Licentiate theses proposal: Towards new Interaction - A Content Centric Data Surface Approach

Rikard Lindell

September 14, 2004

Abstract The staring point for the work presented in this theses came from the insight that users are explicitly dealing with a lot of actions that the computer system is better suited for, and that these actions can be derived from the desktop metaphor interface paradigm and its infrastructure. Why double click on a square icon for launching an application? And, why do you explicitly have to save content that is already in the computer? By removing the components of the desktop paradigm that made is so well designed for the mouse, I have found a possible path towards new interaction. The path was built by constructing prototypes that embodies my ideas and that could be used by subject users. By leverage on cognitive science, observations, interviews, and usability evaluations I have been able to find strong indications that the approach presented in the thesis support users' expected services, users' creativity in action, and users' awareness in collaboration, in a manner that the users find fresh, fun, and pleasing.

1 Introduction

Today's design of the desktop paradigm interfaces found on various platforms is very similar to the original design of the Star system at Xerox PARC in the late 70's. Before this the interface was a blinking cursor by which highly trained expert users wrote more or less confusing command acronyms. Although the desktop interface have been very successful, it was not long ago that people thought of the graphical user interface and the mouse as a toy. The original Star system design was for office applications and desktop publishing. Taken into consideration this context and the fact that the physical size of the Star, its peripheral devices, and its location on an office desktop, together with the weak computational resources memory and storage space: the desktop paradigm metaphor was a wise design decision. Since then, the flow and repositories of information that users have to handle has increased by an order of magnitude. Still, thanks to its success for the office context, the desktop paradigm has been stretched over a numerous application areas, and was maliciously squeezed into

mobile devices. Today PC and its graphical desktop paradigm interface is a commodity.

2 Research Focus

I have deduced the complexity in the interaction design to the underlying infrastructure of file systems, application programs, and the use of metaphors to explain how computer works. The desktop metaphor interface paradigm was not design for the contemporary large flow and repositories of information, for user collaboration, and for multi modal interfaces. The theses present a new interaction paradigm that takes these issues in consideration. The work towards new interaction starts with these questions:

- Can reassessment instead of rejection of the fundamental design values of the desktop metaphor interface paradigm render an interface paradigm that better supports human cognition, collaborative work, users creativity in action, and multi modal interfaces?
- Can this new interface paradigm be more consistent with reassessed design values of the desktop metaphor than the current desktop metaphor interface design?
- What happens if fundamental constructions of the desktop interface, windows, icons, menus, files and application programs, are removed?
- Would users enjoy and will they be satisfied with an interface paradigm that have no windows, no menus, no icons, no files, and no applications?

3 Contribution

There are two main contributions from the theses. One scientific and another more practical. From the scientific aspect I have shown that interaction for personal computers can be based on reassessed design values of the desktop metaphor interface as intellectual foundation and can be significantly different from the current reign of the desktop metaphor, appreciated by users, and more compliant to its design values. The practical contribution is guidelines for interaction designers and software developers who want to design and implement systems based on the data surface paradigm.

3.1 Publications

Users Say: We Do Not Like to Talk to Each Other, in proceedings of Graphical Communication Workshop 2003, pages 87-90, publisher Queen Mary University of London.

When Information Navigation Divorces File Systems - Database Surface Prototype Results, in proceedings of The Good, the Bad, and the Irrelevant: The user and future of information and communication technologies. Conference 2003, pages 76-81, publicher COST Action 269 and the Media Lab of University of Art and Design Helsinki UIAH

4 Methodology

The work presented in the thesis is based on results of research actives conducted in the field of interaction design. The design of the new interaction paradigm was revised and confirmed by building prototype systems based on the ideas for the new interaction paradigm and by letting users to interact with these prototypes in usability evaluation tests. The examined application area was music creativity. Users' expectations of music creativity tools were elicited by in depth interviews. The number of interviewees were six (6), and they were both novice users and expert users of computer music tools, and they were both male and female. The results were reported here [9].

I have also observed users while they have engaged in collaborative interaction with current tools, these results were reported here [9]. This activities and conceptual studies of related work (see section "Sources of Inspiration") were used for designing scenarios. These scenarios encapsulate the ideas for the underlying interaction paradigm, the users' expectations of music creativity tools, and personas of the users. From these scenarios concept prototypes for empirical evaluations were constructed. Three prototype iterations were completed.

