaps include the body, to follow the claims of direct manipulation of adaptable interfaces. Furthermore, while the audience have it easy watching from a distance, the instructor does not: standing so close to one part of the huge screens often leads to the interface widgets he or she may require to be out of reach, or worse still, because the instructor cannot see those widgets, he or she may assume that such actions or commands represented by those widgets do not exist within the board application. We point out here that this may affect the flow of the lecture presentation. In this case, we may be left to rely on the adaptivity of the interface for the wall-mounted boards to proactively assist the user while at the same time ensuring that the user still feels in control of the whole process.

3.3 Control Widgets

Fewer widgets on screen may turn out to be more effective as hypothesized by Bunt *et al.* [3] especially for these types of situations. For if we want to reproduce the physical chalkboard on the digital boards, and still be able to take full advantage of all the capabilities that the digital system can offer, we may need to compromise on the number of interfacing widgets so that the instructor can focus on his or her own lesson delivery and pay less attention to the background technology. In fact, one of the goals of creating a practical interface for the boards is to give as much writing space as possible to the users, and to adopt 'natural' hand gestures whenever appropriate to invoke only the necessary commands and tools given the context, while engaging in a lecture presentation.

4 Contextualizing Ink in Freehand Writings

The digital ink domain is unlike any others that can have all its related information easily organised and cross-referenced, and presented in front of users to allow them direct manipulation of the data. We begin by defining a trace that refers to a trail of digital ink data made between a successive pair of pen-down and pen-up events representing a sequence of contiguous ink points - the *X* and *Y* coordinates of the pen's position. Sometimes, we may find it advantageous to also include timestamps for each pair of the sampled coordinates, if the sampling property of the transducer device is not constant. A sequence of traces accumulates to meaningful graphics, forming what we (humans) perceive as characters, words, drawings, or commands. Each trace can be categorised in terms of the timings noted for its duration, lead and lag times as illustrated in Figure 3.

4.1 Temporal Relations

For a set of contiguous ink components $S_j = \{c_0, \ldots, c_n\}$ in a freehand sentence made up of n traces, we note that the lag-time for the i^{th} component is exactly the same as the lead-time of the $(i+1)^{th}$ component; i.e. $\texttt{lag}(c_i) = \texttt{lead}(c_{i+1})$. Consequently, the timings that separate one set of ink components apart from another are the first lead-time $\texttt{lead}(c_0)$ and the last lag-time $\texttt{lag}(c_n)$ in S_j. These times are significantly longer than their in-between neighbours c_1 to c_{n-1}.

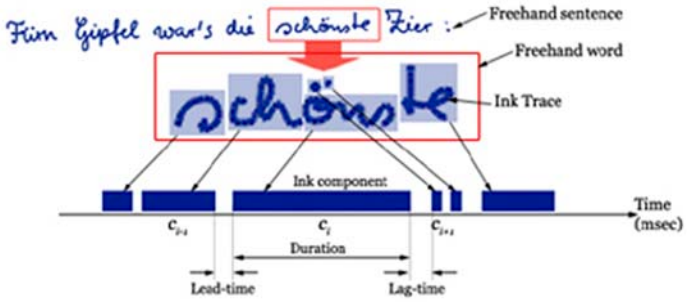

Figure 3 Splitting up freehand writing into ink components on the time line.

Furthermore, if we observe a *complete* freehand sentence made up of a group of freehand words, we can categorise each ink component within those words into one of the following four groups:

- *Beginnings* – Ink components at the *start* of a freehand word;
- *Endings* – Ink components at the *end* of a freehand word;
- *In-betweens* – Ink components in the *middle* of a freehand word; and
- *Stand-alones* – Disjointed ink components.

The groups differ in the demarcations of their lead and lag times, and as such, provide for a way a perceptual system can identify them. The temporal relation that singles out *stand-alone* components (and freehand gestures) is the "longer-than-average" lead and lag times of a single-stroke ink trace, as shown in Figure 4. There is a pause period between samplings of ink components that results in significantly longer lead *and* lag times. It is not very often that we see people gesturing to a control system in the middle of writing a sentence or drawing a diagram. We reported in our previous work [11], that we can convincingly anticipate, based

on the lead and lag times obtained, that the latest ink component is more of an instance of a trace, rather than a gesture.

