Re-framing the Desktop Interface Around the Activities of Knowledge Work

Stephen Voida
Department of Computer Science
University of Calgary
Calgary, AB, Canada T2N 1N4
svoida@ucalgary.ca

Elizabeth D. Mynatt, W. Keith Edwards
GVU Center, College of Computing
Georgia Institute of Technology
Atlanta, GA, USA 33032-0760
{mynatt, keith}@cc.gatech.edu

ABSTRACT
The venerable desktop metaphor is beginning to show signs of strain in supporting modern knowledge work. In this paper, we examine how the desktop metaphor can be re-framed, shifting the focus away from a low-level (and increasingly obsolete) focus on documents and applications to an interface based upon the creation of and interaction with manually declared, semantically meaningful activities. We begin by unpacking some of the foundational assumptions of desktop interface design, describe an activity-based model for organizing the desktop interface based on theories of cognition and observations of real-world practice, and identify a series of high-level system requirements for interfaces that use activity as their primary organizing principle. Based on these requirements, we present the novel interface design of the Giornata system, a prototype activity-based desktop interface, and share initial findings from a longitudinal deployment of the Giornata system in a real-world setting.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces—Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Activity-based computing, desktop computing, context-aware computing, knowledge work, Giornata

INTRODUCTION
The venerable desktop metaphor is beginning to show signs of strain in supporting modern knowledge work. In this paper, we examine how the desktop metaphor can be re-framed, shifting the focus away from a low-level (and increasingly obsolete) focus on documents and applications to an interface based upon the creation of and interaction with manually declared, semantically meaningful activities. We discuss how this class of activity-based desktop interfaces can provide a unified model for organizing work around activities, foster fluid multitasking, simplify resource organization, and incorporate collaboration capabilities into everyday tools.

UIST'08, October 19–22, 2008, Monterey, California, USA.

Our prototype system, Giornata, demonstrates how the traditional desktop metaphor can be re-framed to retain the spirit of simplified interaction with applications and files and yet better support contemporary knowledge workers' practices by emphasizing activity as the primary organizing principle in the interface. Giornata's enhanced desktop serves not only as a display space for application windows, but also serves as an active folder for documents and other information items associated with the current activity (Figure 1). Giornata utilizes lightweight activity- and document-tagging capabilities that enable informal and evolutionary resource organization. Finally, Giornata integrates collaboration tools directly into the desktop to support group information sharing and activity awareness.

In this paper, we make the following contributions:

- We describe an alternative model for organizing the desktop interface—activity-based computing—and identify a series of high-level system requirements for interfaces that use activity as their primary organizing principle.
- We present the novel interface design and implementation of the Giornata system, a prototype activity-based desktop interface.
- We discuss the technical issues involved in realizing Giornata and suggest ways that further research might foster the development of future activity-based systems.
- We share some initial findings from a longitudinal deployment of Giornata in a real-world setting.

To provide an overview of the design rationale and implementation of the Giornata system, we first discuss specific requirements for the design of Giornata based on the state of existing desktop interfaces, empirical studies of knowledge workers' actual practices, and theories of cognition grounded in the construct of activities. We then provide a scenario that depicts a holistic illustration of the system's support for knowledge work, and conclude with specific details about the interaction design and architecture of the prototype implementation.

THE DESKTOP INTERFACE
The desktop metaphor was developed over 30 years ago at Xerox PARC. The interaction techniques comprising the desktop interface responded to the needs of knowledge workers and the capabilities of computer technology in that era. These multi-window environments helped foster the multitasking practices that are now so central to modern knowledge work. The presence of a desktop "surface"

