ZOIL – A Cross-Platform User Interface Paradigm for Personal Information Management

Hans-Christian Jetter　Werner A. König　Jens Gerken　Harald Reiterer
Human-Computer Interaction Group, University of Konstanz
Universitätsstraße 10, 78457 Konstanz, Germany
{jetter,koenig,gerken,reiterer}@inf.uni-konstanz.de

ABSTRACT
In this paper we introduce the novel user interface paradigm ZOIL (Zoomable Object-Oriented Information Landscape). ZOIL is aimed at unifying all types of local and remote information items with their connected functionality and with their mutual relations in a single visual workspace as a replacement of today's desktop metaphor. This workspace can serve as an integrated work environment for traditional personal information management (PIM), but can also be used for PIM tasks in a wider sense. By formulating ZOIL's fundamental design principles we describe the interaction style, visualization techniques and interface physics of a ZOIL user interface. Furthermore we discuss ZOIL's ability to provide nomadic PIM environments for mobile and stationary use.

Author Keywords
personal information management, post-WIMP user interfaces, information visualization, zoomable user interfaces, object-oriented user interfaces.

ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION
Throughout its history Human-Computer Interaction (HCI) and related disciplines have been striving to research and design novel usable user interfaces that unify qualities like learnability, suitability for the task, "intuitiveness" or attractiveness. Thereby the proposed designs have always been influenced by the then-available technology and the personal, organizational or societal values of the time. Thus the user interface (UI) of the personal computer has undergone many changes to become "graphical", "intuitive", "object-oriented" [10] or "direct-manipulative" [35], but also "web-based", "nomadic" [30], "ubiquitous" [5] or "social" [14]. Most functionality has been deployed to the user as local "desktop applications", but some also as "objects", "templates" and "views" [27], as "activities" [5], or as "web widgets", "web services" or "web applications" based on dynamically generated hypertext [1]. Even the local file system has changed its face from a simple hierarchical storage structure into a versatile database with indices of content and metadata [11, 12] to allow new ways of querying and accessing personal information.

Over the decades this great diversity of usage scenarios and design goals has left many traces in today's design of our personal information management tools. Although the desktop metaphor is still thought to be the centerpiece of PIM activities and work environments, our PCs, laptops, smart phones or PDAs are packed or even bloated with a multitude of non-interoperable specialized PIM applications and websites which carry out the actual work (e.g. Google Calendar, Facebook, BSCW or Microsoft Outlook). Most of these use incompatible storage formats and inconsistent interaction models (e.g. desktop GUI applications vs. hypertext-driven web applications [29]) which have further hollowed out the role of the desktop metaphor [20].

For many PIM users these inconsistencies have led to an almost paralyzing amount of necessary workarounds and to a destructive degree of complexity and "information fragmentation" [24]. Content and functionality are scattered over dozens of applications, websites, storage formats, interaction models or devices with each one posing an individual challenge to the user's cognitive skills. Jef Raskin has identified this critical weakness of today's "mazelike" interface (Raskin, [32]) as one of the main problems standing between current technology and tomorrow's "humane interface". He regards "fundamental changes in the design of human-machine interfaces" as inevitable, since "nothing less will do". The prospects of such a fundamental change have led researchers to suggest designs for the "integrated digital work environments" of tomorrow which go "beyond the desktop metaphor" and open all new perspectives for personal information management [23]. The goal is to design a general-purpose interface suitable for many different devices which unifies all kinds of content and functionality under a consistent interaction model while leaving the user the possibilities to establish own workflows, data structures or views on her information space.

At the Human-Computer Interaction Group (HCIG) of the University of Konstanz we are designing and researching such a novel visual user interface concept named "ZOIL" (**Z**oomable **O**bject-Oriented **I**nformation **L**andscape). As an application- and platform-independent UI concept ZOIL is aimed at unifying all types of local and remote information items with their connected functionality and with their mutual relations in a single visual workspace under a consistent interaction model. This visual workspace is named the "information landscape" and can serve as an integrated work environment for traditional PIM tasks, but can also be used for PIM tasks in a wider sense like navigating public information spaces on the Web (e.g. digital libraries, social networking websites or geo-referential content in maps).

At the current stage of our project we have formulated five design principles for ZOIL user interfaces and have applied them in mock-ups and Java prototypes for document management [26] (see figure 3) and digital libraries [20] (see figure 6). The design principles are based on our previous research work on Zoomable User Interfaces (ZUIs) [19, 20, 25], visual information seeking [17] and information visualization (IV) [33]. Furthermore they have drawn from a broad scope of existing designs and theories in the HCI, IV and PIM literature.

By describing these five design principles we will introduce the ZOIL paradigm as subject of our ongoing research work to the scientific community. Furthermore we will outline the potential of ZOIL user interfaces to replace the desktop metaphor and traditional application-oriented PIM environments. The role of ZOIL user interfaces in reducing information fragmentation and providing usable nomadic PIM solutions will be a further topic of our discussion, which will then be concluded by outlining our next steps and future work.