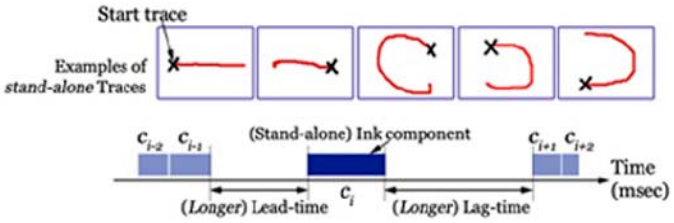

Figure 4 Stand-alone ink components that can be interpreted.

4.2 Recognition

Advanced recognition techniques have been introduced to the digital ink domain, and while a number of works have gone into handwriting and sketch recognition [14], there are among these works that use the same techniques to attempt to recognise hand-drawn pen gestures [7]. Hence we now have the 'writing' mode as well as the 'gesture' mode to help out with the current interface issues. We point out here that by 'gesturing to the system' we mean the actual gesticulated actions resulting from the pen movements that represent a command, and not merely the 'tap and drag the icon' movements. The prospects of gesturing commands in ink-domain applications are encouraging. So much so that many authors think that this is the epitome of the final interface for pen-based applications. Although we agree to a certain extent, that this mode of ubiquitous interfacing may form the fundamentals of future (more enhanced) hardware devices, we do not anticipate this to be the trend in our present context. The PUI guidelines should be adopted and used to create the first steps of the bridging between fully packed screens of user interface widgets to that of a ubiquitous one.

4.3 Ink States

Based on the four categories of the temporal observations guarding each ink entity, our state transition diagram in Figure 5 illustrates the relations (in states) that an ink component can take. Our previous works show that there exist anticipatory action-events between a trace and a gesture, and that those events are primarily associated with the *ending* and *stand-alone* transitions [11]. A trace has a high probability of becoming a gesture when it is categorised as an *ending*, and is almost always a gesture when it is a *stand-alone*. It remains a trace when it is either a *beginning* or an *in-between*. Similarly, a gesture can be degraded back to a trace should the user decide it to be so even if the system has anticipated a gesture judging from the ink component being categorised as an *ending* or a *stand-alone*.

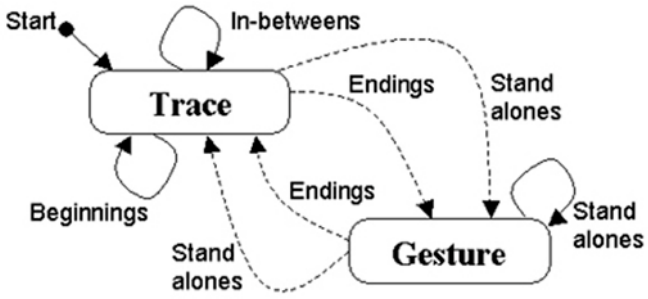

Figure 5 State transitions between an ink trace and a gesture.

The two main states, trace and gesture, and the current context of the writings on the digital boards can be immediately scrutinised. We are able to know

- when and where the user is engaged in continuous writings;
- when and where the user is sending commands to the system; and
- when the user is not interacting with the boards at all.

We formulated an active background environment and found five distinct states that we utilised in our quest to derive an interfacing mechanism that fits the overlaps of the three user interface concepts highlighted in Figure 2 for the huge wall-mounted digital boards.

5 Active Background Tracking

Part of our solution of minimizing the numerous interface combinations of toolboxes and command buttons is to spruce up the background environment with active and reactive mechanisms – placing interface agents as collateral to borrowing processing time from the system. Since Jameson [8] claimed that there are no conclusive proofs that adaptable interfaces are always superior, our system's active background components are also designed to tolerate adaptive user characteristics and tasks, as well as perceptual events received from the ink environment of the huge wall-mounted digital boards, solely from the pen-based input modal [10].

5.1 The Background Model

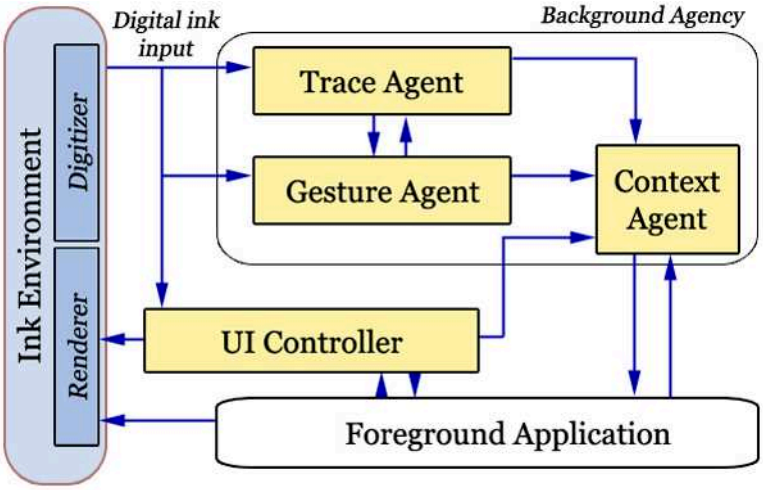

Figure 6 Model of the active background involving the ink environment, an agency, and a foreground application.

