

The post-PARC interface and the visualization of documents and of their relationships

Alain VAILLANCOURT and Yves MARCOUX

EBSI — Université de Montréal

vaillana@MAGELLAN.UMontreal.ca yves.marcoux@umontreal.ca

Abstract

We propose an evolution path for the traditional “PARC” interface found today on most PCs. The proposal, which we call the “post-PARC interface”, draws not only on concepts developed at major research laboratories like PARC and HCIL, but also on the work of independent researchers. It is based on two novel concepts: the representation of documents by visual objects that evolve during their lifetime, and the representation of relationships among documents by a system of flexible, visible boundaries. The interface also incorporates five new functions, which are described in detail.

1 Introduction

Over the last five years, there has been much discussion about what could possibly come after the traditional *direct-manipulation graphic user interface*, found today on most PCs, which we call the “PARC” interface (from the Xerox Palo-Alto Research Center, institution from which most of the concepts of the interface originate).

While some authors have proposed a downright replacement of the PARC interface (for instance, Gentner and Nielsen, 1996), others have rather proposed improving on it, in the line of the original principles of direct manipulation. Although an impressive amount of research has been conducted on possible specific functional improvements to the interface, few major projects proposed an integrated vision of a complete interface. Two of these are the works of Rao, of PARC (Rao 1997) and of Shneiderman, of the Human-Computer Interaction Lab (HCIL), University of Maryland (Shneiderman, 1997), who came up respectively with the concepts of Z-GUI and GENEX.

Naturally, these proposals were strongly influenced by the sum of research and prototyping that had been conducted respectively at PARC and HCIL over the last 15 years, and included very few, if any, of the improvements otherwise proposed in the works of independent researchers. As a consequence, there is today no general, integrated reflection as to how the PARC interface could be improved, given the complete actual state of technology and research. Such an integrated vision is important to guide the next development and implementation steps of the “post-PARC” interface.

Our goal is to propose an integrated vision of a post-PARC interface, that draws not only on concepts developed at PARC and HCIL, but also on the work of independent researchers. For example, we include Kandogan’s elastic windows, as well as Beheshti’s PACE system for Online Public Access Catalogs. We concentrate exclusively on the graphic document representation aspects of the interface.

Our proposal includes two major novel concepts: the representation of documents by visual objects that evolve during their lifetime, and the representation of relationships among documents by a system of

flexible, visible boundaries. The interface also incorporates five new functions, which are described in detail later.

We present our reflection by describing the graphic aspects of a hypothetical integrated virtual desktop interface. Section 2 presents the limitations and implementation considerations of our proposal. Section 3 describes the general features, and Sections 4 through 8 cover the five new functions of our interface. We conclude by indicating avenues for further development. References to related work are given throughout.

2 Limitations and implementation considerations

We have limited our horizon of reflection in two ways: First, we restricted ourselves to considering hardware and software technologies likely to be economically available within a five-year span. Thus, we did not include techniques or innovations that would require the complete development of new environments or new hardware, such as Pad++ (Perlin 1992), Media Blocks (Ullmer 1998), and Informative Things (Barrett 1998). Second, we restricted ourselves to innovations that are in keeping with the notions of document and groups of documents represented visually on a virtual desktop. The reason for this restriction is that, in our opinion, the amount of innovations possible within the realm of the “documents on a desktop” metaphor does not warrant transgressing it at the present time.

Because of these restrictions, our interface proposal could be implemented now, using a number of existing systems as a starting point. The actual Mac OS, for instance, would require few modifications to its current functions, as would Microsoft Windows 9x, especially combined with icon placement add-ons such as Melissa Nguyen's EzDesk (Born 1997). Perhaps the most natural starting point for experimenting with implementation would be the GNOME interface for Linux and other UNIX systems, given its popularity and status of open software. Whatever the starting point, implementation could be done stepwise, by gradual integration of new functions.

3 Generalities

3.1 Glyphs and icons

As mentioned earlier, one fundamental aspect of our proposal is a novel usage of visual representations of documents on a desktop. We distinguish between *glyph* representations and *iconic* representations. The two notions are closely related, and differ more by their intent than by their nature. A glyph is a connected bitmap image (often with an outer shape of a small regular polygon) that can be placed on the interface display to represent an object (in our case, a document). The possible geometries/values of a glyph (i.e., the number, relative position, and color of pixels) are not limited *a priori*, but in order for the glyph to be practically manipulable and to convey useful information, it is usually limited to a finite range of allowable geometries/values. This range of allowable geometries/values is usually described as a multi-dimensional space, in which dimensions correspond to visual properties of the glyphs, such as width, height, color, number of limbs, etc.¹ We call this space the *glyph value-space* of the representation.

¹ We do not consider animated glyphs, whose geometry/value would vary through (microscopic) time.