PREVIOUS AND RELATED WORK

The first use of a zoomable virtual canvas to visually access and spatially arrange information items dates back to William Donelson's and Richard A. Bolt's seminal "Spatial Data Management System (SDMS)" from 1978 [13]. "Pad" of Perlin & Fox and "Pad++" of Bederson & Hollan provided pioneering theories and more elaborated designs of such ZUIs in 1993 and 1994: "Pad" was based on the assumption that navigation in information spaces is best supported by tapping into our natural spatial and geographical ways of thinking [6, 31]. Data objects are organized in space and scale and users can navigate them by performing zooming and panning operations. "Pad" has also introduced the concepts of semantic zooming and portals. We have merged Pad's "portals" and "portal filters" with the "see-through interface" of Bier et al. [9] and the "dynamic queries" of Ahlberg et al. [3] to create the concept of ZOIL portals which form the backbone of every ZOIL-based UI (see following section).

Jef Raskin's vision of the "ZoomWorld" and the "Zooming Interface Paradigm" from 2000 has been a further important source of inspiration: "The zooming interface paradigm can replace the browser, the desktop metaphor, and the traditional operating system. Applications per se disappear." (Raskin, [32]). Thereby Raskin's concepts of "unification" and "commands" have similarities to object-oriented user interfaces which have been discussed by Theo Mandel in 1994 [27] and Dave Collins in 1995 [10].

"Unification" is also discussed by David R. Karger and William Jones in the context of PIM and the "Haystack" work environment [23, 24]. Haystack uses a RDF-based semi-structured data model for organizing personal information items. A related approach can also be found in the DeepaMehta semantic desktop project [36].

Concerning the implementational aspects of ZUIs, Ben Bederson's Jazz and Piccolo frameworks provide valuable input on architectures and data models [8]. A very impressive implementation of a ZUI based on Microsoft Live Labs' "Seadragon" was presented by Blaise Aguera y Arcas at the TED conference 2007 [2]. This demonstration has influenced our decision to develop a web-deployed ZOIL UI software framework based on Microsoft .NET and WPF as part of our research activities.

As will be discussed in more detail in the following section, the ZOIL design principles furthermore reflect some of the findings of William Jones and Jamie Teevan about the nature of PIM, personal spaces of information (PSI) and personal information collections (PIC) [22]. These essential PIM concepts and the growing body of knowledge about PIM practice will be guiding our future design and evaluation of the ZOIL paradigm (see section "Conclusion & Future Work").

THE ZOIL USER INTERFACE PARADIGM

To describe the interaction style, visualization techniques and interface physics of a ZOIL user interface we have formulated five ZOIL design principles which will be introduced in the following sections.

ZOIL Design Principle 1: Object-Oriented User Interface

In [22, p.7] Jones and Teevan introduce the "information item" as a fundamental concept for the consideration of PIM. An "information item" (e.g. a real world printed document or handwritten note, an email message, a web page or a reference to a web page) is a "packaging of information in a persistent form […]" and with an associated "information form" or "information type" (i.e. items have types like "paper document", "electronic message", "web page" or "web bookmark"). This information form is "determined by the constellation of tools and applications that make it possible to manipulate the item."

A more formal model of this abstraction can be easily achieved by following an object-oriented (OO) approach: All information is contained in numerous interrelated information items (or objects) of different information forms (or classes). Each information form can be formally specified by a class definition and each information item can be considered as an instance of a class following this class definition. The attributes section of a class definition defines a template for the possible content, relations and metadata of the individual information items and the methods section of the class definition specifies all possible ways of interacting with the item or manipulating it.

This object-oriented model comes in handy when considering Jones and Teevan's observation that although "a person's interactions with an information item vary greatly depending upon its form" "there are many essential similarities in the way people interact with information items, regardless of their form." By using the OO mechanism of inheritance it is possible to model such essential similarities between different information forms in a common base class. If properly applied this leads to a class hierarchy of information forms which integrates very different types of information while preserving a maximum degree of consistency in attributes and methods. At first glance this might only appear interesting for programmers. However, such a consistent class hierarchy can also have a strong impact on usability and UI design when following the approach of "Object-oriented User Interfaces".

Application-Oriented User Interfaces	Object-Oriented User Interfaces
Application consists of an icon, primary window, and secondary windows	Product consists of a collection of cooperating objects and views of objects
Icons represent running applications	Icons represent objects that may be directly manipulated
Users must start application before working with objects	Users open objects into views
Provide users with function needed to perform a task	Provide users with supplies needed to perform a task
Rigid structure-by function	Flexible structure-by object
Users must follow the application structure	User may perform in their own way or innovate
Many applications required – one per task	Few objects – more reuse of the same object in many tasks

Table 1. Application-oriented UI vs. OOUI (taken from [27]).