There are five components that make up the active background model to track ink inputs perceived from the board environment as shown in Figure 6. The Trace Agent, Gesture Agent and the UI Controller all receive the same information from the digitiser. The Trace Agent classifies each ink component into one of the four temporal categories and determines statistically the state of the ink. The Gesture Agent uses the Trace Agent's verdict and its own percept of the ink environment to determine the kind of gesture (command) to tell the foreground application, should the ink component not be a trace. The UI Controller provides the appropriate graphical visualization of the user input, drawing its instructions from the

foreground application by leaving trace marks on the screen or showing on-demand menu options near the user's current position with respect to the boards.

A Context Agent monitors feedback interactions between the user and the system. This agent communicates internally with both the Trace and Gesture agents to determine all spatial and temporal ink activities on the ink environment. It also requests further information from the UI Controller and the Foreground Application to confirm the ink states and to double-check its own data collection of previous events to better anticipate current and future ink activities [12]. Among the agent's more notable properties are:

- Determining the consistency of the freehand writings;
- Determining the frequency of gestures with respect to the writings;
- (Assisted) Acting and reacting to the foreground applications; and
- Deciding the most appropriate context given any (writing) situation.

In terms of coming up with the appropriate context for a particular (digital) board application, the agent works on the initial information of the current ink state. Here, traces are straight-forwardly associated with writings (or drawings) and then depending on the user's editing choices, the agent sets the 'context' to allow specifically for changing pen attributes, canvas attributes, or any other annotating activities. Single-stroke gestures are tagged to the more specific menu items for browsing the boards (and/or slides) contents, accessing files, invoking help, as well as performing editing commands such as copy, cut, paste, insert, delete, etc. On the forefront, this resembles very much like the popup menu ideology, but with the exception of dynamic menu contents reacting on the same set of gestures and traces. In our case, the contents are dependent on the current user situation.

5.2 Strategy

Robertson [18] proposed a strategy for designing an effective interface that we find satisfies issues of adaptability, adaptivity, and *perceptuality*. He concentrated on three key components:

- Leveraging human capabilities;
- Exploiting the available technologies; and then
- Accessing and manipulating all information as we see fit.

This is in relation to the Seeheim model pointed out by Morse [13], which highlights the "functional core" that directs the "presentation component" and the "dialogue controller".

The state transition diagram in Figure 7 encompasses Robertson and Morse's concepts, as well as our active background-tracking model that optimally manages a way of communicating with the user in a purely digital ink environment. Five transitable states are based upon the input ink states received (either as traces or gestures), or the lack of it (either as tapping on the board, or no-action, or prolonged no-action) with the help of a pervasive timer mechanism for this reactionary stratagem.

As a rule of thumb, we leaned towards the play-safe strategy; for whenever there is a "doubt" (a level of un-assuredness beyond a certain threshold value [11]), the user must be given a choice to see and select from the full-range of commands offered by the foreground application. This is prominently reflected in State 1. States 2 and 3 are on-demand menus and interface options proffered by the system based on the current written ink context. Only the selected (anticipatory) widgets are displayed on screen while all other options that are deemed "not-in-context" are packed under a command button marked "`More >>`". Referring to the fact that we want to provide the instructors with an uncluttered writing canvas, the interface

widgets will disappear once the system detects continuous writings. This strategy works on all our board scenarios including those that we use for purely "writing"-applications, as well as those that integrate lecture slides in them (see Figures 1 and 10 for examples). The emphasis of which is directed towards a single digital pen input by the user to put him or her in control of the class and the technology within.