In representing documents by glyphs, the visual properties of the glyphs are usually linked to properties of documents, by associating each dimension of the glyph value-space with the magnitude range of some document property (for instance, size in bytes, number of embedded images, creation date, etc.). Variations in glyph values intend to reflect variations in document properties.

A common way of showing glyphs that represent documents is to place many of them in an n -dimensional space (projected onto a two-dimensional region of the display area), with each of the n dimensions corresponding to an additional document property (typically, metadata) not represented in the glyph value-space. Thus, document properties are represented not only by the geometry/value of the glyphs, but also by their relative placement within an n -dimensional space.

An *iconic* representation of documents can be thought of as a particular type of glyph representation but, as we shall see, there are also differences in the intent. As a glyph representation, an iconic representation is characterized by a single geometry (and thus, size) for all glyphs, which are called *icons*. Moreover, there is no necessary connection between a dimension of the glyph value-space and the magnitude range of a document property. Any icon (within the value-space) could be associated with any document. In fact, the glyph value-space might not have any natural “dimension” at all, the extreme case being a discrete, one-dimensional collection of arbitrary, unrelated glyph values.² A typical example of iconic representation (as defined here) is an environment in which the user can explicitly assign any icon from a predefined library to any document. An environment in which the user can draw new icons from scratch and assign them arbitrarily to documents would also be an iconic representation, provided all icons have the same size.³

Typically, when icons are placed within an n -dimensional space, they are placed explicitly by the user, and not based on the value of some document properties. In terms of metadata, we could say that the icon associated to a document in an iconic representation, as well as its placement (if any) within an n -dimensional space, do not *represent* metadata about the document, but rather *constitute in themselves* metadata associated with the document.

An interesting aspect of icons is their scalability, not only upwards to larger display sizes, but also downwards, to *smaller* displays. Indeed, while you will fit less icons on a small display and more on a large one, you are still using the same basic representation. This would not be the case of representations using very large or unbounded glyphs, or relying entirely on an automatic placement of the glyphs within an n -dimensional space. Such a representation would lose its interest if only a small part of the space could fit on the display. This scalability aspect is important, given the increasing popularity of handheld, palmtop, and other types of portable computers, and the foreseeable advent of very large displays, based for example on OLED (Organic Light-Emitting Diodes) technology (Gross 1998).

It is worthwhile pointing out that the use of an iconic representation for documents is not necessarily detrimental to other types of representation. Thus, in our interface, icons are combined with textual elements, namely file names, much in the way they presently are in current Microsoft Windows 9x and Mac OS systems. We do not include more sophisticated devices, like aural representations, because, although some research is being done in this line (for example Lo Presti 1996 and Fernstrom 1997), very

² The real degenerate case is a single icon for all documents, which is the common approach in Windows applications. This, in a sense, negates the value of icons for document retrieval and recognition by the user.

³ Our restriction to icons of identical size may be deemed too strong. It would probably suffice to have lower and upper bounds on the size of the icons. However, if we allow icons to include “transparent” pixels, this restriction becomes less of a problem.

little is actually implemented in existing systems. Neither do we seek leveraging textual elements beyond the inclusion of file names, though we suspect a lot of improvements could be achieved in this area.

3.2 *Evolving iconic representation*

The representation we adopt for documents is an iconic one, but one in which the choice of icon for a document, as well as its placement on an n -dimensional space (typically, the two-dimensional desktop), result from a *combination* of automatic calculations based on document properties, and of explicit user intervention. Moreover, the icon associated with a document, as well as its placement within the various display areas, can change during the lifetime of the document, and can also be temporary (for instance, temporary linear rearrangement of icons by document size, with later restoring of the default, user-assigned positions).⁴

A fundamental aspect of our interface is that, to the greatest possible extent, the system will always propose a number of icons to the user for assignment to a newly created or acquired document. These proposed icons will include not only icons from standard libraries, but also icons computed from document properties, as well as icons “imported” from various sources, including the source of the document itself (in the case of an acquired document), or the source of the original material from which the newly created document is derived (for example, the visual representation of the result of a database search). This implies an “iconification” function, to which we shall come back shortly.

3.3 *Reversible, two-sided icons*

Another important—and original—aspect of our iconic representations is the fact that icons are always considered to be two-sided, that is, with a front-side and a back-side. Icons can be “flipped” by selecting the appropriate item (usually, the first one) in the “contextual menu” of the icon. The contextual menu of the icon can be displayed in the usual ways, for example, by clicking on the icon with the right-hand side button of the mouse. The contextual menu could also be bypassed by adopting the convention that a double-right-click on the icon flips it.