Object-oriented user interfaces (OOUIs) were introduced to the PC GUI in connection with IBM's OS/2 operating system in the early 1990s. At that time OS/2's object-oriented approach was perceived as the direct competitor to the application-orientation in the GUI of Microsoft Windows and was regarded as more "intuitive" and "natural" by many HCI researchers [27]. However, due to the rapidly growing popularity of the Windows operating system and the advent of the hyperlink-driven World Wide Web, OOUIs have never gained much attention and momentum in UI design or PIM since then.

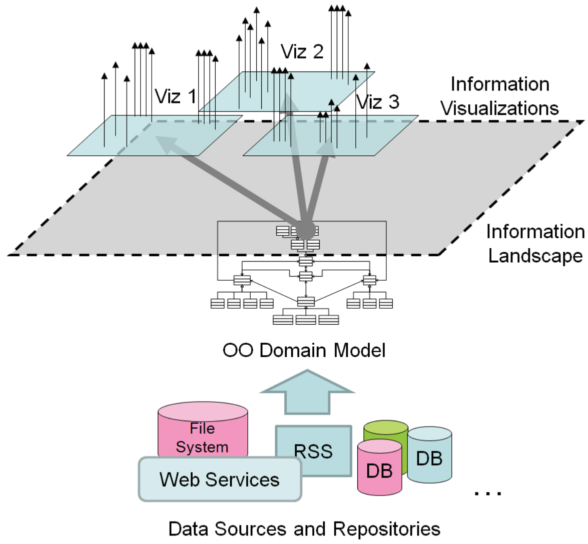

Figure 1. Domain model and data sources of a ZOIL-based UI.

One of the key concepts of OOUIs is to deploy functionality as a collection of cooperating objects and views of objects which are directly integrated as new components into a common visual workplace. Instead of creating monolithic applications running in isolated windows with a specialized interaction model, rigid workflows and non-interoperable data formats, new functionality is seamlessly integrated into the existing desktop environment. This preserves a high degree of consistency and interoperability on the UI (see table 1 from Theo Mandel [27]). Unlike in monolithic applications, tasks in OOUIs are then carried out in the desktop environment by direct manipulations in a very flexible "model world" interface instead of indirect interactions in specific applications with rigidly structured "conversational interfaces" and dialogs [18]. Therefore OOUIs appear especially suitable for such complex and unstructured "knowledge work" [10] like PIM where workflows and tasks are highly user-specific and cannot be easily automated or supported by conventional dialogs.

Only fragments of the original OOUI principles have found their way into today's desktop metaphor and applications, e.g. the object-action sequence of context/pop-up menus or the "pick, drag and drop" interaction with icons. ZOIL takes up OOUI concepts in their original sense to provide the user with a flexible though consistent direct manipulation interface for PIM.

For this reason all local and remote information items in ZOIL are regarded as objects of different classes as described in the aforementioned OO model of information in PIM. The class of an object defines its attributes and metadata fields (e.g. "From:"-field of a mail or GPS coordinates of a photo) and the possible relations to other items (e.g. a mail is sent by a person, a mail is sent to a person, a person can be a friend of other persons). Furthermore the class defines the available user interaction possibilities (or methods) connected with the object and their visibility and presentation on the user interface depending on the available screen estate and the currently active views and filters (see figure 1). The class definitions in ZOIL thereby resemble a basic domain model of PIM that constitutes a "model world" interface defining visible attributes, relations and functionality of information items including their presentation and behavior on the user interface. OO inheritance and polymorphism are then used to achieve a maximum degree of consistency in the classes and therefore also in the ways the user can interact with the system and manage different information forms.

However, an OOUI with a basic domain model might still be too restrictive to satisfy all user needs in PIM. As David Karger points out in [25] developers can hardly predict which information objects need to be stored, retrieved, and viewed or what relationships and attributes are worth storing and recording in day-to-day practice. Thus the domain model within a ZOIL user interface should be extensible and customizable by the user through easily adding new information forms with new functionality or adding further properties, relations and views anytime during runtime.

ZOIL Design Principle 2: Semantic Zooming

In a ZOIL user interface all information items from all data sources appear as visual representations at one or more places in the information landscape. However, unlike in today's desktop metaphor, this landscape is not limited to the visible screen size but resembles a virtual canvas of infinite size and resolution (as known from [31] and [32]). All items and their connected functionality can be accessed by panning to the right spot in the information landscape and zooming in. ZOIL thereby uses "semantic zooming" [31] which means that the geometric growth in display space is not only used to render more details, but to reveal that content and functionality which is of most use to the user. If the user zooms into more complex items like documents, drawings or spreadsheets they become editable on-the-spot without the need to open a dedicated application window (see figure 2). Thus the available functionality is always coupled with the information item itself as it is proposed by object-oriented user interface design.

Zooming out of the information landscape leads to a decrease of display space. The visual representation of an information item gradually collapses to an icon or glyph. Eventually an information item is represented as a single remaining pixel. Together with the neighboring pixels this pixel can then be perceived as a cluster or a part of an overview visualization aggregating a large amount of items.

