

# Managing Activities with TV-ACTA: TaskVista and Activity-Centered Task Assistant

Victoria Bellotti

Jim Thornton

Palo Alto Research Center  
Palo Alto, CA 94304, USA

bellotti@parc.com; jthornton@parc.com

## ABSTRACT

We have developed a prototype to-do list called TaskVista that enables a user to create to-do items by typing or by drag-and-drop of files or email messages. TaskVista allows the user to promote items into Activities in our Activity-Centered Task Assistant (ACTA). Activities are pre-structured containers that are created inside the email folder hierarchy to support PIM. Specialized subfolders called Components within each ACTA Activity automatically organize and present information appropriately for aspects of the activity at hand. ACTA is designed to create a more efficient PIM environment with the ultimate goal of providing context metadata for machine learning and automation techniques.

## Keywords

Task management, PIM, context-aware computing.

## INTRODUCTION

Task management and personal-information management (TM and PIM) are significant challenges in knowledge work [2]. Complex, interdependent tasks, requiring people to monitor and wait for the work of others, seem to be a major source of the sense of overload that many feel [2]. For this reason we focus on supporting complex interdependent tasks in our system design efforts to reduce the burdens of task management.

We have designed a PIM construct we call an *Activity* to provide the user with a new organizational tool for managing complex tasks that also will facilitate the use of machine learning techniques to infer what the user is doing and provide support. The Activity construct captures context that is currently missing in folder-based information management environments such as Windows or OSX. Before describing Activities more fully, we review some of the prior work in this area.

## RELATED WORK

Recent years have seen a growth in research that attempts to use algorithms for data and lexical analysis, analytical modeling techniques and machine learning to reduce the burdens of task management. This work can be loosely characterized as a desktop-based contextual computing approach to TM and PIM and includes the following:

Firstly, some researchers have focused on reducing the overheads of filing incoming email for later retrieval and re-use using automated methods [4,15,16].

Secondly, others have used computation to anticipate users' goals and provide mechanisms for system initiative in office tasks such as calendaring. [9,13].

Thirdly, some attempt to identify threads of task activity from analyzing the contents of emails and documents that people are working on [5,14].

The approaches cited above all adopt a largely bottom-up approach to inferring task-related properties of human action related to items encountered, created and used. We do not dispute the value of this approach, but we propose that it can be made more powerful with activity-based UIs for TM and PIM. In theory, UIs designed to reflect core, stable properties of common activities can reduce ambiguity in interpreting user behavior. Using the kinds of automated methods described above, the system can bring a model of the activity context to bear upon its analysis of what transpires in interaction. Thus our work is intended to be complementary to the approaches mentioned above.

We also contrast our work with content grouping tools [11,17]. While this technique lets users gather together related content and resources, it provides no system semantics to characterize the user's different activities.

## ACTIVITIES FOR CONTEXTUALIZING INTERACTION

Our Activities are system schemas for common activities such as organizing a meeting, preparing a presentation, or conducting a performance review. To the user, an Activity is a task-specific collection containing structured elements called *Components* that embody common resources of a complex task. Just as a user might have different folder instances for different projects, she would have different Activity instances for different meetings, for example. However, unlike folders, Activity instances are of different types, created from various pre-designed templates.

We depend upon the fact that the majority of people file useful electronic items for future use [3,8] and that people reuse email either in the inbox or stored in folders to support prospective tasks [1,10], although this is not well supported by current email clients [2,10,20].

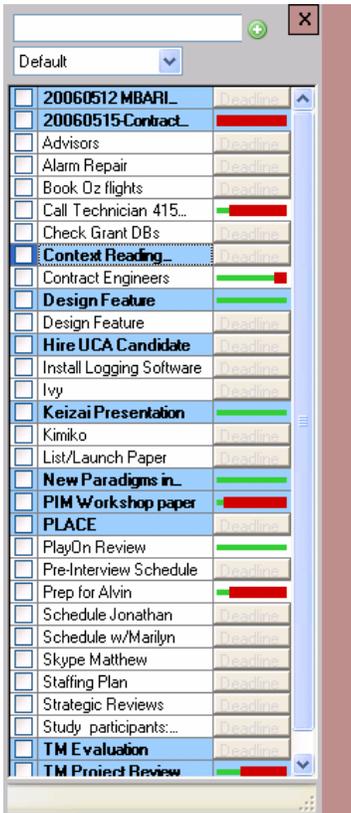


Figure 1. TaskVista to-do

When users create folders they imbue them with personal meaning [12]. Every time they drag an item into a folder or some other PIM or TM resource, the cognitive and manual effort they invest in characterizing the item and dragging it somewhere is wasted from a system perspective. The system has no way of capturing meaning. This problem is solved in a related system, TaskTracer, [7,19] by having the user enter a descriptive label for the current task they are working on as they manipulate files. This label (while active) is attached to all the items that are touched

until the user enters a new term. The idea is to tag items to allow the user to easily find all task-related content, no matter where it is. However, this is at the expense of users having to remember to label items.