Icon flipping is, by default, temporary, in that a flipped icon automatically returns to its front-side when it loses focus (e.g., the pointing device leaves it), or a minimum delay has elapsed, whichever occurs last. However, another item of the contextual menu allows for permanent flipping. Although the icons themselves are not animated, the actual flipping of icons are (which can be done with a simple, efficient algorithm). Any environment that displays many icons also has a “global-flip” function, which flips (permanently) all the icons in the environment to their front- (or back-) side.

The metaphor of reversible icons actually allows two different icons to be associated to a document, and moved about together with it in a cohesive way, in all contexts. This is *not* like having different icons for a document in different contexts. Two-sided icons allow us to “pack” more information into the display area occupied by an icon, but the type of information shown on the back-side must be chosen carefully.

We sometimes distinguish between the “front-icon” and the “back-icon” associated to a document, or else, treat them as a whole and just talk about “the icon” associated to a document.

⁴ These changes through time take place on a *macroscopic* time-scale (the lifetime of the document), as opposed to changes involved in icon animation, which take place on a *microscopic* time-scale.

3.4 Usage, version, and weight icons

As mentioned earlier, the choices of icon for a document result from a *combination* of automatic calculations based on document properties, and of explicit user intervention. Our interface has a set of “standard” system-computed icons that represent various document properties typical of “directory” metadata normally maintained by the operating system. These include document usage, document version, and document weight icons. The document usage icon represents the usage (read/write) history of the document, the document version icon represents the version history of the document, and the document weight icon represents the “weight” of the document, that is, some size measure appropriate for the type of document, e.g., the number of words for textual documents. These icons are always dynamically updated to reflect all document operations (which may also occur through a network).

Some of these “metadata” icons are linked to unbounded one-dimensional properties of documents, for example weight. In these cases, icons have a one-dimensional value-space, with a user-selectable scale defaulting to a saturating logarithmic scale. The rate of progression, corresponding to the base of the logarithm for logarithmic scales, is also user selectable, with a default value related to the number of elements in the glyph value-space and to the typical range of the actual document property represented. The actual glyph value-space is user-selectable from a set of value-spaces based on abstract visual elements progressing linearly in a one-dimensional way, for example: two-part pie-charts, n -edge polygons, etc. We purposely avoid “gratuitous” metaphors (thermometers, bottles filling-up, etc.) for this type of one-dimensional data.

Other metadata properties are inherently time-based, for example, usage and version. These provide an opportunity for innovation in visual representation. For example, we have started experimenting with calendar-like icons indicating visually on which days of the past year the document has been read, written or both.

We have also started experimenting with icons combining two and even three metadata properties. The key idea is that these system-generated and updated icons complement the explicit icon assignment by the user, and do not replace them. They are ideal candidates for back-icons of documents; in fact the default back-icon for a newly created document is always one of these standard icons, with a user-selectable default choice. Environments displaying many icons have, in addition to the global-flip function, a “global back-icon set” function that assigns any of the three system-computed icons as a back-icon to all documents in the environment.

3.5 Icon sizes and colors

We believe that the interface must offer various icon sizes, for use in different environments, contexts, etc. We do not have any strong opinion as to the “right” number of different icon sizes to offer, nor as to what the actual sizes should be (except in a few cases which we point out). By default, we adopt the sizes currently offered by Microsoft Windows 9x systems, i.e., 16x16, 32x32, and 48x48 pixels (except in the OPAC function, which uses different sizes). Since the number of colors is significant for icon useful recognition (Brami 1998 and Calvo 1996), we assume a palette of at least 65536 colors.

4 Generation and positioning of icons

4.1 *Iconification*

At the heart of this proposal is a function which takes images and “iconifies” them, that is, transforms them into fixed-size icons. These images can come from several applications, such as those outlined in the next four sections. In addition, original icons can be derived from images created in any drawing or image-generating application the user happens to have.

The goal of this proliferation of distinct icons is to permit users to develop a graphical milieu on their virtual desktop. This is the virtual equivalent of the build up of an office environment using small paper memos, paper sheets, paper spreadsheets, etc. In this way, direct manipulation can exploit the spatial memory capabilities of humans and apply it to data management over short and long periods of time.

The iconification function takes images from any application, transforms them into icons, stores them in a library, and offers them to the user as icons to be assigned to a document whenever a document is acquired or created, for example when the user first saves it, or saves it again under another name.

4.2 *Icon placement*

The user can group document icons on the desktop in three different ways. First, it is possible to simply place icons explicitly close to (or distant from) each other (but not on top of each other). In this case, spatial proximity indicates topic or usage closeness. Second, one can group icons in directories, with a discrete boundary system discussed in detail below. Third, one can do the equivalent of clipping several documents together with a paperclip by dragging one document on top of the other. After such an operation, the system will visually respond by always showing the group of icons together, namely touching each other (which cannot be achieved by explicit positioning).