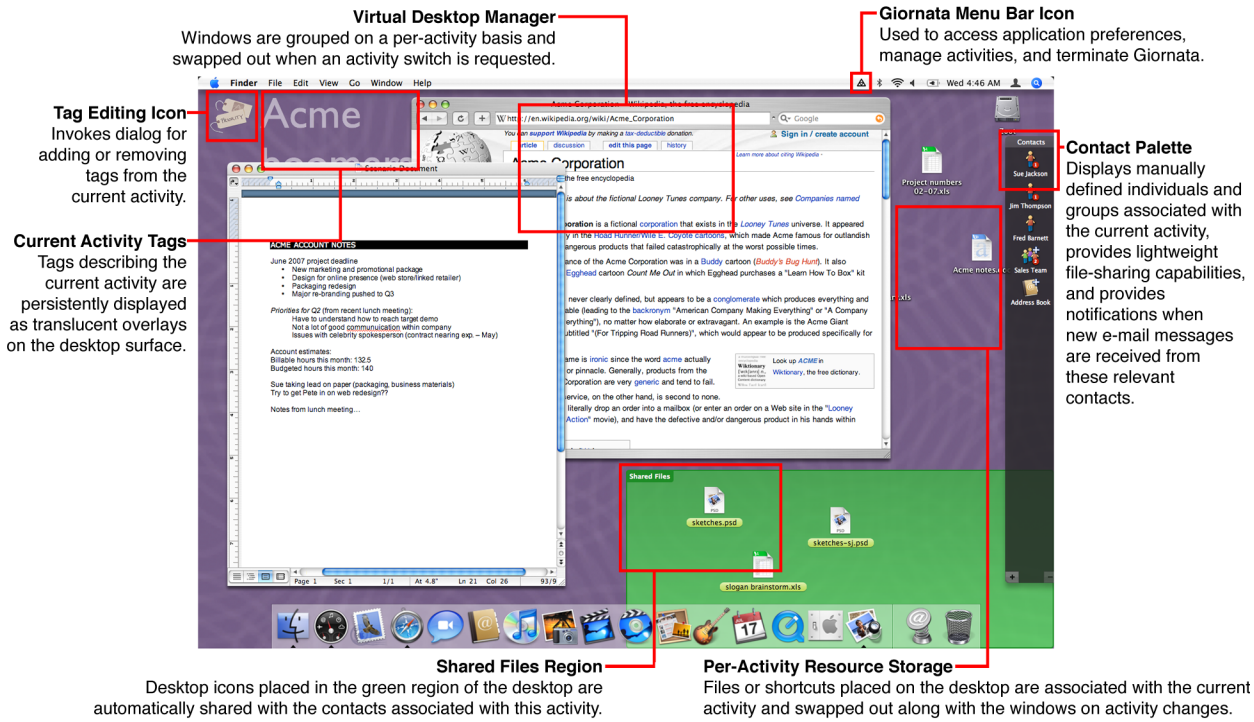

Figure 1. The Giornata interface. In this screenshot, an individual is engaged in managing a particular client's business account. There are several tags (including the client's name, "Acme"), two open windows, six files (three of them shared), three colleagues, and one group associated with this activity. During typical use, the Contact Palette automatically slides off-screen and application windows cover other Giornata interface elements until they are needed.

behind application windows also provided spatially oriented, persistent storage for icons representing files, application shortcuts, disk drives, and, eventually, the computer, itself.

As computers have grown more powerful and expectations about their capabilities have evolved, the desktop and the personal computing environment that it serves to ground have also evolved to enable new kinds of interactions. These changes can be broadly classified as new ways to manage space on the screen, new ways to manage stored information, and new tools to connect to other individuals.

One of the first major extensions to the desktop metaphor was the development of virtual desktops, exemplified in the Rooms system [12]. Rooms was based on a study of knowledge workers' task management practices and acknowledged that individuals tend to focus their interactions within semantically meaningful clusters of windows. This model was subsequently incorporated into the majority of X-windows window management tools. Other approaches to screen space management included space-filling tiled window techniques [14], grouping application windows ("pages") into manually-defined groups ("binders") that behave as a single window [4], grouping windows using existing window management tools like the Windows Taskbar [26], and even projecting the window environment into the third dimension [23]. Other space-related extensions include the incorporation of information awareness "widgets" alongside regular application windows (e.g., Apple's Dashboard and Microsoft's SideShow [6]).

Different models for information storage have also begun to disrupt the original model derived from information management on the physical desktop, which maps individual documents to individual files in the filesystem and each of these documents to a single window. Piles [19] and BumpTop [1] investigated grouping behaviors similar to those provided for windows via virtual desktops, but did so at the level of managing iconic representations of documents and applications where they are stored. Some information types—most prominently, e-mail, but also media files such as music and photos—are often not managed through the traditional desktop interface but are instead managed in separate information "silos" [5], stored separately from "traditional" documents and accessible only through a dedicated application, such as an e-mail client or a music "jukebox" application. The migration to more web-based storage and manipulation of documents is extending this distance between the desktop metaphor and individual documents; it is not uncommon to have a window be the *only* representation of a document available locally, with the file itself stored in a web-based repository.

Finally, the desktop metaphor was designed primarily for supporting a single individual; the intervening years have seen a dramatic increase in reliance upon collaboration-focused tools like e-mail and IM and much more pervasive use of remote servers to store all kinds of content. Most desktop interfaces provide relatively impoverished representations of these connected and collaborative resources. Attempts to create desktop-like collaboration interfaces (e.g., [25]) have demonstrated the potential in integrating collaborative functionality into systems at a

deeper level. However, despite their focus on desktop-like collaboration support, these tools are typically realized as stand-alone applications and do not integrate into existing desktop interfaces or more diverse work practices.

AN ALTERNATIVE TO THE TRADITIONAL DESKTOP METAPHOR: ACTIVITY-BASED COMPUTING

Rather than attempting to displace the existing desktop metaphor entirely, we posit that reframing the existing desktop metaphor around the higher-level construct of *activity* can address many of the limitations inherent in current desktop interfaces.

Cognitive theories can inform the computational representation of activities and help to define the design space for activity-based desktop interfaces. For example, activity theory models activity from the perspective of an individual (the *subject*) through three mutual relationships:

- the relationship between the subject and her objective (i.e., how she approaches, understands, and works toward satisfying the objective of the activity);
- the relationship between the subject and the surrounding community (i.e., how she interacts with others while working towards the objective); and
- the relationship between the community and the objective (i.e., what others do to help—or hinder—the subject's accomplishment of the objective) [31].

These relationships are *mediated* by other components of the activity (Figure 2), including the tools used or created in accomplishing the activity and the social structures dictating interaction within the larger community. Activity theory suggests the importance of encoding the tools (applications and resources) associated with the activity, the larger social context (individuals and groups, and perhaps the roles each play) in which the activity takes place, and some indication of the temporal evolution and connections among activities in activity-based systems.

However, adopting these kinds of structured representations of activity *within* an activity-based system does not necessarily imply that the interface representations of those activities need—or even should—be rigid or prescriptive. An alternative theory of cognition, situated action, posits that representations of activity serve different purposes at different times [27]. Flexibility at the moment of action is essential in enabling the activity to unfold, but structured records of the activity can serve as important

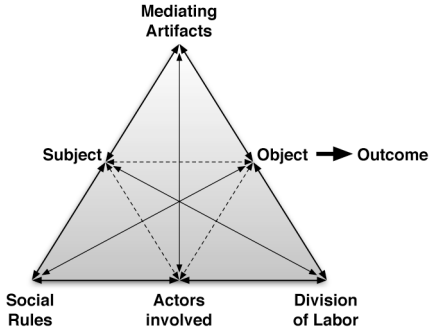

Figure 2. Engeström's visualization of the mediating relationships in activity theory (after [9]).

organizational, communicative, and collaborative artifacts. This theory suggests that activity-based tools need to allow for flexible specification and modification of activities, and that activity representations should persist after the activities themselves are completed, so they can be used as communicative tokens and templates for future activities.