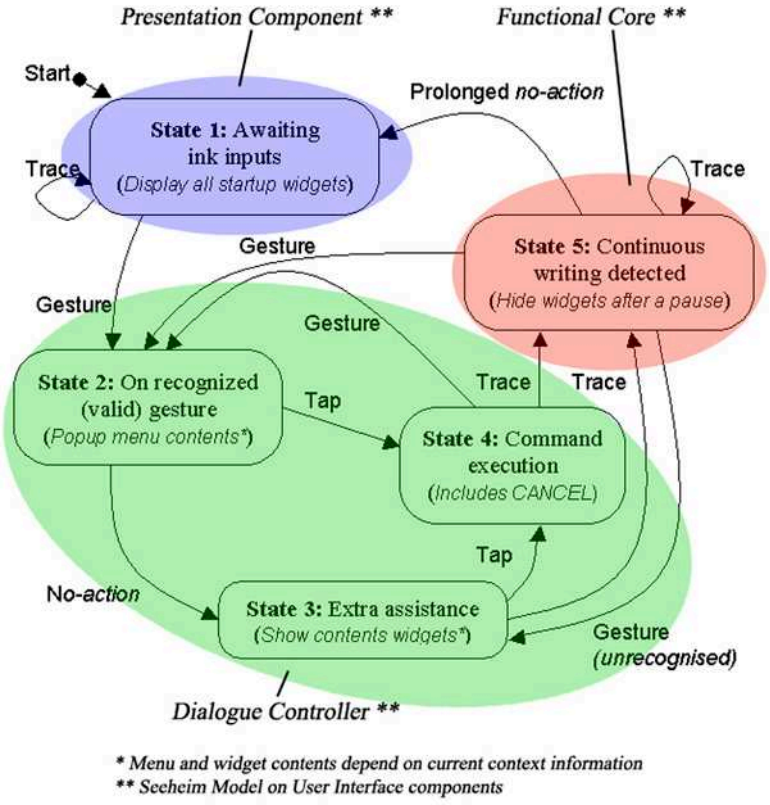

Figure 7 Tracking ink events and reactionary stratagem.

5.3 Deriving Interfaces from Ink Inputs

To complete the visualization of the on-demand assistance, we bank-in on the last known coordinate position of the user's latest written trace (or gesture). Popping up dynamic-content pie menus and/or other static command buttons in that vicinity seems most appropriate and serves a convenience for the user. Earlier research on pie menus proved an advantage for users of digital screens [4], as people tend to better remember the cyclic position, and as such expedites the selection process. Our command button (toolbar) interface is logged to the bottom of the screens, appearing on-demand with only an array of anticipated widgets (see Figures 1 and 10 for examples). The combined lengths of the number of widgets will not exceed the arm's length of the instructor to ensure that all widgets are within reach and within the instructor's line of sight.

6 Findings and Evaluation

To measure the response of users who utilize the huge wall-mounted digital boards, based solely on the digital pen inputs, we collected and analysed all ink information received in

classroom lessons for a period of one semester week. The digital boards were set up to record the amount of writings by instructors, and to see how much of those writings are 'meant by' and 'interpreted' as command gestures or for invoking on-demand menu options. The data are then used to infer how well our support system reacts in providing valid and useful assistance to instructors. While the system records all the above performance data, we also stationed a human observer for each lesson conducted that utilized the digital boards. The sole purpose is to track "human errors", where we define this to be the changing of the instructor's mind on a 'set action'. For example, after purposely calling up an on-demand menu, the instructor decides that the menu is not needed after all (subsequently, the Cancel option was selected), or the instructor decides that he or she really wanted to access a different option that is completely out of context (subsequently, the "`More >>`" button was selected).

6.1 Ink, Commands and the UIs

There was a total of 125,866 ink inputs received and processed by the digital boards during our observation period. All lessons had PowerPoint presentation slides embedded on the screen. Instructors were able to browse the contents of the slides and the boards, and write anywhere on the screen they please. We noted that over 94% of the digital inks collected were freehand writings (and drawings), and only 5.15% were anticipated by the system as gestures for commands and for invoking the hidden (on-demand) interface. Figure 8 depicts the further breakdown of the command gestures and the extent the correct anticipation led to the invoking of the appropriate UIs. "Human errors" amounted to 12.43% of the system's anticipation process. This is as opposed to the system's own error in judgment where it either reacted with a wrong command or interpreted the wrong context, which were all due to the misinterpretation of ambiguous single-stroke gestures. This is a prevalent problem that we highlighted in our previous works [12] and is also explained by various authors [7, 14], especially in cases where the legibility of the instructor's freehand writings are by themselves questionable.