Another desktop function is the capacity of document icons to miniaturize themselves and “squeeze” themselves off in a corner of the desktop when an application or directory is opened. Thus, they are always in sight and never disappear completely (unless explicitly requested by the user). When the application or directory is closed, all the icons which had squeezed themselves out of the way bounce back to their former positions on the desktop. This feature is inspired by Kandogan’s elastic windows (Kandogan 1997).

The most important desktop function is perhaps the discrete boundary system that allows showing the contents of directories (and subdirectories) as groups of documents on the desktop. The idea is to avoid presenting the contents of a directory in a separate window superimposed to the desktop. Upon opening a directory, instead of using a separate window, the objects contained in directory are placed directly on the desktop, but with the boundary of the directory indicated by discontinuous lines (or other regularly spaced small objects). In this way, the eye will not “bounce off” the solid lines of a separate window; nevertheless, the open boundaries are very present to the eye and serve well their role of delimiters (Tufte 1997).

A major use of this system of boundaries is for the spatial positioning of documents by the user, who can then use spatial memory for retrieval. But it can also be used automatically by the system to present incoming documents (e.g., from instant messaging, e-mail, or broadcast channels) directly to the desktop, while not “mixing them up” with the other documents. With the discrete boundaries, these objects are

more visible than if they were in a separate window, yet, they do not interfere significantly with the rest of the desktop.

5 Information Visualization and Scientific Visualization

In this section we propose the regular importation of two types of images to the desktop, small glyphs, and compact shapes expressing hierarchical or other data in a relatively full context

The field of information visualization is in many ways a continuous spin-off from the field of scientific visualization (Hays 1997, De Fanti 1990). It shares many techniques, and among them we find the use of glyphs and glyph assemblies to represent data points on a starfield display or scatter chart (Ahlberg 1994). Glyphs are usually very small images, mostly well below the smallest 16x16 pixel standard icons. In most of early works, glyphs did not vary much in shape or size, but recently, variable shape glyphs expressing a wide variety of data for a same point have appeared. In the work of Fox et al. (Fox 1993), variable glyphs have been investigated for searches through full text databases to represent quantitative data, such as publication date, as well as nominal or qualitative data, such as author name, of any given document. Vaillancourt (Vaillancourt 1998) has proposed the use of binary glyphs to represent data on documents, mapping directly the absence or presence of keywords or other nominal document information, to the absence or presence of features on a glyph. Glyph recognition during the search and long term memorability would be enhanced by offering the user a variety of possible basic glyph shapes before the start of a given search.

Thus, a typical document search session based on information visualization with variable glyphs is likely to produce a large quantity of distinct, significant glyphs. Each time a transfer or a “save” of database elements is made, icons generated by the iconification function from those glyphs are offered to the user. For example, a document saved as a Word file from a search in a full text database would be assigned an icon bearing the image of a glyph chosen by the user from a selection of those that have appeared during visualization of the database search results.

The same is true for a user taking numerical extracts from a database in glyph-based scientific visualization software. The system offers a choice of possible icons of the file containing the saved data by looking at glyphs used during the last session with the software. In both cases the small glyph images would be often “blown up” to regular icon sizes.

Sometimes, information visualization software expresses search results or small databases by means of compact assemblies or shapes which place a document in context within a single screen. This is the case of Spoerri's Infocrystal (Spoerri 1993), Noirhomme-Fraiture's Zoomstars (Noirhomme 1998), and a variety of others such as the Aleph (Das Neves 1997), and the CHEOPS system (Beaudoin 1996). These are usually “compact” images designed to fit within a single screen. If we think in terms of a bitmap display, most could fit within a 600 by 600 pixel square.

In our post-PARC system an iconification process similar to that of the glyphs is proposed for such images, with the difference that they are scaled down instead of blown up, to fit the standard square 16-32-48 pixel series.

The big advantage over glyphs is that these images show context. The big disadvantage is that when small contextual changes are concerned, the scaled down iconified versions do not show much visible differences.

6 E-mail and instant messaging

This aspect of our proposal is the most far reaching into the future. For a full implementation, it would require several participants and several institutions adopting the same practices and certain identical standards in messaging. It does not require new technical standards and there is no significant increase in the bandwidth needed, but all must exchange the same basic types of icons for everything to work gracefully.

However, not all messaging traffic is over the broad expanse of the internet. Much of it is within corporations or institutions, which can implement standards for their use of messaging, on the internet between their different offices or within intranets and extranets.

At the heart of the system is the automatic sending of a minimum of two icons with each message. This is the default mode of the messaging system. These two icons are the institutional icon and the individual icon. For instance, a message sent out by a user from Sherbrooke University would always carry along a Sherbrooke University logo icon and the user's portrait icon. If documents are attached to the message, then their icons are sent along as well.