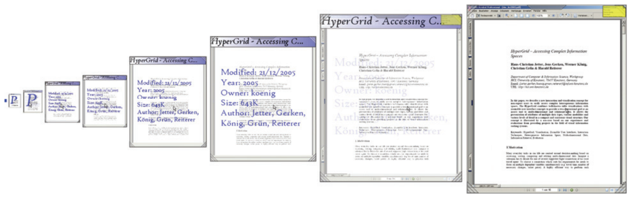

Figure 2. Semantic zooming into an information item reveals not only more content but also the necessary functionality.

An important benefit of using semantic zooming to navigate the information space is the natural "feel" and the "intuitiveness" of zooming because of its similarity to the principles of the physical world. However, a pure ZUI has some disadvantages because of the absence of peripheral views. This will be part of our future research work and is discussed in the last section of this paper.

ZOIL Design Principle 3: Nested Information Visualization

Since the number of information items in PIM can easily add up to thousands or millions of items, it is necessary to find ways to structure large amounts of items and to efficiently access them according to the user's current information need and activity. In current practice we typically use physical piles of papers on our desktops, electronic folders and subfolders on our hard disk or the inbox of our email client to establish such "islands of relative structure and coherence." [22, p.12] Jones and Teevan refer to these islands as "personal information collections" (PICs) where "people have made some conscious effort to control both the information that goes in and also, usually but not necessarily, how this information is organized". Thereby PICs should be considered "not as defined by technical format or application, but rather as defined by activities of people […]."

In ZOIL, equivalents to such PICs can be created persistently as "portals" at arbitrary locations in the information landscape by specifying a size and location in the landscape and assigning the desired information items and a visualization type (e.g. geographic maps, time lines, networks, tree views or simply an empty canvas) to the new portal. These visualizations are then used to display each contained information item according to the items' properties or relations. For example a collection of photos could appear as zoomable thumbnails on a geographic map using GPS coordinates or in temporal order within a timeline using their timestamp. Since ZOIL user interfaces make use of an object-oriented domain model, different kinds of information items can be freely combined within a single portal using their common base classes. This enables the user to place and manage different forms of information in a single PIC without being limited to certain data formats or application boundaries as is often the case in today's application-oriented work environments.

To control the information that goes into such portals a ZOIL UI offers different direct manipulation techniques. For example the user can manually assign items or item collections to a portal by using a pick, drag and drop interaction as an equivalent to copying in today's desktop metaphor. Furthermore dynamic queries [4] can be used to formulate search queries on the information space (e.g. desktop and web search) that automatically fill the portal with the result set of information items. Figure 6 illustrates how metadata like "Year" and "Persons" could be used to search for DVDs in the information space and to iteratively specify the desired items. In this example a range slider is used to specify the desired years of publication. Furthermore an incremental keyword search is used to find all DVDs with a certain cast member by typing the first letters of the person's name.

A further way to formulate complex queries is the approach of "nested information visualization". This approach enables the user to select arbitrary rectangular regions in the information landscape or in portals (e.g. with a bounding box, see figure 4) to create a new portal within the existing portal that only contains the items from the selected region. As suggested by Perlin & Fox in [31] portals can also be used like the "magic lenses" of Bier et al. [9] to temporarily or persistently view the content of a portal with different visualizations and filter criteria. Figures 4 and 5 show how a rectangular region of a map visualization is selected and is transformed into a portal displaying all documents within the selected region in a scatter plot.

Using the information landscape to create and manage multiple portals enables the user to create persistent views on arbitrary subsets of the information space dedicated to the user's typical information needs or current activities. For example the user could create a portal that is prominently positioned on the information landscape and that contains a zoomable table view of recent emails similar to today's inboxes in email clients. Figure 3 shows a screenshot of a ZOIL-based work environment for document management which contains six different portals to access documents: A person-oriented access is provided by the floor plan below "Persons" which allows exploring the documents of an author according to the location of the author's office. The calendar visualization on the top of the screen positions the documents according to their creation and modification date and allows a time-oriented zoom into their content or to keep track of recently received documents. Furthermore the user can use the world map under "Location" to zoom into scientific papers according to the geographic location of a conference (see figure 4). Further visualizations ranging from hierarchical treemaps [7] to social networks [14, 33] or zoomable tables [19] can be integrated into the information landscape by downloading them as modular "plug-ins" as it is known from today's web browsers. Thereby it is important to notice that information items in ZOIL are not located at a