One: An early concept music tool prototype. A concept prototype implemented using a multimedia tool for evaluation users attitudes towards zoom interface, users attitudes towards index map navigation and zoom history navigation aid, and users attitudes towards flat infinite surface of information. This prototype was evaluated by five users, using field studies and the collaborative evaluation method, the result were reported here [10].

Two: Fluent zoom approach prototypes. Results from the first prototype showed that zoom had to be smooth and fluent to be appreciated by users. Therefor a couple of prototypes for evaluating zoom navigation techniques were designed and implemented. These were evaluated by user observation of four users in a controlled environment. Evaluation data was collected using video and by debriefing interviews. Users preferred to have separated actions for zoom and navigation, the testes and results can be read about in [10].

Three: Music tool prototype for usability evaluation. The revised design from the previous investigations were used to implement and evaluate a third prototype. It was a real functioning collaborative tool for music creativity, functionally equivalent to the features used in the evaluation of the music

tool reported in [9]. The prototype implemented command invocation by incremental content context driven command completion, pre-visualised feedback, and non-visual typed commands. The prototype implemented mutual awareness of action and mutual modifiability by synchronisation of commands, visual content, and acoustic content as well as a chat component. The prototype was evaluated by 10 users, 2 collaboration pairs and 6 solo tests, using field studies in the users normal context (music studio and school environment). The users were observed utilising notes, log, and video for protocol. They were debriefed with interviews and focus groups.

5 Background: Values of the Desktop

Central to the desktop interface are the overlapping windows, files and folders, menus, trash can, and of course the desktop itself. But what is the intellectual foundation for this design? The Macintosh Human Interface Guidelines (referred to as the guidelines throughout this section) [3] neatly lists values that constitutes the Mac interface. Other desktop systems are very similar to the Mac and many of the great minds behind the desktop paradigm user interface at Xerox PARC were recruited to Apple including Alan Kay. Hence, the Mac interface stands here as a representative for desktop interface systems in general. The guidelines state 11 values as the founding principal for the desktop interface which are: Metaphors, Direct Manipulation, See-and-Point, Consistency, WYSIWYG, User Control, Feedback and Dialog, Forgiveness, Perceived Stability, Aesthetic Integrity and Modelessness. Gentner and Nielsen wrote in 1996 the article "The Anti-Mac Interface" [4], in which they go through all these values and rejected them in the view of what future interfaces should be. I have found that many of these values are above the Mac, and that a different interaction paradigm can be designed and implemented reflecting on these values. I will run through these values and give a brief review of why Gentner and Nielsen rejected them.

Metaphors The guidelines states that by using metaphors to convey concepts and features of your application metaphors, you can take advantage of peoples knowledge of the world. They further state that metaphors should comist of concrete, familiar ideas. The guidelines explaine: "For example, people often use file folders to store paper documents in their offices. Therefore, it makes sense to people to store computer documents in computer-generated folders that look like file folders."

As Gentner and Nielsen pointed out out the target domain (the computer interface) does not have the same attributes, properties and features as in the source domain (in this case the desktop). Metaphors are only useful in explaining how a system works for novice users, whereas, over time, the metaphors vanish and the interface stand by itself. Introducing limiting constraints from the real world source domain into the virtual world target domain restrains the power of the interface, and thus the power of the computer.

Direct Manipulation The guidelines strongly advocates Shneiderman's [18] direct manipulation. The idea is that objects should be visible on the screen while the user perform physical actions on these objects. The impact of those operations on the object should also be immediately visible. The most widely known operation in this manner is the drag 'n' drop action.

Gentner and Nielsen pointed out that direct manipulation tend to work on an atomic level. For instance try to remove all the PDF-files of a folder and its subfolders. You have to recognise each file and drag it to the trash can. Whereas command line interfaces can be much more powerful for this action the single line: rm *.pdf -R irreparably removes all the PDFs: indeed powerful but also unsafe.

See-and-Point The guidelines tell us that: "users perform actions by choosing from alternatives presented on the screen". The screen and the visual appearance of objects and the visual appearance of actions available for the object constitute the main interaction style for users of desktop interface computer systems. The mouse is the main tool for interaction. The guidelines instruct interface design to utilise noun-then-verb interaction grammar.