These two theories provide the basis for our computational model of activity: a semantically defined cluster of tools (applications), information resources (documents) and social context (colleagues) that incorporates a history of use and can be flexibly appropriated on a moment-to-moment basis. In order to establish more detailed design guidelines for activity-based systems, we examine three challenges with which activity-based systems need to engage (adapted from [30]), grounding each in previous empirical research on knowledge work practices that highlight the function and structure of activity in day-to-day computer use.

Supporting Fluid Work Practice

Knowledge work is often associated with the practice of multitasking. At any point in time, knowledge workers—particularly managers—are involved in multiple, interwoven activities [2, 11]. These activities tend to exist in parallel, that is, "users rarely complete any time-consuming activity before beginning another task" [2].

First and foremost, activity-based tools need to support multitasking (or "multi-activity") behaviors and to avoid creating additional work for people to manage these activities electronically. Giornata uses as its starting point the virtual desktop metaphor popularized by the Rooms system [12]. As a result, Giornata inherits several requirements for supporting fluid work practice that focus on integrating activity management tools into the underlying desktop infrastructure and helping to ensure that the interfaces used to control the virtual desktop aspects of the system necessitate as little interaction overhead as possible during typical use of the system.

Requirement 1. To integrate into existing work practice, activity-based systems should provide a unified activity model across all applications, rather than being embedded in a single application.

Requirement 2. Activity-based systems should provide lightweight mechanisms to create, change, and alter activities, since heavyweight interaction techniques are likely to deter adoption and use.

Supporting Multifaceted and Evolving Activities

González and Mark's extensive studies of knowledge workers led them to identify activities (in their language, "working spheres") as being inherently multifaceted; that is, each activity

> shares a common motive (or goal), [involves] the communication or interaction with a particular constellation of people, uses unique resources and has its own individual time framework. With respect to tools, each working sphere might use different documents, reference materials, software, or hardware [11].

Their definition emphasized the interrelationships among the various components of an activity and suggested that

activity-based systems would need to incorporate sufficiently sophisticated activity models to represent these often-complex structures.

Activities also relate to the ways in which information is stored, organized, retrieved, and used. In her study of knowledge workers, Kidd noted that:

- knowledge workers rely little on filed information, taking prolific notes as part of the meaning-making process, but rarely revisiting them after the fact; and
- the spatial layout of a knowledge worker's materials is important as a "holding pattern" for short-term organizational purposes and before the materials have been classified and can be filed [16].

Malone's study of how knowledge workers organize their desks revealed a related distinction between *files* and *piles* in the office environment:

> [F]iles are units where the elements (e.g., individual folders) are explicitly titled and arranged in some systematic order (e.g., alphabetical or chronological)…In piles, on the other hand, the individual elements (papers, folders, etc.) are not necessarily titled [or]…arranged in any particular order [18].

Kidd and Malone both highlight the significance of information organization in the meaning-making process. Prior work in creating and studying the use of personal information management tools also resonates with this position (e.g., [13]). This research suggests that a combination of informal and formal mechanisms for storing information can help knowledge workers to organize information throughout the meaning-making process and that search-oriented and semantically meaningful retrieval techniques will likely be more useful than their browsing-oriented counterparts for working with previously filed information.

Requirement 3. Activity-based systems should provide tools for informally and formally organizing disparate information within activities. Informal information organization tools should emphasize quick storage and retrieval, without forcing people to explicitly name or find a permanent place for artifacts; formal mechanisms should correspond to long-term storage and retrieval practices.

Requirement 4. Real-world activities "overlap" in the way they use artifacts; a given artifact may be used in multiple contexts. Activity-based systems' representations of activity should support this overlap, rather than prescribing that activities be orthogonal or that their artifacts exist in only one context.

Requirement 5. Activity-based systems should allow post hoc definition of activities, enabling individuals to map their evolving understanding of the activities into the system; individuals should be able to create initially unnamed activities and then refine them after the fact. Artifacts used in unnamed activities may need to acquire these refined declarations of use as these activities evolve.

Supporting Collaboration Through Activities

Most knowledge work is inherently collaborative [3, 15, 28] and cognitive models of activity (e.g., activity theory) almost always take into account the social context within which work takes place [9, 31]. The information transformations most common in this class of work require discussion and cooperation among multiple stakeholders.

Even when collaboration isn't critical for a particular activity, that activity almost certainly draws upon information created by others at an earlier point in time or results in some deliverable that is then handed off to others [28]. However, most collaboration takes place within tools that do not distinguish among different work contexts. This suggests that activity-based systems should help people organize their communication and collaboration channels in ways that parallel the organization of their activities and, when possible, explicitly provide links between the two.

With Giornata, we focus on exploring the ways that activity management tools are adopted and appropriated in the context of everyday collaborations. Although our eventual goal is to support sharing entire activities, for the time being, we seek to understand some of the more fundamental issues in how the availability of activity representations and activity-based organizational tools affect the way that individuals manage their collaborations.

Requirement 6. Activities in activity-based systems should be usable as structuring mechanisms for collaboration (i.e., an activity-based perspective should be integrated into common collaborative tools).

Requirement 7. Because information sharing is a "common case" in knowledge work, lightweight sharing capabilities should be integrated directly as a first-class interaction technique.

INTERACTION DESIGN

Giornata[1] takes as its starting point the virtual desktop metaphor of the Rooms and Kimura systems [12, 17]. In addition to providing straightforward activity "spaces" into which focused work on single activities can be concentrated and their constituent components organized, Giornata provides a number of novel information organization and collaboration features.

Scenario of Giornata Use

Bob returns from a business lunch with representatives of Acme Inc. and logs into his computer. He switches to the activity tagged "Acme," which automatically populates his desktop with the files associated with the activity, restores the visibility and positioning of relevant open windows, and shuffles the contents of his Contact Palette to display his colleagues also working with the company. He opens a word processor document associated with the activity and jots down a few notes about the outcomes of the meeting.