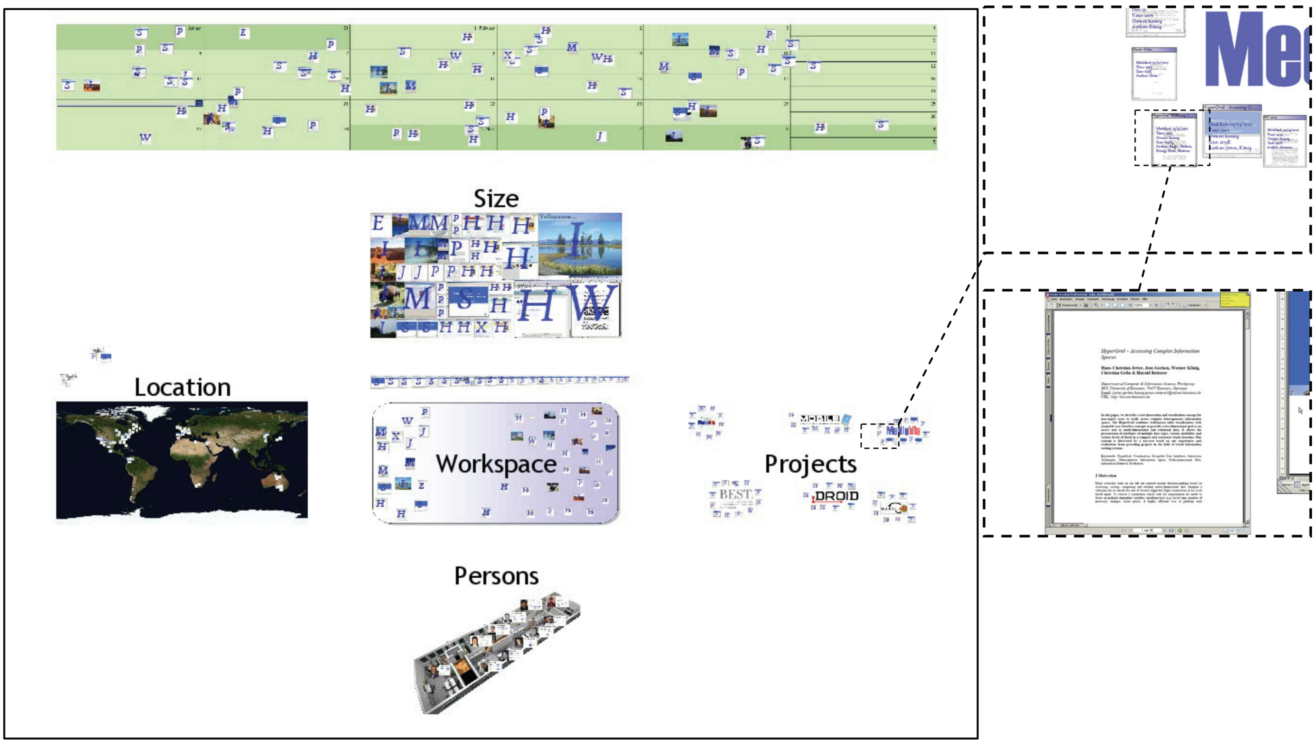

Figure 3. A ZOIL-based work environment for document management (The dotted boxes illustrate the semantic zooming in the information landscape).

single spot in the information landscape. Unless they are filtered out with dynamic queries and nesting or unless they are manually assigned exclusively to one portal, each item in the information space is placed in each of the portals. This allows a flexible complementary use of different visualizations depending on the current information need.

Figure 4. A zoom into a map visualization allows a geographical access to documents. Furthermore a bounding box is drawn to select the Eastern regions of the US.

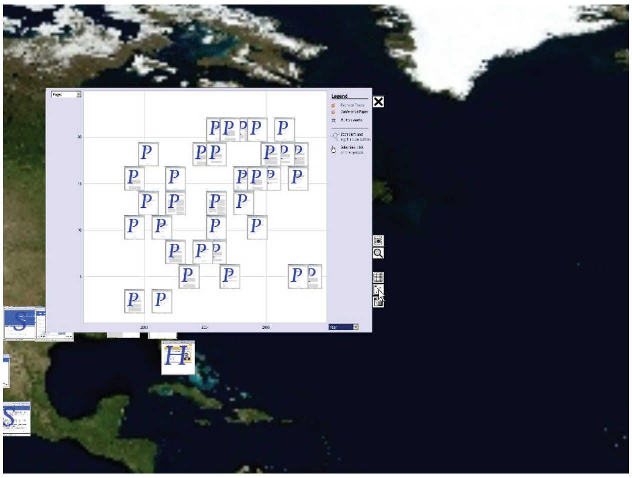

Figure 5. The bounding box in figure 4 is transformed into a portal visualizing the contained documents in a scatter plot.

ZOIL Design Principle 4: The Information Space as Information Landscape

All PIM tasks of a user take place within the user's single "personal space of information" (PSI) that "includes all information items that are, at least nominally, under that person's control." [22, p. 10] A user's PSI spans the virtual and physical world and contains a great variety of information forms ranging from the physical book in the bookshelf to the text message on the mobile phone. Since the PSI's content and boundaries are uncertain and constantly changing, an information system can neither provide an complete overview of a person's PSI nor can it process the whole PSI to search or manage all PICs or information items within. However, for facilitating PIM tasks it appears reasonable that a digital PIM environment should try to integrate and provide as much of the PSI and the contained PICs as possible – regardless of the information form or source.