All of these icons are of the 48x48 pixel type, in order to ensure facial recognition. A user who does not supply a portrait icon to the system has a choice of several generic or caricatural faces to send instead. These range from blown-up versions of the extremely simple "emoticons" made up from punctuation marks and used traditionally in traditional e-mail, to more complicated faces made up from basic elements. The more complicated pieces have a great number of changeable elements, much like the software for making composite faces for police ID purposes, but all those in between are simple incremental build-ups from the traditional keyboard "emoticons". If no emoticon or other face is specified by the user the system defaults to the last emoticon used in the last message. This carryover from e-mail tradition is very important since it builds on an established basis for virtual communications.

The sender of the message thus always sends out the two icons, institutional, and "portrait", unless s/he goes to the control panels and chooses to send only portraits, only institutional logos or none of the two. There is also in that same deeper choice of defaults the possibility of sending out document icons.

This option of sending out a document icon is offered as an option for all messages, but it is the default whenever the system detects a cut and paste operation from a document to a message or an attachment of a document to a message. This is the 48 pixel version of the series of 16, 32 and 48 pixel icons assigned to each document. There is of course more than one document icon to send out in several circumstances. For instance, any document has its own original icon, distinct in shape and color from other icons representing documents made from the same software, but there is also a standard version icon, expressing the version or revision status of the document. This version icon is also sent out with the original document icon when a user chooses to send out document icons instead of facial and institutional logo icons.

Individuals exchanging several times per day or week a common document over e-mail, each working on it in turn, can choose to always see the versioning icon "flipped" in order to keep track of the status of their changes to it. Another basic option for all of the messages is for the icon to bear the beginning of the message. This is the default in the specific case of instant messaging as opposed to "normal" e-mail.

In the case of these instant messaging notes the portrait or institutional icons appear as “headers” to columns of notes, with the default presentation being one column for instant messaging notes.

The instant messaging options, when the user specifies it, is applied as well to normal e-mail, with the normal e-mail message having its start or sections from its place within the 48 pixel message icon. This is ideal for those circumstances where one sends out very short e-mail to acquaintances or colleagues sharing intimate knowledge of a topic, personal or professional. Telenotes, a research project at Lotus Corp. has studied much of this (Whittaker 1997).

In the case of instant messaging or very short e-mail messages the entire message fits into the 48x48 pixel icons space, giving users the ability to view columns of “dialogues.”

As a receiver, the user has two choices, applicable to all messages in general or only to messages from specific institutions or individuals. For example, the user chooses that all messages ending with @magellan.umontreal.ca show the portrait icon and that all messages ending with @mcgill.qc.ca show the document icon, or vice-versa. The basic choices always depend on the sender's address. Within these two choices, there are other choices such as showing the original document icon or the versioning or revision icon.

All of the message icons end up on the user's desktop, as a default, unless the user specifies that only 16 pixel reminders in small numbers appear or no messaging icons at all appear on the desktop surface. As in the other sections of this proposition, expanding windows automatically make the message icons retract to smaller 16 pixel sizes off to the sides instead of covering them over.

Still another option given to the user receiving messages is to turn on a text analysis tool used also for creating original icons from word processing documents. This text analysis tool is a form of information visualization software working only on the icon shape and color instead of the icon placement on a 1D or 2D display as well as icon shape and color, as is the case for the information visualization of databases in the second section of this proposition.

7 Intranet Web browsing with icons

Our proposal is based on the use of (iconified) thumbnail images for representing Web pages. We distinguish between intranet browsing and Internet browsing, because the topology of hyperlinks is much simpler or easier to discover in an intranet setting. We concentrate here on the intranet case.

Normal browsing is aided by “hierarchical access structures” to the Web under consideration. These structures can be static, such as a corporate filing structure in the case of an intranet, but are more typically dynamically generated, such as the “community” hierarchies generated by hyperlink topology analysis in Kleinberg's HITS (Gibson 1998). Each user can use any number of access structures, including all the static ones built into the intranet, but also dynamically generated ones, which each correspond to an interest profile of the user. To each profile are associated a “topic definition”, typically a query for a search engine, and an analysis technique, for example, Kleinberg's HITS (in this case, the topic definition would be the initial search engine query used to build the root set of the analysis). The simplest “analysis” technique would be a simple hierarchical drawing of a (arbitrary) spanning forest of the set of pages “matching” the topic definition. We restrict ourselves to hierarchies in order to avoid too complex topologies that are difficult to show and difficult to interpret. Analysis techniques that do not

yield hierarchies would not be directly integrated in the interface, but rather considered applications that the user has to invoke explicitly.

The profiles are used to dynamically analyze, as background tasks, the changing structure of the Web and update the corresponding access structures. As new analysis techniques develop, they can be integrated in the system and offered as a choice in the definition of profiles. Profiles and static access structures are accessed much in the same way as directories, by “opening” them.