A few minutes later, the e-mail icon in his Dock changes, indicating that two new e-mail messages have arrived. Bob resists the temptation to switch over to his e-mail client, suspecting that the new e-mails are unrelated to his current

[1] *Giornata* is Italian for "day's work," and, in the context of *buon fresco* (wet plaster) painting, denotes the area of a painting—the amount of work—that can be completed in a single session.

task and will distract him from finishing his notes. However, a moment later, the Contact Palette also updates to show that one of the messages is from Sue, a colleague working on the Acme project. He clicks Sue's icon and is taken to a filtered version of his inbox, displaying only messages sent by Sue. He reads Sue's latest e-mail and discovers that she is planning a meeting to discuss the progress on the Acme account. He quickly finishes working on his notes, saves the file back to the desktop, and then drags it into the shared region of his desktop so that Sue and his boss—both associated with the activity—can access the file through their corresponding activity workspaces.

Having completed the most pressing business, Bob opens his activity overview to take stock of what else needs to be accomplished. Seeing an activity tagged "home" and "renovations," Bob remembers that he had been asked to provide a recommendation for a contractor that worked on his house. Rather than closing the windows associated with the lunch meeting, he simply switches to the other activity. He begins to work on the letter when another colleague, Jim, drops by to determine when Bob is available to review an upcoming presentation. Bob uses a keyboard shortcut to quickly switch to the presentation activity, decides on a meeting time with Jim, and returns to work. Jim casually asks about Bob's letter, and suggests that Bob post his experiences to an local review website, "valleybook." Bob adds the tag "valleybook" to the activity (automatically tagging the file containing the letter) as a reminder to post the finished recommendation online.

Activity-Based Multitasking

In Giornata, each activity is associated with a corresponding virtual desktop. In order to support fluid—and often fast-paced—work, the system enables creation of a new, empty, untagged activity using a single keystroke (*per requirements 1 and 2*). This action hides all on-screen windows and desktop contents, presenting a clean canvas on which work can begin on a new activity without distraction or the need to manually manage digital clutter.

Giornata allows an individual to navigate among open activities using a status bar menu, accelerator keys, or a quick activity switcher (Figure 3), which operates using the same interface principle as the application switching service available both in Windows (invoked using alt + tab) and the OS X operating system (via command + tab).

Although we acknowledge the need for incorporating advanced activity management tools into activity-based interfaces (e.g., support for resuming prior activities, use of "activity templates" to streamline repeated tasks, and tools for merging or splitting running activities), we did not include these capabilities in the first iteration of Giornata since their availability was not essential for observing how this class of systems would be adopted in actual use.

Several recent research systems have explored enhancing the desktop to support multitasking practices. Activity-Based Computing [3], TaskTracer [8], and Kimura [17] all extended the virtual desktop ideas espoused in Rooms [12], focusing respectively on supporting mobile work across

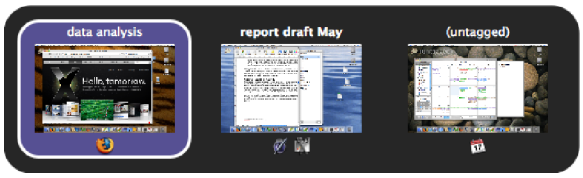

Figure 3. The quick activity switcher interface. The text across the top indicates each of the activities' tags and the icons below each thumbnail represent the applications associated with each activity.

multiple devices, informing the development of machine learning algorithms for automatically classifying activities, and using peripheral displays to provide activity awareness. GroupBar [26] and Scalable Fabric [24] explored the benefits of integrating window grouping and "focus + context" window management into the existing Windows desktop interface. Task Gallery [23] extended the desktop into the third dimension, allowing activities to be clustered and manipulated in a spatially rich environment.

Giornata distinguishes itself from previous virtual desktop management systems in three important ways. First, the number of virtual desktops available in Giornata is not fixed as it is in many virtual desktop implementations; individuals have exactly the number of virtual desktops at their disposal as they have ongoing activities. This prevents unnecessary overloading of virtual desktops and is intended to speed transitions among them (*requirements 1 and 2*). Second, the objects stored on the desktop and the contacts in the Contact Palette transition in and out along with the associated windows. This serves to provide a dedicated storage space associated with the activity and helps to ensure that activities are perceived as cohesive units, including tools, artifacts, and contacts (*requirement 3*). Finally, Giornata allows—but does not require—activities to be tagged for quick identification (*requirement 5*).

Activity-Based Resource Storage

In Giornata, the desktop serves not only as a display space for application windows, but also as a readily accessible folder for documents and shortcuts associated with the current activity. Any file saved or copied to the desktop is automatically associated with the current activity; as an individual switches among ongoing activities, these resources are "swapped out" along with application windows and temporarily stored in a folder associated with the activity until the activity is resumed. The effect of this feature is that the desktop workspace is automatically repopulated with the files, folders, and other information resources associated with each activity as an individual's focus changes (*requirement 3*). This behavior is similar to the approaches taken by Lifestreams [10], Time-Machine Computing [22], and the Context Browser [21], with the main difference being the underlying organizing principle determining the visibility of the desktop's contents, ours being activity instead of time.

These capabilities filter the information displayed on the screen at any time to the most relevant applications, information resources, contacts, and communications (*requirements 1 and 3*).

Activity Tagging

Each activity in Giornata can be annotated with optional, freeform tags to describe its semantics. Activities are initially created without tags; the ability to create and work in an unnamed desktop allows work to proceed even when an individual might not know the significance or eventual meaning of an activity at its outset (*requirement 5*).

An activity's tags help individuals identify the activity in which they are currently working and distinguish among background activities. The active activity's tags are persistently visible, rendered over the desktop wallpaper; they can also optionally be displayed in the menu bar.