Gentner and Nilesen's objection to the See-and-Point value was that it throws away the power of language. Extreme amount of information is condensed into human "natural" language. The computer power should be, according to Gentner and Nielsen, unleashed by reinforcing a language interaction style.

Consistency Consistency means that interface element behaviour should be consistent within different applications, making users' knowledge transferable from application to application. As an example, font selection should be similar and evoked by the same command in a simple text editing tool, an e-mail client, and a layout application package. If this holds, the users' interaction can be habitual and automated making the interaction smooth, efficient and safe.

Gentner and Nielsen attacked the Consistency value by interpreting consistency very literate. They argue that the guidelines only nurse the hope that users would more easily learn and use similar tools if they look and behave the same. They also claimed that the real world does not work this way, and exemplified this by the pointing out that people have no difficulty switching between ball-point pens and fibertip pens.

WYSIWYG (What You See Is What You Get) According to the guidelines WYSIWYG is about displaying the document on the screens in the same way as it will look on paper. Contesting this value statement is not in the scope of this thesis.

User Control The computer should not "take care" of the users by protecting from detailed judgements and decisions. User control means that the user, not

the computer, initiates all actions. This may be true for office applications however the process the user interacts with may very well be dynamic.

Gentner's and Nielsen's response to this was that the user have to be in control in a desktop interface environment. That the users have to monitor boring tasks or perform them themselves. They also point out that agents and daemons are incompatible with the User Control statement.

Feedback and Dialog The guidelines feedback and dialog value advise interaction design to always inform users about the application status. Feedback should be as immediate as possible, lengthy actions should inform of progress and when they are ready.

Gentner and Nielsen said that detailed feedback is needed for detailed control, and that a background agent only needs to provide sparse feedback.

Forgiveness The forgiveness value means that actions should be reversible. The guidelines also instructs you to use alert dialogue boxes for possibly dangerous irreversible actions, such as emptying the trash can. Forgiveness is supposed to make people feel more comfortable and safe while using the computer.

Gentner and Neilsen's view of forgiveness was that the desktop interface computer does not keep track of the user's actions. Thus information that could have been anticipated by bookkeeping of the user's intentions is thrown away which may lead to repeated annoying alert dialogue boxes.

Perceived Stability According to the guidelines perceived stability makes the computer environment predictable. Users that leave a document file on the desktop expect it be there when they come back. The interface is filled with consistent graphic elements (window frames, menu bar, buttons, etc.) that maintain illusion of stability.

Gentner and Nielsen said to this that you only have to look at the web, if no content were changed by others the world wide web would be boring. Other situation where stability is undesirable is in computer games and learning applications.

Aesthetic Integrity This value means that the graphical element layout should be consistent with visual information design principals. The guidelines encourage keep a minimum of interaction elements with a consistent elegant look to enhance usability. The screen layout should, according to the guidelines, not be cluttered and overloaded.

Gentner's and Nielsen's objection was that for increasing magnitude of information repositories coherent visual appearance of information elements make the navigation experience unpleasant and confusing.

Modelessness The guidelines argues for the interface design to be modeless. This means that all possible operation should be available at all times and

unrestricted by software modes. Users should be free to perform any operation at any time.

Gentner and Nielsen argued that one should yield to that it is impossible to create a modeless interface, they pointed out that even the guidelines devote the bigger part of the modelessness section to guide the correct use of modes. However, many of the most devastating user errors come from the user not recognising the mode of the system: these errors are known as mode-errors [14]. Desktop paradigm brings about a number of different modes, for instance dialogue boxes that prevent users from further actions in an application until they have completed the dialogue, the overlapping windows, applications, open and closed files.

5.1 Discussion

I have listed above the fundamental design values of the desktop metaphor interface. Gentner and Nielsen have pointed out weaknesses of the desktop paradigm and its design values. Their conclusion was that new interaction can only be achieved by rejecting these values. In my work I have instead reassessed these fundamental design values. Take for instance the modelessness value; observed novice computer users have trouble develop cognitive concepts for application programs. Simple tasks such as saving a document file, which severely break the modelessness value, present difficulties for users; they believe the content is already in the computer. If, in these cases, the design had complied with the modelessness value the complexity of developing cognitive concepts would have been reduced.