When an access structure is opened, its corresponding hierarchy is displayed in a natural top-down fashion, as in Sévigny and Marcoux’s “detailed structure guide” (Sévigny 1996, figure 2), with simple arrows representing hierarchical links, but using thumbnail 48x48 pixel icons for Web pages. By default, the back side of a Web page icon is a “metadata” icon showing the weight of the page and the number of outgoing links (all of them, not just the hierarchical ones displayed). The number of incoming links (all those known) can also be shown, if they were counted by the analysis technique. A distinctive visual element (for example, a light frame around the icon) will alert the user that a page has changed since s/he last visited it. Of course, double-clicking an icon will go to that page in normal browsing mode.

When a Web page is copied to the desktop, its thumbnail icon is automatically transferred with it, and when an excerpt of a Web page is used in a fresh document or message, the page icon is automatically proposed as a choice when first saving the document or sending the message.

8 OPAC for digital libraries

This function presents the users with thumbnail representations of books in three sizes, depending on the context of the view chosen: 60x90 pixels, 48x48 pixels, and a flattened 16x90 pixels. The “flat” 16x90 pixel form shows a generated image of the spine of the book. Much like the PACE system developed by Andrew Large and Jamshed Beheshti, it is used to show each a series of books, in context within an order determined by the user. Again like the PACE system the relative thickness and the height of the book images are generated by information in MARC records such as the height of the book and the number of pages (Beheshti, 1996).

Unlike the Pace system, however the book series are not shown as if they were on a shelf but as if they were in a pile of books on the floor, thus making it easier to read the book titles. Any view of a single pile also lets a peek on the side to segments of neighboring piles to alert the user of their proximity and availability. The pile metaphor has been proposed and studied a great number of times for office documents (for instance, Rose 1993). We believe this metaphor is most useful for representing published documents such as books, instead of unpublished documents such as letters and memos.

At any time, the user can alternate between a view of the book piles and views of the books arranged as “bookwalls”, as in the exhibitions presented by master bookbinders in the Europe or as in bookstores, when it is advisable to show the covers of books to the public instead of their spines. As in the previous view, the order of the books (ordered by title or author or subject or other) can be rearranged at any time by the user and it is possible to browse along the series without using the keyboard, much as in the PACE system. The thumbnail series permits a better view of the cover of the book while the 48x48 pixel series permits a greater number of books to be shown on the screen.

We anticipate that eventually most digital libraries would get their hands on thumbnail images of the covers of the books they offer online, much in the same way as the present Web bookseller Amazon has

thumbnails of the covers of the most recent books they offer. However, this process could take some time and some digital libraries might choose not to buy or make such thumbnails, for reasons of cost or speed. For these instances, the system generates false thumbnails in some manners analogous to the generation of glyphs in the information visualization outlined above.

When a single book or an entire library of books does not have a thumbnail image readily available, one can pick in their places thumbnail images generated from the MARC records in two ways: By generating vertical, and horizontal bars form of contextual link to the browsing mode, and by a binary image generation coming from any non-browsing search using the search function. This binary image generation is similar in a way to the binary glyphs generated in the information visualization function outlined earlier.

The absence or presence of the thumbnail image does not make any fundamental change to the “book pile” browsing view. As in the first case, the colour and texture is derived from a sampling of the cover, whether it is from a thumbnail of the “real” cover or a generated version. Even if the thumbnails are available, any user can choose to see the generated images instead of them, as the front-icon of the book, thus replacing the default thumbnail cover, or as the back-icon of the book.

In the first kind of MARC generated book cover, the proportions of the book are extracted from the MARC record. The height is taken directly from one of the fields while the width is more or less accurate, being a simple calculation of the arbitrary ratio of 1 to 1.5 to ensure that the book looks rectangular instead of square. From book to book the ratio changes very slightly, to give visual notice to the user that one is progressing through a series of different books. Then, vertical, and horizontal coloured bars are generated, with their placement on the book cover being decided from two different fields in the MARC record and the position of the alphabet from A to Z. The default fields are the title, and author, fields. The user can at any time change these defaults, by just changing their order of bar assignment or bringing in different fields completely, making for example the vertical bar correspond to the LC subject headings, the horizontal bar correspond to the publication date, or any other mix.

The defaults will also change automatically according to the browsing mode chosen by the user. The colour of the bars is redundant and serves simply to better distinguish them. However, the user is always prompted to choose a colour for each of the bars, or accept their defaults, because of the power of chosen colours. The background is a light grey which changes very slightly in shade from book to book to give a discrete visual notification of difference from one book to another in a series, in complement to the small rectangular ratio change.