When an activity has one or more tags associated with it, these tags are transferred to each file touched over the course of working in that activity. This design serves to "stamp" files with information about the context in which they were created or edited, and helps to overcome the burdensome process of manually adding semantic metadata to each individual file associated with an activity, an approach similar to that taken by Dourish et al. [7]. It also allows documents that are shared across multiple activities to be stored elsewhere in the filesystem and still "inherit" tags from all activities in which they are used. Because the Spotlight framework automatically indexes these tags, individuals can find information resources using the semantically meaningful tags they assigned to the activity, regardless of where the files associated with the activity are actually stored on the disk (*requirements 3 and 4*).

As an individual comes to understand the meaning of a particular activity, she can edit the activity's tags by clicking on a tag icon on the desktop surface. She is then given the option to tag the activity's files from that point forward or to retroactively tag all of the files previously associated with the activity as well. This ability to create post hoc tags on activities and files enables individuals to refine the meaning of an activity as that meaning emerges or changes over the course of the work. It also helps to ensure that the system's activity representations are sufficiently flexible to adapt to the individual's evolving work environment (*requirements 2 and 5*).

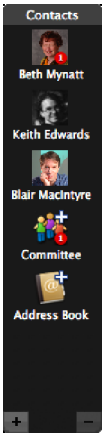

Figure 4. Giornata's Contact Palette. In this case, the icons represent three colleagues and one ad hoc group. The numbered, red "badges" indicate that the e-mail inbox contains one unread e-mail from the contact "Beth Mynatt" and one from a member of the "Committee" group. The "Address Book" icon reveals contacts from the OS X Address Book, allowing drag-and-drop association of contacts with the activity.

Activity-Aware Collaboration Support

Giornata provides two features to support activity-aware collaboration. First, Giornata integrates a subset of the Sharing Palette interface to enable lightweight collaboration [29]. This "Contact Palette" component, attached to one side of the display space, provides a persistent visual summary of the colleagues and groups an individual has associated with the current activity (Figure 4). Like open windows and files stored on the desktop, contacts are swapped out when transitions among activities are invoked, providing a persistent display of contacts salient to the current task (*requirement 6*).

The Contact Palette provides several awareness and collaboration services. The system periodically connects to a specified e-mail client and annotates the icons in the palette with "badges" indicating the number of unread e-mails in the inbox originating from each person or group, providing a summary of unread e-mails potentially relevant to the current work context (in contrast to the overall number of unread messages, which can have little bearing on the activity at hand). A contact's icon can also be clicked to reveal a variety of information about the contact; in the current implementation, options include displaying the Address Book card for the contact or a filtered view within an e-mail client, showing only messages sent by the selected contact. Finally, the Contact Palette can be used to share files with the individuals who are associated with particular activity using the same interaction design used in the Sharing Palette [29]. Files can be dropped onto the Contact Palette to share a file with a particular contact or group (*requirement 7*).

Giornata's desktop also includes a "shared files" region, which provides a persistent, spatial connection among collaborators' activity desktops. When files are dragged into this region, they are automatically replicated on each of the collaborators' desktops and updated each time the files' contents are changed[2]. This region acts as an information-sharing portal across all collaborators' desktops, but also allows all participants in an activity-based collaboration to control, as is contextually appropriate, the degree to which information is shared (*requirements 6 and 7*).

Several prior systems have attempted to use activity as a means for fostering collaboration, including UMEA [15] and ActivityExplorer [20]. However, the main distinction

[2] Currently, a naïve replication strategy is used to enable the shared files region: dragging a file into the highlighted area causes a read-only copy of the document to be created on collaborators' desktops, distributed and kept up-to-date using a peer-to-peer networking protocol. Future work includes investigating options to enable more flexible document sharing.

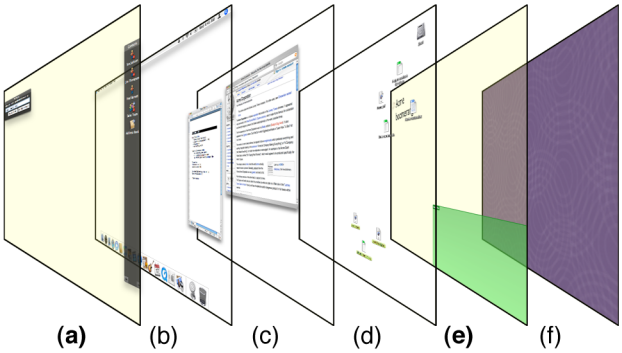

Figure 5. Explicit and implicit interaction layers in the Giornata system, and their relationship to existing window manager interaction layers. This figure illustrates the interaction layers of Figure 1: (a) Giornata's explicit interaction layer, including activity management dialogs and the Contact Palette; (b) the system menu and Dock; (c) application windows; (d) desktop icons; (e) Giornata's implicit interaction layer, including activity tag display and sharing space; and (f) the desktop wallpaper.

between these systems and Giornata is that Giornata integrates both the activity representations and the collaboration tools into the desktop interface itself; the others rely on interaction within a standalone application.

Support for Implicit and Explicit Interactions

Giornata's interface integrates closely with the existing file and window management components of Apple OS X (*requirement 1*). The OS X window manager emulates the physical manipulation of paper on a desk by compositing application windows on various layers above the desktop file icons and wallpaper, but below system-wide interaction widgets like the menu bar and Dock (Figure 5). Giornata augments this visual stack by inserting two additional layers: an explicit interaction layer on top of all other layers (Figure 5a), providing persistent visibility of the Contact Palette and allowing individuals to control the activity management system, and an implicit interaction layer below the desktop file icons but above the background wallpaper (Figure 5e). This non-interactive layer serves as a persistent information display for information such as the current activity tags. It also passively monitors interactions with existing desktop objects (such as desktop file icons), providing the system with input as a side effect of other, typical desktop interactions.

The implicit interaction layer is a particularly powerful component of the Giornata interface design. Because it serves as a persistent information display and is "anchored" to the desktop wallpaper and rendered translucently, a quick overview of the activity state can be quickly surmised by invoking the "show desktop" feature of Exposé. The seamless augmentation of the desktop background also helps to convey Giornata's status as an integral part of the desktop environment. It also serves to reduce visual clutter, as Giornata's interface elements are typically hidden behind application windows until needed.