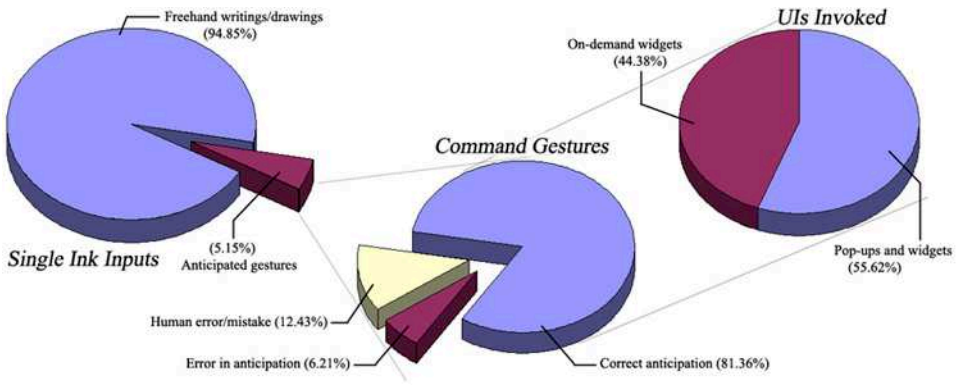

Figure 8 Deriving interfaces from single-stroke digital ink inputs.

Our data shows that of all the correct (assisted) interpretation made by the system in providing the interface-on-demand, more that 55% of the time involved popup pie menus. The rest of the time required the appearances of toolbar widgets near the instructors. Our observers noted that the instructors took the advantage of the absence of tool-widgets and naturally utilized the boards as if they were chalkboards – writing everywhere and anywhere they please. The instructors did not seem to mind the lack of visible technology, as to them the system seems to conjure up the interface whenever it is that they need to use it. Making choices from the anticipated menu options gave users the impression that the boards have always been in their control.

6.2 Users' Choices

To obtain a better picture of how users take to having their menu options anticipated by the system in whatever context they are currently working on and then presented to them on-demand, we also made provisions to monitor the kinds of widget options users click. The graph in Figure 9 depicts this finding.

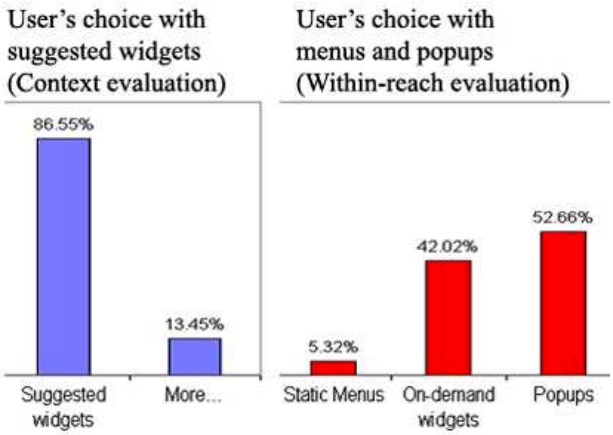

Figure 9 Users choices on widget options.

The first two bars highlight the frequency of users' choices whenever any options were selected from the bottom toolbar. Our system was able to correctly anticipate the context in which the users are working more than 86% of the time. That is, within the limited number of command buttons allowed for the arm span of the users, the system is able to place at least one button that the user can tap for immediate action. Only 13.45% of the time do users have to locate their commands buried inside the "`More >>`" button. The last three bars show the number of times users click on either the static pull-down menus, the on-demand widgets (bottom toolbars) and the dynamic-content popup-pie menus respectively. Over half the time we see users utilizing the convenience of the popups and 42.02% of the time on the on-demand widgets.