If we take as an example the default value of title, and author (or authors) corresponding to the vertical and horizontal bars, we can note that a book with the title “Enlightenment, The” and “Gay, Peter” for an author will have a vertical bar about a fifth of the width of the book from the left since the letter “E” is the fifth out of the twenty six letters of the alphabet. The same book will have an horizontal bar about a seventh of the way up from the bottom for the same reasons.

A book with two authors has two horizontal bars of two different colours at their respective positions, and so on for three or four authors. The horizontal bar is three times thicker than the vertical bar and does not touch the sides of the book frame. This ensures a relatively “bookish” look to the generated images by avoiding the “religious cross” effect or the “Christmas gift” effect that would arise with certain vertical and horizontal bar combinations. The horizontal bar always superimposes itself on the vertical bar, in the instances when it crosses it.

Intuitively, any user browsing along and “testing” the system by browsing in different orders (title, then author for instance) will grasp the correspondence of the bars to the fields by seeing one of them “move” very gradually from book to book when browsing in the same field as the bar, while other bars shift around less gradually from book to book.

The idea behind all this however, is not to give the user an exact indication at all times of the exact title or author or other item (the user has text close by to reveal that) but to give a fast visual representation of context and also create distinct images which can later be transferred to icons when one moves information from the OPAC application to the desktop. This is the same for the binary image generation when a user is searching for a precise author or title or other instead of browsing.

As in the information visualization section, the absence or presence of a search term is expressed visually by the absence or presence of a colored “slice” on the face of a glyph. In this case the face of the glyph is the rectangular shaped book cover instead of the circle of the abstract glyph seen previously.

After a user has chosen a book or excerpts of books to be downloaded to his or her system, a similar function to that of the Information Visualization Application offers a possible icon for the book. The user can then choose between a variety of 48x48 pixel icons, all of them featuring the proportions of the books drawn from the MARC records.

9 Conclusion

We have proposed an integrated vision of a post-PARC interface, that draws not only on concepts developed at PARC and HCIL, but also on the work of independent researchers. The two major novel concepts of our proposals are the representation of documents by visual objects that evolve during their lifetime, and the representation of relationships among documents. The interface also incorporates five new functions, which were described in detail.

Although some of the characteristics of our proposal are derived from previous research work, others are original and were inspired either by readings or by years of observation (by the first author) of real people managing real and virtual documents. These original elements, like two-sided reversible icons and “metadata” icons, would naturally need to be evaluated for efficiency. They would need to be evaluated separately, but on a more global level, the interface as a whole would need to be incrementally evaluated, because two ideas good side by side do not necessarily combine well within a single product.

One of the natural first steps towards testing the ideas in this proposal would be evaluating the efficiency, on document retrieval tasks, of recycling database search-result visualization-glyphs as document icons through an iconification function. We are currently starting up a research project in this line.

10 References

Ahlberg, Christopher; Shneiderman, Ben. "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays". In: CHI 94: Proceedings of ACM SIGCHI 94, Boston. New York: ACM Press, 1994. pp. 313-317.

Barrett, Rob; Maglio, Paul P. "Informative Things: How to Attach Information to the Real World". UIST'98 Eleventh Annual Symposium on User Interface Software and Technology. Proceedings of the

ACM Symposium on User Interface Software and Technology, San Francisco, California, USA
November 1-4, 1998. pp. 81-88.

Beaudoin, Luc; Parent, Marc-Antoine; Vroomen, Louis. "Cheops: A compact explorer for complex hierarchies". InfoVis 96: Proceedings of the 2nd IEEE Symposium on Information Visualization, San Francisco, California, October 1996. Los Alamitos, Calif.: IEEE Computer Society Press, 1996. pp. 87-92.

Beheshti, Jamshed; Large, Valerie; Bialik, Mary. "PACE: A Browsable Graphical Interface". Information Technologies and Libraries. December 1996 pp. 231-240.

Born, Guenter. Inside the Registry for Microsoft Windows 95. Redmond, Washington: Microsoft Press, 1997.

Brami, Russell. "Icons: A Unique Form of Painting". Interactions. September + October 1997. Vol. 4 No. 5. pp. 15-26.

Calvo, Alex. The Craft of Windows 95 Interface Design: Click Here to Begin. New York: Springer-Verlag, 1996.

Das Neves, Fernando. "The Aleph: A Tool to Spatially Represent User Knowledge About the WWW Docuverse". In: Bernstein, Mark. ; Carr, Leslie; Osterbye, Kasper ed. Hypertext 97, Southampton UK, The Eighth ACM Conference on Hypertext, April 6th-11th 1997. New York: The Association for Computing Machinery, 1997. pp. 197-207.