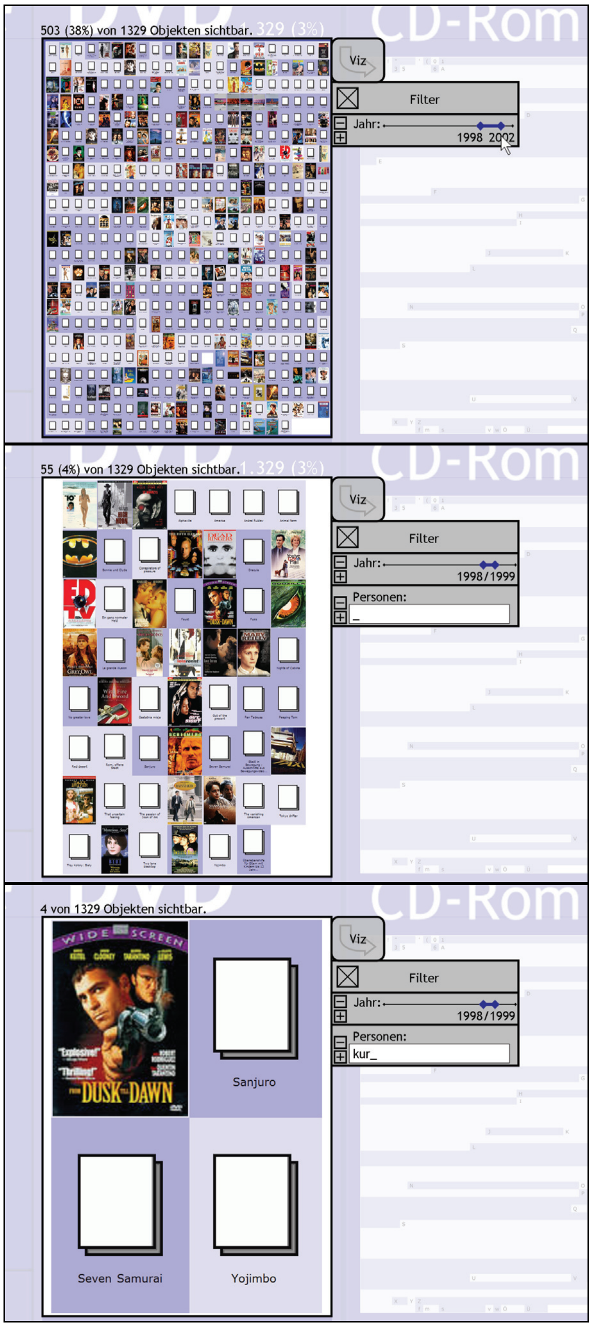

Figure 6. A mock-up of a ZOIL-based library catalog. The amount of displayed information is reduced by specifying the desired years of publication ("Jahr") and the name of a desired cast member ("Personen").

The information landscape in ZOIL can be regarded as such an attempt to visually map as much as possible of the user's PSI and to provide a visual information workspace unifying heterogeneous information items of different forms and from different data sources and devices (e.g. the file system, remote servers, web services, currently connected devices). Thereby the default layout of visualizations and content in the information landscape is predefined by the UI designer. However, the concept of portals and direct manipulation enables the user to customize or create own layouts according to the user's typical information needs or her personal view of the PSI. As described in the previous section, important views or visual queries on the information space can be stored at prominent places for later use or subsets of information items can be dragged to arbitrary locations to arrange or annotate them freely on a canvas. This allows the user to design the appearance of the work environment, for example to create regions in the information landscape that can be used as visual equivalents to to-do lists or bookmarks supporting typical tasks in PIM.

Furthermore sketching, drawing or annotation tools can be used to create own visual arrangements and visual landmarks to facilitate information access and spatial orientation or to allow zoomable presentations of content within the information landscape. Thus the information landscape can also be used as a versatile direct manipulation authoring and annotation tool comparable to the multiscale editor "MuSE" of Furnas & Zhang [15]. In combination with pen-based devices like Tablet PCs or in environments with large displays this allows the use of a ZOIL-based user interface not only as a PIM tool, but also as a novel artistic or collaborative work environment.

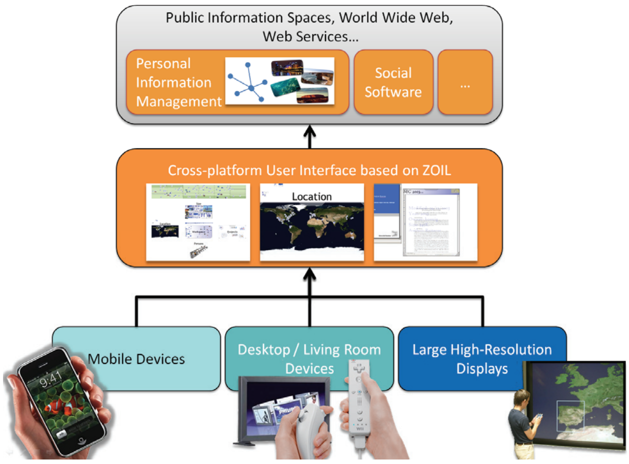

Figure 7. ZOIL as a cross-platform user interface paradigm deploying the personal information space to various devices.