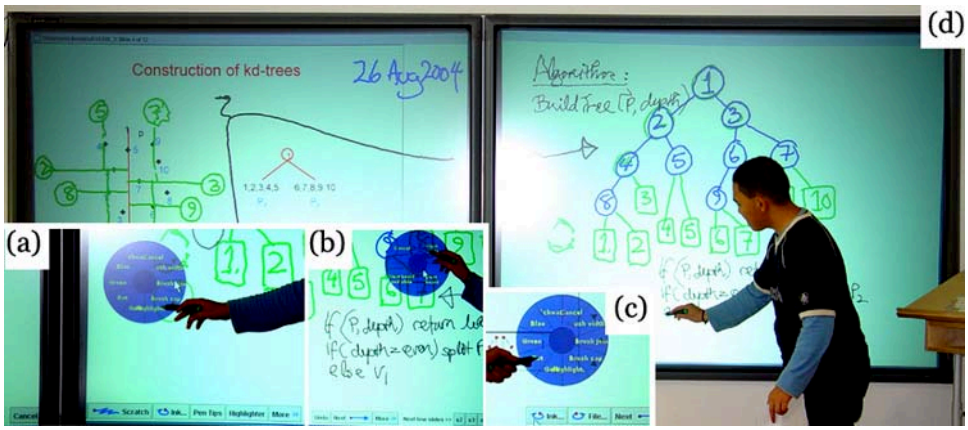

Figure 10 Huge wall-mounted digital boards as teaching medium. Insets: (a) dynamic-content pie menu for pen attributes, (b) dynamic-content pie menu for browsing slides/board-canvas, (c) bottom toolbar widgets within reach and in-sight of user, (d) normal freehand writings left as trace marks on the ink environment.

7 Conclusion

Our derivation of a system to support three UI concepts of adaptability, adaptivity, and *perceptuality,* for the benefit of huge wall-mounted digital boards based on just the single input modal of the digital pen, is reliable, understandable and relevant. Indeed in our case, fewer widgets on screen seem to be better in classroom situations. But without concentrating too much on the technology, we got instructors to still feel in control of their teaching devices and putting in more of their concentration in presenting their lessons. We find that having UIs on-demand-and-within-reach for huge wall-mounted digital boards appropriate for classroom situations.

Acknowledgement

This research is funded by the Deutschen Forschungsgemeinschaft (DFG) as part of the research project „Algorithmen und Datenstrukturen für ausgewählte diskrete Probleme (DFG-Projekt Ot64/8-3)“.

References

[1] R. J. Anderson, C. Hoyer, S. A. Wolfman, and R. Anderson. A study of digital ink in lecture presentation. In *Proceedings of the 2004 Conference on Human Factors in Computing Systems*, pp 567-574, 2004.

[2] D. Bargeron, and T. Moscovich. Reflowing digital ink annotations. In *Proceedings of the Conference on Human Factors in Computing Systems*, pp 385-393, 2003.

[3] A. Bunt, C. Conati, and J. McGrenere. What role can adaptive support play in an adaptable system? In *Proceedings of the 9th International Conference on IUI*, pp 117-124, 2004.

[4] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 95-100, 1988.

[5] N. S. T. Chong, and M. Sakauchi. Back to the basics: a first class chalkboard and more. In *Proceedings of the 2000 ACM Symposium on Applied Computing*, pp 131-136, 2000.

[6] G. Golovchinsky, and L. Denoue. Moving markup: repositioning freeform annotations. In *Proceedings of the 15th Annual ACM Symposium on UIST*, pp 21-30, 2002.

[7] T. Igarashi, W. K. Edwards, A. LaMarca, and E. D. Mynatt. An architecture for pen-based interaction on electronic whiteboards. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp 68-75, 2000.

[8] A. Jameson. Adaptive interfaces and agents. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pp 305-330, 2003.

[9] J. McGrenere, R. M. Baecker, and K. S. Booth. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 164-170, 2002.

[10] K. A. Mohamed, and Th. Ottmann. Pen-based digital screen interaction. In Claude Ghaoui, editor, *Encyclopedia of Human-Computer Interaction*, to appear.

[11] K. A. Mohamed. Increasing the accuracy of anticipation with lead-lag timing analysis of digital freehand writings for the perceptual environment. In *Proceedings of the ICCMSE 2004*, to appear.

[12] K. A. Mohamed, and Th. Ottmann. Fast interpretation of pen gestures with competent agents. In *Proceedings of the IEEE Second International Conference on CIRAS*, pp PS09-4, 2003.

[13] A. Morse, and G. Reynolds. Overcoming current growth limits in UI development. In *Communications of the ACM*, 36(4):72-81, 1993.

[14] D. Rubine. Specifying gestures by example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, pp 329-337, 1991.

[15] M. Turk, and G. Robertson. Perceptual user interfaces (introduction). In *Communications of the ACM*, 43(3):32-34, 2000.

[16] G. Russell, Y.-M. Chee, G. Seni, L. Yaeger, C. Tremblay, K. Franke, S. Madhvanath, and M. Froumentin. Ink markup language [online], Available: http://www.w3.org/tr/inkml. *W3C Working Draft*, 2003.

[17] B. Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of the 2nd International Conference on IUI*, pp 33-39, 1997.

[18] B. Shneiderman, and P. Maes. Direct manipulation vs. interface agents. *Interactions*, 4(6):42-61, 1997.