De Fanti, Thomas A.; Brown, Maxine D.; McCormick, Bruce H. "Visualization: Expanding Scientific and Engineering Research Opportunities". in: Nielson, Gregory M.. Bruce Shriver ed. Visualization in Scientific Computing. IEEE Computer Society Press : Los Alamitos, Calif. 1990.

Fernstrom, Mikael; Bannon, Liam. "Explorations in Sonic Browsing". People and Computers XII. Proceedings oh HCI 97, Bristol, UK August 12-15 1997. London: Springer verlag, 1997.

Fox, Edward A.; Hix, Deborah; Nowell, Lucy T.; Brueni, Dennis J.; Wake, William C.; Heath, Lenwodd S.; Rao, Durgesh. "Users, User Interfaces and Objects". Journal of the American Society for Information Science. September 1993 Vol. 44 No. 5 pp. 480-491.

Gentner, Don; Nielsen, Jakob. "The anti-Mac Interface". Communications of the ACM. Vol. 39. August 1996. pp. 70-82.

Gibson, David; Kleinberg, Jon; Raghavan, Prabhakar. "Inferring Web Communities from Link Topologies". IBM Research Report. RJ 10122 (91928) March 26, 1998.

Gross, Neil. "Flatter, Brighter and Easy to Make? A New Screen Technology has the Industry Charged Up". Business Week. No. 3600. October 19, 1998. pp. 120, 124.

Hays, Nancy. "InfoVis 96 meets high expectations". IEEE Computer Graphics and Applications. Vol. 17 (1). January-February 1997. pp. 87

Kandogan, Eser; Shneiderman, Ben. "Elastic Windows: Evaluation of Multi-Windows Operations". In: Pemberton, Steven. ed. Looking to the Future: Proceedings of ACM SIGCHI 1997, Atlanta. New York: Association for Computing Machinery, 1997. pp. 250-257.

Lo Presti, Eric; Harris, W. Michelle. "loudSPIRE, An Auditory Display Schema for the SPIRE system". The Proceedings of ICAD 96: International Conference on Auditory Display. Palo Alto California, USA, 1996. 10 pages.

Noirhomme-Fraiture, Monique; Rouard, Manuel. "Visualisation des données multivariées: évaluation de la représentation en étoile zoom". IHM 98 Interactions Homme-Machine 1998, Nantes, France, 2 au 4 septembre 1998. Actes du congrès en voie de publication.

Perlin, Ken; Fox, David. "Pad: An Alternative Approach to the Computer Interfaces". Computer Graphics. 1992. Vol. 26 No. 3.

Rao, Ramana. "From Research to Real World With Z-GUI". IEEE Computer Graphics and Applications. July/August 1997. Vol. 17. No. 4. pp. 73-75.

Rose, Daniel E.; Mander, Richard; Oren, Tim; Ponceleon, Dulce B.; Salomon, Gitta; Wong, Yin Yin. "Content Awareness in a File System Interface: Implementing the Pile Metaphor for Organizing Information". in SIGIR 93. Pittsburgh, Pennsylvania. Proceedings of the 13th annual ACM SIGIR conference on Research and Development in Information Retrieval. New York ACM, 1993. pp. 260-269.

Sévigny, Martin; Marcoux, Yves. "Construction et évaluation d'un prototype d'interface-utilisateurs pour l'interrogation de bases de documents structurés." Canadian Journal of Information and Library Science / Revue canadienne des sciences de l'information et de bibliothéconomie, vol. 21, no 3/4, septembre-décembre 1996, pp. 59-77.

Shneidermann, Ben. Designing the User interface: Strategies for Effective human-Computer Interaction. Reading MA: Addison-Wesley, 1997. 3rd edition.

Spoerri, Anselm. "InfoCrystal: A Visual Tool for Information Retrieval and Management". CIKM 93: Proceedings of the Conference on Information Knowledge and Management, Washington DC, November 1993.

Tufte, Edward R. Visual Explanations: Images and Quantities, Evidence and Narrative. Cheshire, Conn.: Graphics press, 1997.

Ullmer, Brygg; Ishii, Hiroshi; Glas, Dylan. "mediaBlocks: Physical Containers, Transports, and Controls for Online Media". SIGGRAPH 98: Proceedings of the Conference on Computer Graphics, July 19-24 1998, Orlando, Florida. New York: ACM Press, 1998. pp. 379-386.

Vaillancourt, Alain D.M.G. "Generated Glyphs as Memorable Desktop Icons". ACM Conference on Information and Knowledge Management, New Paradigms in Information Visualization and Manipulation Workshop '98. Bethesda, Maryland, November 6-7, 1998.

Whittaker, Steve. Jerry Swanson. Jakov Kucan. Patti L. Zila. Candy Sidner. "Telenotes: Managing Lightweight Interactions in the Desktop." ACM Transactions on Human-Computer Interactions. June 1997. Vol. 4. No. 2. pp. 136-68.