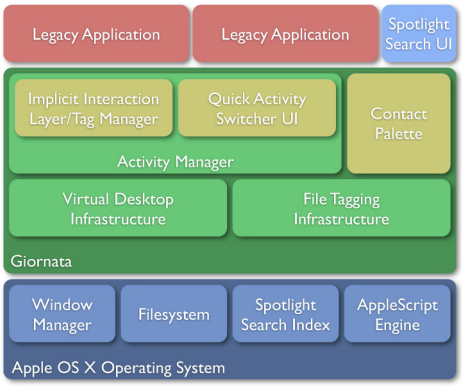

Figure 6. A high-level overview of Giornata's architecture. The prototype builds on several core system services and presents itself as an integrated component of the operating system, enabling representations of activity within existing applications without requiring that these applications be modified.

Additionally, the implicit interaction layer, together with a filesystem change-monitoring daemon, serves to manage the public/private sharing status of desktop items based solely on their location on the desktop. An individual can indicate that a file associated with an activity is to be shared freely with relevant colleagues by adding those colleagues to the Contact Palette and moving the file to a "shared files" region on the desktop. Giornata automatically notifies the colleagues whenever files are added to this region or existing files in that region are changed. Likewise, dragging the file icon out of the shared file region and dropping it elsewhere on the desktop suspends further automatic sharing of the file.

SYSTEM ARCHITECTURE

Giornata is implemented on OS X as a hybrid Carbon-, Cocoa-, and AppleScript-based application. The application is designed to run continually while an individual is logged in and provide activity-management services alongside and independently of other system applications.

Although Giornata is technically just another application running on the system, it is designed to integrate as closely as possible into the fabric of the underlying operating system. This integration is accomplished in part by building upon high-level operating system services, which ensures that activity-related actions within the system are immediately reflected in other applications and the operating system, itself (Figure 6). This design emphasizes the role of activities while allowing an individual to run existing applications alongside Giornata without penalty or modification.

In the following sections, we describe Giornata's implementation details, focusing on our experiences in constructing an activity-based system suitable for real-world use on top of a commercially-available, mainstream operating system, the ways that our system's modules leverage lower-level system services and APIs to accomplish the goals of the system's interaction design, and persistent shortcomings of existing operating systems that create barriers for activity-based system design.

Virtual Desktop Infrastructure

Giornata's virtual desktop code is based on an open-source virtual desktop application named VirtueDesktops[3]. In order to implement the core virtual desktop functionality, VirtueDesktops and Giornata use an existing-but-undocumented API supporting multiple virtual workspaces in Apple OS X, recently (officially) utilized by Apple's Spaces[4]. This API provides functions to determine the current virtual workspace, perform an optional, animated transition between workspaces, and get and set the virtual workspace on which a particular window appears.

However, some of the capabilities needed to implement a fully functional virtual desktop system (e.g., the ability to move windows from one workspace to another) are available only when executed in the same process context as the core window server. In order for these calls to succeed, Giornata takes advantage of a feature of the Mach kernel known as *code injection*. Using this technique, the code of a running process—in this case, the Dock application, which also serves as the host for the window server—is dynamically modified to relay information and commands between the main Giornata application and the window server. Another open-source library, *mach_inject*[5], enabled the use of this code injection technique in Giornata.

File Tagging and Implicit Interaction Infrastructure

Giornata's tag manager is implemented as an Objective-C category extending Cocoa's NSFileManager class and provides additional functions for converting between activity tags and comment strings and for setting and retrieving Spotlight Comments for specified files via AppleScript. Activity tags used to annotate a file are each prefaced with an "@" character and appended to any existing contents in the Spotlight Comments field using a space character as a tag delimiter. This encoding scheme is computationally straightforward, ensuring that the system can quickly read or write tags for a large number of files without incurring significant overhead. It also provides a human-readable representation of the tags that can be viewed or edited using the Finder or used as search keywords in Spotlight.

When Giornata starts up, it launches a file-monitoring daemon to observe filesystem changes and automatically apply tags to files that are "touched." This process, running with root-level privileges, takes advantage of the *fsevents* kernel-level filesystem monitoring facility typically used by Spotlight to detect when files are created or changed so they can be indexed for rapid search. This approach ensures that Giornata "sees" any work taking place in the filesystem and allows the system to automatically tag changed files with semantically meaningful metadata without incurring any additional interaction costs.

When the daemon detects that the desktop database file has been modified, indicating that items have been added to, removed from, or moved to a different location on the desktop, it sends a notification to the main Giornata application that an implicit input action has taken place. When this notification is received, the main Giornata application examines each of the items on the desktop using an AppleScript to determine if their desktop positions fall within the boundaries of the sharing space. When an item is found to be within this space, Giornata turns on the item's Finder highlighting (as a confirmation that the system has recognized and begun sharing the item) and adds the file to the list of shared files for the activity.

The implicit interaction layer is also responsible for maintaining per-activity desktop file storage. When an activity switch is requested, the (X, Y) position of each file on the desktop is captured using an AppleScript and then the current contents of the desktop are moved to a storage folder associated with the activity, typically in the folder named "/Users/*username*/Activities/*activity tags*". Once the desktop has been cleared, the desktop contents of the incoming activity are restored and each item is manually re-positioned at its previous location on the desktop.

Contact Palette

The Contact Palette operates as a semi-autonomous component of the Giornata system, providing a persistent visual representation of individuals and groups associated with the current activity using semi-transparent, HUD-style windows and arrays of custom, icon-centric widgets to mimic the interface of the Sharing Palette [29].