ZOIL Design Principle 5: Nomadic Cross-platform User Interfaces

Due to the nature of ZUIs a ZOIL user interface scales to different display sizes and screen resolutions. This scalability facilitates the preservation of a consistent user interface and interaction style regardless of the employed device. ZOIL therefore leads to a significant improvement of the cross-platform consistency leading to high a degree of "horizontal usability" [34].

Figure 8. ZOIL-based user interfaces on a PDA, Tablet PC, Multitouch Display and a large high-resolution display.

In our vision of nomadic personal information management (see figure 7) the user can easily access her/his personal information from any device connected to the World Wide Web. Similar to current web-based email accounts the user simply logs into a web server which then provides the user's personal information in a familiar ZOIL-based rich internet application. This creates the illusion of a pervasive nomadic PIM environment which is not bound to specific operating systems or devices but is traveling with the user from device to device.

In course of our “permaedia” project [21] we will research how ZOIL-based user interfaces support this complementary use of such different devices like small display devices (e.g. Ultra Mobile PCs, PDAs or smart phones), desktop and living room devices (e.g. Tablet PCs, Home Theater PCs, set-top boxes) and large high resolution displays (see figure 8).

CONCLUSION & FUTURE WORK

With the ZOIL user interface paradigm we have outlined a novel approach to the design of integrated work environments for PIM. By formulating the five fundamental ZOIL design principles and illustrating them in mock-ups and prototypes we have made a first step towards an alternative to the traditional application-oriented desktop metaphor. We believe that ZOIL can evolve into the foundation of a new generation of cross-platform PIM user interfaces with high usability and an intriguing user experience. In particular the web-based deployment of ZOIL user interfaces as rich internet applications would open new possibilities for a pervasive and nomadic personal information management on a wide range of devices under a consistent interaction model.

For this reason we will develop provisional user interface libraries, input device managers and backend interfaces based on C# and .NET/WPF to allow a quick realization and evaluation of elaborated ZOIL-based prototypes for different stationary and mobile devices. The results from user tests will be used to optimize our designs in various contexts of use and to refine the formulation of the fundamental ZOIL design principles. In particular following important research questions remain unanswered at this stage and will be part of our future work:

How does the visual-spatial orientation in a ZUI stand the test of day-to-day use? The user’s orientation and navigation in ZUIs will be subject to empirical research and close examination as this ability is critical for the benefits of a ZUI and has not been researched sufficiently yet [16].

How can the ZOIL paradigm be enhanced by integrating peripheral views or polyfocal navigation? A user interface that is purely based on zooming and panning in the information landscape lacks permanently visible indicators for events like “an email has arrived” or lacks free floating regions of the screen for activities like instant messaging or video chats. Furthermore comparing items or transferring parts of one item into another should be facilitated by offering multiple simultaneous views on different locations in the landscape (e.g. by splitting the screen, by collapsing and expanding regions of the landscape or by using polyfocal distortion).

How can the ZOIL paradigm be enhanced through alerting? As Jones and Teevan point out in [22, p. 9] deferring processing until later is a frequent activity in PIM. To this day ZOIL does not contain possibilities to easily mark an information item for later processing and to keep track of these items, their changes or deadlines. Alerting the user about new arrivals, changes or at prescribed points in time should be integrated without impairing the consistency and visual character of the ZOIL UI.

Although the ZOIL paradigm is still at a very early stage, we are convinced of its potential for the future. In [28] Moran & Zhai have postulated seven dimensions in which future work environments should emancipate themselves from the traditional desktop metaphor. Regarding the new possibilities that the ZOIL paradigm introduces to user interface design we are convinced that ZOIL will be able to contribute innovative solutions in all these dimensions.

REFERENCES

1. Adler, S. WebOS: say goodbye to desktop applications. *netWorker* 9, 2005, 18-26.
2. Talks: Blaise Aguera y Arcas - Jaw-dropping Photosynth demo. http://www.ted.com/.
3. Ahlberg, C., Williamson, C. and Shneiderman, B. Dynamic queries for information exploration: an implementation and evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems,* ACM Press (1992), 619-626.
4. Ahlberg, C. and Shneiderman, B. Visual information seeking using the FilmFinder. In *CHI '94: Conference companion on Human factors in computing systems,* ACM Press (1994), 433-434.
5. Bardram, J.E. From Desktop Task Management to Ubiquitous Activity-Based Computing. In *Beyond the desktop metaphor : designing integrated digital work environments,* MIT Press (2007), Cambridge, Mass., 223-259.
6. Bederson, B.B. and Hollan, J.D. Pad++: a zooming graphical interface for exploring alternate interface physics. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology,* ACM Press (1994), 17-26.
7. Bederson, B.B., Shneiderman, B. and Wattenberg, M. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Trans. Graph 21*, ACM Press (2002), 833-854.
8. Bederson, B.B., Grosjean, J. and Meyer, J. Toolkit Design for Interactive Structured Graphics. *IEEE Trans. Softw. Eng 30*, IEEE (2004), 535-546.
9. Bier, E.A., Stone, M.C., Pier, K., Buxton, W. and Derose, T.D. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques,* ACM Press (1993), 73-80.