6 Sources of Inspiration - Related Works

Here is my sources of inspiration and related works from the research community and industry within the field of interaction.

Persistence Other models than traditional file-systems for persistent storage was utilised in the Data Soup object storage framework of the Apple Newton OS [19]. Apple's perspective was that developers had to write extensive object serialising code with traditional file systems and that an object oriented persistent system along with the embedded NewtonScript programming language would relieve developers from writing this code.

Zoom The idea of using semantic zoom for interaction was put forward by Perlin and Fox [7]. They proposed to siggraph 93 the Pad as an alternative to the desktop paradigm. In the pad information is put on a big white board, hierarchy is created by different scales where small scale object show only a caption of themselves hiding the rest of the document.

Bederson and Hollan, whom are the most cited for zoomable user interfaces, developed the Pad ideas into Pad++ [2]. Pad++ is a layer based on Tcl/Tk that allow semantic zoom into portal subspaces of an elastic surface of information.

The inspiration form Pad and Pad++ was the idea of using zoom as navigation technique. However, they are not compound content centric.

Focus+Context Focus+Context visualisation or was first presented as Flip-Zoom for reading text documents on small screen PDA devices, and later as WEST [6] a web browser for small screen devices. The basic idea for context-zoom was to let the current page be displayed in the centre of the screen, while all other pages are visualised as small thumbnails. A click or a tap on a thumbnail page, which zooms it into focus, flips pages.

An approach to face the context problem of zoom interfaces were the ideas of Pook et. al. [17]. The semantic zoom hides displays condensed information in overview state. When users zoom in details pop up. However, the information pop cause users to lose context; the inspected information and overview information does not have the same visual appearance and visual cue. Pook et. al. introduced a historic layer and a tree overview layer for people to guide their way through the information space.

Spatial Semantics Another approach to find new ways to think about interaction is Dieberger and Franks' City Metaphor [1]. A metaphor that embodies Lynch's Image of a City [12] Dieberger and Frank propose that cities can stand as roll model for navigating information.

Dourish et. al. have developed another use of spatial semantics for information navigation and retrieving documents in an ordinary desktop system [16]. The basic idea is that attribute meta-tags and spatial semantics can help finding and reminding document files.

Flatland [13] is an augmented whiteboard interface. The interface is a big white board where everything resides during a collaborative work-phase. Flatland was designed for activities of thinking, pre-production tasks, everyday content, clusters of persistent and short lived content, and semi-public to personal use. Flatland has no pan or zoom technique.

Sens-A-Patch is hierarchical context preserving spatial semantic label cluster visualisation technique invented by Löwgren [11]. The main inspiration from this work was in the early prototypes to display a hierarchical and condensed view of information content.

Islands of Music [15] is a geographic maps metaphor to music collections. Islands represent music genres. Closeness in space maps similarity of genre and piece. Music is analysed by Self Organising Map (SOM) clustering technique [8]. What Pampalk suggests is that spatial semantic for organising content information can be automated.

7 Research Results

The thesis present a content centric data surface interaction paradigm. This paradigm have been applied to music creativity improvisation. The paradigm state that all content must be presented on an infinitely large two-dimensional data surface mainly navigated by zoom. There are no windows, no application programs, and no menus. A database keeps persistent storage of information content. Information has only one state opposite to file systems for which information is either opened or closed. Commands are written by contextual aware text completion. Real-time synchronising of data surface contents supports collaboration.

7.1 Persistent storage

Persistent storage of content provide modelessness of content. There is no transition from closed to open state as with document files of desktop systems. Content is always displayed as "opened". The user does not have to be responsible for transferring content from secondary to primary memory of the machine. Content is always persistent, so there is no need for an explicit save action or save command. The representation of content is bookkeeping the actions supporting multiple steps of undo sustaining the forgiveness value.

7.2 2D Data Surface

The content is visualised on an infinitely large two-dimensional surface. Everything is in the open. The visualisation model leverage on spatial semantics and humans' metal models of the environment such as cognitive maps and cognitive collage. The location of content on the surface remains static and stable yielding to preserved stability value. Navigation is done by zoom and pan. Language style navigation is brought by incremental search.