The Contact Palette maintains awareness about the current activity by registering for distributed notifications from the activity manager component. When an activity transition is requested, the palette synchronizes its contents with the centralized activity model, updating its display to show the contacts associated with the destination activity. When the Contact Palette receives a notification from the implicit interaction infrastructure that a file has been moved into or out of the sharing space on the desktop, the palette passes the full path of the shared file and the e-mail addresses of the contacts currently associated with the activity to a peer-to-peer file sharing library originally developed for the Sharing Palette prototype, which manages replicating the files across the network initially and whenever modified.

Additionally, the Contact Palette serves as one of the key bridges between Giornata and the surrounding desktop ecosystem, connecting to a number of external services and applications. The palette uses the OS X Address Book framework to dynamically update the list of contacts that can be associated with an activity; all contacts in the Address Book database with an e-mail address are automatically added to the "Address Book" group within the palette. The palette also connects to the specified e-mail client periodically via AppleScript, retrieving a list of unread messages and updating the palette's individual and group icons with badges to provide awareness of electronic communications most relevant to the current activity.

[3] http://virtuedesktops.info

[4] http://www.apple.com/macosx/features/spaces.html

[5] http://rentzsch.com/mach_inject

Persistent Technical Challenges

A number of technologies enabled the realization of the Giornata system with sufficient robustness to support a long-term deployment: a handle-based and metadata-focused filesystem (handle-based file referencing allowed Giornata to move files in and out of the desktop folder without breaking existing applications); widespread availability of application scripting tools (to increase the degree of integration with the operating system and existing tools); and increased operating system stability and multiprocessing capabilities (to support a greater number of concurrent activities over a longer time).

However, a number of significant technical challenges remained in developing Giornata; these challenges are likely to frustrate other future efforts to develop and deploy activity-based systems. First, the fragmentation of commonly used information resources—first and foremost, e-mail—make it difficult for activity representations to adequately capture both local work products and communicative interactions. Second, developing innovative interaction techniques for managing open windows is challenging since the window manager in most mainstream operating systems is closed-source and offers limited extensibility. (X11-based systems are one exception, although prototyping for Linux or UNIX can limit potential opportunities for studying real system use.) Finally, activity-based systems still have no way to reliably restore the internal state of running applications following a system crash or reboot (also noted previously by Robertson et al. [23]). Solutions to this problem might include the development of new programming frameworks requiring developers to provide hooks for acquiring a snapshot of an application's state and restoring it, or, perhaps, the use of explicit resource virtualization on a per-application level.

DEPLOYMENT AND STUDY

Instead of evaluating Giornata's interface in a controlled, laboratory setting, we believed that it was essential to deploy and study a fully-implemented system to be used in the context of real-world work for two reasons: we wanted to investigate how participants organize their own, familiar resources and collaborations when an activity-based tool is available, and we felt it critical to observe how long-term use of the system would reveal emergent strategies for organizing knowledge work within activities.

We deployed the Giornata prototype to five participants (two university faculty members, two graduate students, and one industrial HCI practitioner), who used the system as part of their everyday work for an average of 54 days (min = 22 days; max = 82 days). For the deployment, we instrumented Giornata to log information about all activity-based interactions. At the conclusion of the deployment, we asked participants to rate the usefulness of several aspects of the system and conducted semi-structured interviews with each of the participants to elicit specific feedback about their experiences using the software.

Although an in-depth discussion of participants' use of the system is beyond the scope of this paper, we present results highlighting participants' reactions to several facets of Giornata's activity-oriented interface:

- Participants reported having generally positive experiences using the system, rating its usefulness an average of 4.1 (SD 0.6) on a 5-point Likert scale.
- Participants logged substantial real-world use of the system, with an average of 7.6 open activities per participant over the course of the study (SD 3.5). The software logs also revealed that the participants engaged in an average of 28.2 activity switches per day (SD 15.9). None of the participants reported problems with the system scaling to meet their needs over the duration of the deployment.
- The system supported a wide variety of working styles. Some participants maintained longer lists of fine-grained activities, while others created only a few, high-level activities. Some participants concentrated their personal information management tools in just one activity, while others intentionally replicated aspects of these tools across nearly all of their activities.
- When participants did not know how to name an activity or needed a temporary place to work, Giornata did not get in their way: 14.8% of all activities created during the study remained untagged for the entire study.
- Two of the five participants had used other virtual desktop software in the past. One commented that keeping his desktops appropriately partitioned had been a challenge when using these systems and noted "that can still happen with Giornata, but I think by…binding specific activities to specific [desktops] has helped with that…. It may be that there's not that fixed layout…."
- The per-activity resource storage was frequently cited as one of the "biggest wins" in using the system. All of the participants used this feature (to varying degrees), and most commented that having a place to store files without having to negotiate the hierarchical filesystem was valuable. One participant noted that routinely saving files to the desktop "feels better than filing."

CONCLUSION

Giornata demonstrates that re-examining the assumptions underlying the design of the desktop interface can lead to interfaces that more closely match real-world work practices than systems based on a traditional, increasingly dated application- or document-based interaction metaphor. We anticipate that activity-based systems will provide a variety of benefits, including better task awareness, simpler multitasking, more natural organization of information, and improved collaboration.

In this paper, we presented seven requirements, grounded in cognitive theory and observations of real-world work, for foregrounding representations of activity as the primary organizing principle in the desktop interface. Building on these requirements, we presented the Giornata interface, one instantiation of an activity-based system that enhances the OS X desktop in a number of key areas:

- Giornata *supports fluid work practice* by explicitly supporting multitasking, closely integrating with existing features of the desktop environment and

allowing individuals to quickly create and switch among activities;

- Giornata's activity representations *encapsulate evolving work* based on the integration of per-activity persistent storage of relevant files and of a lightweight means for incrementally labeling activities and their constituent resources using semantically meaningful tags; and
- Giornata *supports collaboration* through the inclusion of individuals and groups as first-class objects in activity representations, providing collaboration awareness and a lightweight and powerful mechanism for sharing files and maintaining ongoing collaborations grounded in a set of shared information artifacts.