10. Collins, D. *Designing object-oriented user interfaces.* Benjamin Cummings, Redwood City, CA, USA, 1995.

11. Cutrell, E., Dumais, S.T. and Teevan, J. Searching to eliminate personal information management. *Communications of the ACM* 49, ACM Press (2006), 58-64.

12. Dittrich, J., Salles, M.A.V., Kossmann, D. and Blunschi, L. iMeMex: escapes from the personal information jungle. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases,* VLDB Endowment (2005), 1306-1309.

13. Donelson, W.C. Spatial management of information. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques,* ACM Press (1978), 203-209.

14. Fisher, D. and Nardi, B. Soylent and ContactMap: Tools for Constructing the Social Workscape. In *Beyond the desktop metaphor : designing integrated digital work environments,* MIT Press (2007), 171-190.

15. Furnas, G.W. and Zhang, X. MuSE: a multiscale editor. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology,* ACM Press (1998), 107-116.

16. Gerken, J. Orientierung und Navigation in zoombaren Benutzungsschnittstellen unter besonderer Berücksichtigung kognitions-psychologischer Erkenntnisse. Master Thesis, University of Konstanz, 2006.

17. Grün, C., Gerken, J., Jetter, H., König, W. and Reiterer, H. MedioVis - A User-Centred Library Metadata Browser. In *Proceedings of the 9th European Conference, ECDL, Research and Advanced Technology for Digital Libraries,* Springer (2005), 174-185.

18. Hutchins, E.L., Hollan, J.D. and Norman, D.A. Direct Manipulation Interfaces. In *User Centered System Design: New Perspectives on Human-Computer Interaction,* Erlbaum (1986), 87-124.

19. Jetter, H., Gerken, J., König, W., Grün, C. and Reiterer, H. HyperGrid - Accessing Complex Information Spaces. In *People and Computers XIX - The Bigger Picture, Proceedings of HCI 2005,* Springer, 2005.

20. Jetter, H. Informationsarchitektur und Informationsvisualisierung für die Post-WIMP Ära. Master Thesis, University of Konstanz, 2007.

21. Jetter, H. permaedia - Personal Nomadic Media for the Coming Decade. http://hci.uni-konstanz.de/permaedia.

22. Jones, W. and Teevan J. *Personal Information Management*. University of Washington Press, Seattle, WA, USA, 2007.

23. Kaptelinin, V. and Czerwinski, M.P. Beyond the desktop metaphor: designing integrated digital work environments. MIT Press, Cambridge, Mass, USA, 2007.

24. Karger, D.R. and Jones, W. Data unification in personal information management. *Communications of the ACM* 49, ACM Press (2006), 77-82.

25. Karger, D.R. Haystack: Per-User Information Environments Based on Semistructured Data. In *Beyond the desktop metaphor : designing integrated digital work environments,* MIT Press (2007), 49-99.

26. König, W.A. Referenzmodell und Machbarkeitsstudie für ein neues Zoomable User Interface Paradigma. Master Thesis, University of Konstanz, 2006.

27. Mandel, T. Windows vs. OS2, the GUI-OOUI war : the designer's guide to human-computer interfaces. Van Nostrand Reinhold, New York, USA, 1994.

28. Moran, T.P. and Zhai, S. Beyond the Desktop Metaphor in Seven Dimensions. In *Beyond the desktop metaphor : designing integrated digital work environments,* In *Beyond the desktop metaphor : designing integrated digital work environments,* MIT Press (2007), 335-354.

29. Müller-Prove, M. and Ludolph, F. Dueling Interaction Models of Personal-Computing and Web-Computing. In *Methodic and Didactic Challenges of the History of Informatics (MEDICHI),* Österreichische Computer Gesellschaft (2007), 34-38.

30. Paal, S., Kammüller, R. and Freisleben, B. A Cross-Platform Application Environment for Nomadic Desktop Computing. In *Net.ObjectDays; Lecture Notes in Computer Science,* Springer (2004), 185-200.

31. Perlin, K. and Fox, D. Pad: an alternative approach to the computer interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques,* ACM Press (1993), 57-64.

32. Raskin, J. The humane interface: new directions for designing interactive systems. Addison-Wesley, Reading, Mass., USA, 2000.

33. Rexhausen, S., Demarmels M., Jetter, H., Heilig, M., Gerken, J., Reiterer, H. Blockbuster – A Visual Explorer for Motion Picture Data. In *INFOVIS '07: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'07),* IEEE Computer Society (2007).

34. Seffah, A. and Javahery, H. Multiple user interfaces: Cross-platform applications and context-aware interfaces. In *Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces,* John Wiley & Sons (2004), 11-26.

35. Shneiderman, B. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces,* ACM Press (1997), 33-39.

36. The Deepamehta Community. DeepaMehta: Maintain your head, and share it, http://www.deepamehta.de