7.3 Fluent Zoom - No Windows

Non-semantic zoom without based on level of detail reduction from the game development community remedy users loss of context with semantic zoom interfaces. Stateless fluent transitions from out zoomed over view to in zoomed focused view provides context of whereabouts since the appearance of the content only change in scale, there is no poping of information. Fluent transitions makes users understand that the sequence of images displayed on the monitor as responding to their action constitutes zoom.

More profound support for the modelessness value was achieved by removing the familiar model overlapping windows. The transition from closed state of a file to open by opening a window displaying the file's content was replaced by zooming in on the content.

7.4 Content Centric - No Applications

A ubiquitous component model sustains modelessness of service. It is not my intention to present such a component model, but merely the users experience of it, assuming it exists. In such an environment service components supply functionality for the contents allowing user to focus on the content and not on the tools. Users experience the components as always running and always there. The model allows only one component for each service, thus in contrast to application programs found on desktop interface systems, users have only to learn one behaviour a service instead of multiple behaviours, hence a more thorough support of the consistency value.

7.5 Incremental Command Invocation - No Menus

A command invocation model that replaces the menus found on desktop systems was design by leverage on the noun-then-verb part of the see-and-point value, the consistency value, the feedback and dialogue value, and the forgiveness value. Commands are invoked for selected or appointed content, called the selection set. Content in the selection set - *the noun* - afford the commands - *the verbs* - based on the present service components for that content. By hitting a help key, the user can probe the available commands for the selection set, listed in descending ordered of the expected frequency of use. To select the command of interest the user types it, each key-press returns a list of the commands containing the typed substring fulfilling the feedback and dialogue value. The user actually conducts an incremental search for the command, contrasting to the navigation and recognition users have to do in current desktop systems.

The command item order must be stable to support the consistency value; the same substring must yield the same command provided the command is available in the selection set. The first command item can be invoked by hitting a command completion key, but before this key is pressed the result is pre-visualised which enables the user to evaluate the result and possibly withdraw before the action is submitted; this scheme weaves forgiveness into the model.

7.6 Collaboration

Real-time synchronizing of data surface contents supports collaboration, mutual awareness of action, and mutual modifiability. Every action and command is echoed to the other collaborating participants machine. The visual and acoustic contents appearance are exactly the same on all participating machines' shared space, thus all participating users are aware of all actions. Mutual awareness of action and mutual modifiability is inconsistent with the original interpretation of the user control value, since other agents (of flesh and blood or of ones and zeros) than the user itself can change the content, thus all actions are not initiated by one user.

7.7 Subject Satisfaction

User evaluation, 10 users for the final prototype, showed that the interaction model of the data surface paradigm were appreciated by the users. They had no trouble with navigation, they explicitly appreciated the modelessness of content, in other words they were happy to not have to open files. The largest diversity was for command invocation, one user did not like it at all an insisted on a menubar with menus whereas other users really enjoyed the command invocation design. One very young user thought it was really revolutionary to write the commands. Not long ago the revolution was to select commands from menus.

8 Outline of the Thesis

The chapters of the thesis will have the following outline and content:

- **Introduction.** This chapter describes the initial case, a brief history of the desktop metaphor paradigm, and what the fundamental design principals have been for my work.
- **Scenario.** This short chapter describes an interaction scenario as introduction to the paradigm, to the application area, and to to the primary expected and observed users.
- **The Values of the Desktop.** Here I go through and reassess the fundamental design values of the desktop metaphor interface. The values come from the Macintosh Human Interface Guidelines [3]. Each value is criticised by Gentner and Nielsen from their article The Anti-Mac Interface [4].
- **Sources of Inspiration.** I go thorough all the related work, how it affected my, and why.
- **Spatial Semantics.** Takes a deeper look into the some of the cognitive psychology for this work. It is also combined with observations and interviews.
- **Application Area.** Here I present the music application area selected for the prototypes, and why it was selected. I also present observation of collaboration test using desktop paradigm music creativity tools.
- **Early Concept Prototype.** In this chapter describes the design rationale and the evaluation of the early concept prototype.
- **Fluent Zoom Prototype.** In this chapter describes the design rationale and the evaluation of fluent zoom redesign from the results of the earlier studies.

- **Music Tool Prototype.** With this chapter I close the prototype design, implementation, and evaluation iteration. The prototype design rationale comprise results from from all previous chapters. Evaluation of the prototype is also presented.
- **Data Surface Guidelines.** This chapter presents a practical guideline to be used by interaction designer and software developers who want to implement software based on the ideas in my thesis. It also contains a table comparing the design values of the desktop metaphor paradigm, the anti-mac paradigm, and the data surface paradigm.