ACKNOWLEDGEMENTS

We would like to thank Gregory Abowd, Blair MacIntyre, and Tom Moran for their feedback on the design of the Giornata system and the presentation of this research. This work was supported by Steelcase Inc. and the GVU Center.

REFERENCES

1. Agarwala, A. and Balakrishnan, R. Keepin' it real: Pushing the desktop metaphor with physics, piles, and the pen. In *Proc. CHI '06*, ACM Press (2006), 1283–1292.
2. Bannon, L., Cypher, A., Greenspan, S. and Monty, M.L. Evaluation and analysis of users' activity organization. In *Proc. CHI '83*, ACM Press (1983), 54–57.
3. Bardram, J.E. Activity-based computing: Support for mobility and collaboration in ubiquitous computing. *Personal and Ubiquitous Computing 9*, 5 (September 2005), 312–322.
4. Beaudouin-Lafon, M. and Lassen, H.M. The architecture and implementation of CPN2000, a post-WIMP graphical application. In *Proc. UIST 2000*, ACM Press (2000), 181–190.
5. Bergman, O., Beyth-Maron, R. and Nachmias, R. The project fragmentation problem in personal information management. In *Proc. CHI '06*, ACM Press (2006), 271–274.
6. Cadiz, J.J., Venolia, G., Jancke, G. and Gupta, A. Designing and deploying an information awareness interface. In *Proc. CSCW '02*, ACM Press (2002), 314–323.
7. Dourish, P., Edwards, W.K., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry, D.B. and Thornton, J. Extending document management systems with user-specific active properties. *ACM Transactions on Information Systems 18*, 2 (April 2000), 140–170.
8. Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L. and Herlocker, J.L. Tasktracer: A desktop environment to support multi-tasking knowledge workers. In *Proc. IUI '05*, ACM Press (2005), 75–82.
9. Engeström, Y. *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research*. Orienta-Konsultit Oy, Helsinki, Finland, 1987.
10. Freeman, E. & Fertig, S. Lifestreams: Organizing your electronic life. In *Proc. AAAI Fall Symposium (FS-95-03)*, AAAI Press (1995).
11. González, V.M. and Mark, G. "Constant, constant multi-tasking craziness": Managing multiple working spheres. In *Proc. CHI '04*, ACM Press (2004), 113–120.
12. Henderson, J.D.A. and Card, S.K. Rooms: The use of multiple virtual workspaces to reduce space contention in window-based graphical user interfaces. *ACM Transactions on Graphics 5*, 3 (July 1986), 211–241.
13. Jones, W., Klasnja, P., Civan, A. and Adcock, M.L. The personal project planner: Planning to organize personal information. In *Proc. CHI '08*, ACM Press (2008), 681–684.
14. Kandogan, E. & Schneiderman, B. Elastic windows: Evaluation of multi-window operations. In *Proc. CHI '97*, ACM Press (1997), 250–257.
15. Kaptelinin, V. UMEA: Translating interaction histories into project contexts. In *Proc. CHI '03*, ACM Press (2003), 353–360.
16. Kidd, A. The marks are on the knowledge worker. In *Proc. CHI '94*, ACM Press (1994), 186–191.
17. MacIntyre, B., Mynatt, E.D., Voida, S., Hansen, K.M., Tullio, J. and Corso, G.M. Support for multitasking and background awareness using interactive peripheral displays. In *Proc. UIST '01*, ACM Press (2001), 41–50.
18. Malone, T.W. How do people organize their desks? Implications for the design of office information systems. *ACM Transactions on Office Information Systems 1*, 1 (January 1983), 99–112.
19. Mander, R., Salomon, G. and Wong, Y.Y. A 'pile' metaphor for supporting casual organization of information. In *Proc. CHI '92*, ACM Press (1992), 627–634.
20. Muller, M.J., Geyer, W., Brownholtz, B., Wilcox, E. and Millen, D.R. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proc. CHI '04*, ACM Press (2004), 375–382.
21. Park, Y. and Furuta, R. Keeping narratives of a desktop to enhance continuity of on-going tasks. In *Proc. JCDL 2008*, ACM Press (2008), 393–396.
22. Rekimoto, J. Time-machine computing: A time-centric approach for the information environment. In *Proc. UIST '99*, ACM Press (1999), 45–54.
23. Robertson, G., van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D. and Gorokhovsky, V. The Task Gallery: A 3D window manager. In *Proc. CHI 2000*, ACM Press (2000), 494–501.
24. Robertson, G., Horvitz, E., Czerwinski, M., Baudisch, P., Hutchings, D., Meyers, B., Robbins, D. and Smith, G. Scalable Fabric: Flexible task management. In *Proc. AVI '04*, ACM Press (2004), 85–89.
25. Roseman, M. and Greenberg, S. TeamRooms: Network places for collaboration. In *Proc. CSCW '96*, ACM Press (1996), 325–333.
26. Smith, G., Baudisch, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D. and Andrews, D. GroupBar: The TaskBar evolved. In *Proc. OZCHI 2003*, University of Queensland, Brisbane, Australia (2003), 34–43.
27. Suchman, L. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, UK, 1987.
28. Tang, J.C., Drews, C., Smith, M., Wu, F., Sue, A. and Lau, T. Exploring patterns of social commonality among file directories at work. In *Proc. CHI '07*, ACM Press (2007), 951–960.
29. Voida, S., Edwards, W.K., Newman, M.W., Grinter, R.E. and Ducheneaut, N. Share and share alike: Exploring the user interface affordances of file sharing. In *Proc. CHI '06*, ACM Press (2006), 221–230.
30. Voida, S., Mynatt, E.D. and MacIntyre, B. Supporting activity in desktop and ubiquitous computing. In Kaptelinin, V. and Czerwinski, M. (eds.), *Beyond the Desktop Metaphor: Designing Integrated Digital Work Environments*. MIT Press, Cambridge, MA, 2007.
31. Vygotsky, L.S. & Cole, M. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978.