9 Conclusions

By leverage on cognitive science, observations, interviews, and usability evaluations I have been able to find strong indications that the approach presented in the thesis support users' expected services, users' creativity in action, and users' awareness in collaboration, in a manner that the users find fresh, fun, and pleasing. The validity and the reliability of the results comes from triangulation of conceptual studies, empirical studies, and technological studies. The conceptual studies contains the related work, the reassessment of design values, and the field of cognitive psychology. In depth interviews, focus groups, qualitative evaluations in controlled environment, qualitative evaluations by field studies constitutes the empirical studies. The technological studies were the implementation of the prototypes using a multimedia tool, and later on general purpose programming language and open source game development APIs.

From the activities listed above I can conclude that users do not need windows, icons, and menus to experience interaction described by them as free, good for creativity, easy, and fun.

10 Plan for Remaining Activities

- Present the proposal. 2004-10-06
- Revise and iterate the thesis.
- Print the thesis. 2004-11-07
- Present the thesis. 2004-11-23

References

[1] Andrew U. Frank Andreas Dieberger. A City Metaphor to Support Navigation in Complex Information Spaces. In *Journal of Visual Languages and Computing*, volume 9, pages 597–622, 1998. LNCS.

[2] James D. Hollan Benjamin B. Bederson. Advances in the Pad++ Zoomable Graphics Widget. In *Third Annual Tcl/Tk Workshop sponsored Toronto, Ontario, Canada*, July 1995.

[3] Apple Computer. *Macintosh Human Interface Guidelines*. Addison-Wesley, 1992.

[4] Jakob Nielsen Don Gentner. The Anti-Mac Interface. In *Communications of the ACM*, August 1996. ACM Press.

[5] Peter Gärdenfors. *Conceptual Spaces*. MIT Press, 2000.

[6] Lars Erik Holmquist, Staffan Bjoerk, Johan Redstroem, Ivan Bretan, Rolf Danielsson, Jussi Karlgren, and Kristofer Franzn. WEST: A Web Browser for Small Terminals . In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, November 1999.

[7] David Fox Ken Perlin. Pad an Alternative to the Computer Interface. In *Proceedings of SigGraph 93, ACM Press 93*, 1993.

[8] Teuvo Kohonen. *Self Organizing Maps*, volume 30. Springer, 2001.

[9] Rikard Lindell. Users Say: We Do Not Like to Talk to Each Other. In *Graphical Communication Workshop*, pages 87–90, 2003.

[10] Rikard Lindell. When Information Navigation Divorces File Systems - Database Surface Prototype Results. In *The Good, the Bad, and the Irrelevant: The user and future of information and communication technologies.*, pages 76–81, 2003.

[11] Jonas Löwgren. Sens-A-Patch: Interactive Visualization of Label Spaces. In *Proc. Fifth Int. Conf. Information Visualization (IV2001*, pages 7–12, 2001. CA: IEEE Computer Society.

[12] K. Lynch. *The Image of the City*. MIT Press, 1960.

[13] Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Anthony LaMarca. Flatland: New Dimensions in Office Whiteboardsm. In *Proceedings of CHI99*, pages 346–353, 1999.

[14] Donald A. Norman. *The Design of Every Day Things*. Basic Books, New York, 1992.

[15] Elias Pampalk. *Islands of Music Analysis, Organization, and Visualization of Music Archives*. PhD thesis, 2001.

[16] Anthony LaMarca Paul Dourish, W. Keith Edwards and Michael Salisbury. Using Properties for Uniform Interaction in the Presto Document System. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 55–64, 1999.

[17] Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. Context and Interaction in Zoomable User Interfaces. In *Published in the AVI 2000 Conference Proceedings (ACM Press)*, pages 227–231, May 2000. Palermo, Italy.

[18] Ben Shneiderman. *Designing the user interface (2nd ed.): strategies for effective human-computer interaction.* Addison-Wesley Longman Publishing Co., Inc., 1992.

[19] Walter R. Smith. The Newton Application Architecture. In *Proceedings of the 1994 IEEE Computer Conference, San Francisco*, 1994.