By contrast, our trick is to have the user implicitly inform the system of what they are doing whenever they drag an item to a specific Component in an Activity. In this way their decision-making and filing effort fleshes out the system schema that can be used to reason about the information items. This enables the system to “understand” the content as related to a user activity in a particular way with no user labeling step.

**DESIGN**

In this section we briefly review the design of TaskVista and ACTA (Activity-Centered Task Assistant).

**TaskVista**

TaskVista (Figure 1) is a to-do list that resides on the desktop. It can be kept out of view by clicking on a tab which remains visible for easy access. To-dos can be sorted in various ways, most interestingly, by deadline, which is indicated by warning-bars which turn from green to red as the deadline approaches within two weeks.

Users create to-dos by typing into the field at the top of the list, and they can also create a to-do by dragging a file item from anywhere or a message out of email (in which case the name of the to-do becomes the filename or the message subject, though it can be edited).

If the to-do becomes more complicated, involving more items, the user can add these to the existing to-do, which promotes it to become a Generic Activity. Alternatively, the user can select from a menu a specific Activity type. Individual to-dos and Generic Activities may both be promoted into typed Activities such as:

- Meeting
- Trip
- Writing
- Production/Project Mgmt
- Editing (e.g. Journal)
- Interview Job Candidate

**Activity-Centered Task Assistant**

Once a user creates an ACTA Activity, a folder appears in their email folder hierarchy with the same name (Figure 2). Depending on its type, it will have some of at least the following Components (where italics indicate subtypes of a preceding non-italicized Component):

- Contacts
- Attendees
- Authors
- Candidates
- Interviewers
- Reviewers
- Calendar
- Tasks
- Scratchpad
- Correspondence
- Documents
- Submissions
- *Presentations*
- Forms
- Venue
- Agenda
- Interview schedule

The Components inside Activities appear as sub-folders. However, unlike regular folders they have specialized behaviors and views:

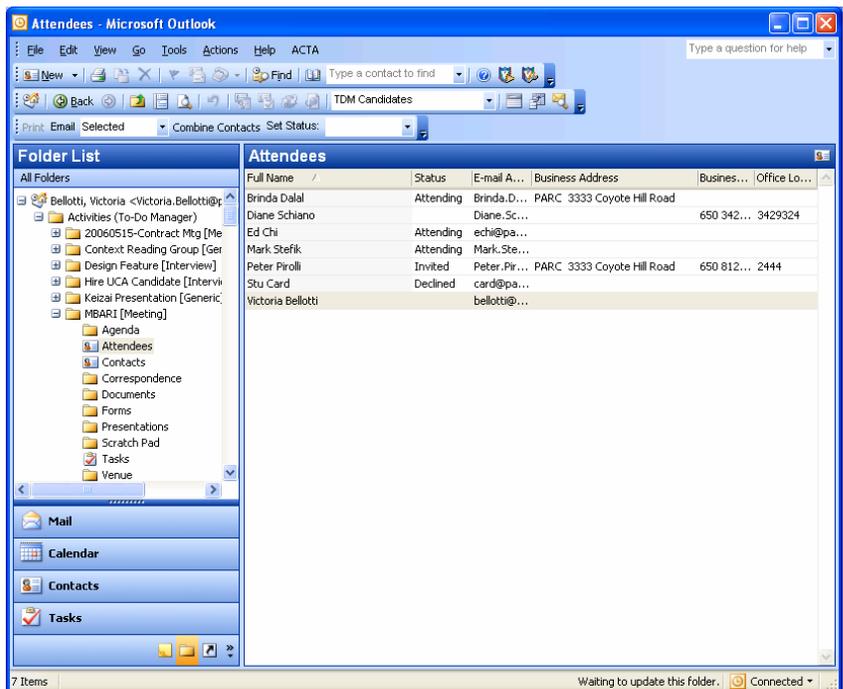


Figure 2. ACTA showing Attendees Component

- **Contacts** (and its sub-types) shows a list of names with contact details and role and status fields. Relevant buttons allow easy creation of email messages to selected people or people with a certain role or status. Contacts are also stored in Outlook's main contacts database.
- **Tasks** shows a list of tasks with status and assignees. Tasks are also stored in Outlook's tasks database.
- **Correspondence** is similar to a mail folder.
- **Documents** (and its subtypes) shows files and links.
- **Forms** shows links to Activity-related forms on the web.
- **Venue** shows location and a map.
- **Agenda** and **Interview Schedule** are special-purpose resources for arranging items (people and documents) by drag-and-drop. They are the most complex Components we have designed so far as they provide a custom interface for managing two types of live documents.

Users can drag items into an Activity or a Component and ACTA will extract relevant information and place it in the appropriate locations. For example, if the user drags a mail message with some links and attachments into the Attendees Component in a Meeting Activity, ACTA will:

- Ask if the user wants to add the sender and recipients to the Attendees Component.
- Add the same addresses to Contacts.
- Add the attendees to a panel in the Agenda, from which those who are speaking can be dragged to appropriate agenda items.
- Add the message to Correspondence.
- Add attachments and links to the Documents Component.

Whilst there is some similarity of such behaviors across Activities, some are specific to the Activity type and the Component in question.

To summarize, the significance of Activities and their Components is that they have pre-determined meanings valuable to both the user and the system:

**User Semantics:** Activity Components correspond to aspects of an activity that will be familiar to most people. Users who are unfamiliar with an activity can use some elements (particularly forms) as a guide to what the Activity might entail. Thus Components represent reminders, educational tools, and relevant resources for the activity at hand.

**System Semantics:** Activity Components correspond to elements of the system's model of the activity. Each Component performs a different kind of function to process and present information. For example, it may process an item dragged to it by:

- Looking for and extracting different entities such as dates, locations, contacts, special or characteristic terms, etc. from the information item.
- Displaying the information item differently, e.g., as a list item or a form with populated fields.
- Propagating relevant information to other Components.

- Enabling inferences based on extracted information entities

Our prototype is currently implemented as a Microsoft Outlook™ add-in, using the standard APIs to inspect and manipulate data, extend the native user interface, and detect user actions such as dragging files or messages into Activity folders. TV-ACTA is implemented in C# plus small bits of HTML and VBScript and uses the .NET framework, the Redemption library [18], and some third-party .NET components. All persistent state is saved in an Outlook store. We are refining a compositional architecture for reusing our Component functionality with the goal that new Activity templates will be built eventually by end users without significant programming.

We have implemented three types of user interface in and around the standard Outlook UI. Some Components present customized folder views with supplementary controls on an ACTA toolbar. In other cases, we exploit the ability of Outlook to render web pages as the presentation of a folder within an Outlook frame. Finally, our most complex interfaces have their own custom windows and we use a variety of custom dialog boxes throughout. TaskVista is implemented as yet another custom window of the same add-in, albeit one intended to remain continuously accessible at the edge of the desktop.

#### *Illustrative Scenario*

The following scenario gives a flavor of the way in which Components support special-purpose entity extraction and the building of a model of what is going on in the user's world. We also hope that the scenario gives a hint of the potential of task-specific structures to simplify a person's work when they can be used with little more effort than that required for conventional filing.

Alice uses the Activity Centered Task Assistant (ACTA) to help with her TM and PIM work. She is organizing a meeting to discuss joint-business opportunities with a number of contacts that she has developed over the last year. Recently she created a to-do item on TaskVista to remind herself to start organizing the meeting. She now promotes this into a Meeting Activity called 'Opportunities Meeting' which causes a new Activity folder with this name to appear in her Microsoft Outlook folder hierarchy.

She then drags some of the emails she has received on the subject to the Attendees Component of the Opportunities Meeting. As she does so, ACTA asks her which of the addresses of sender and recipients to add to Attendees and Contacts. She drags other messages to correspondence and ACTA asks if other additional addressees should be added to the Contacts.

Alice opens the Attendees Component and selects a button to create a message to all the attendees. She then writes an invitation to the meeting. As replies come in, she updates the status of the attendees to be 'attending' or 'declined' so that she can keep track of all the answers. Finally she deletes all those that have declined.

When Alice first opens the Agenda Component she sets the start-time and default item duration for the meeting at the top. The attendees area shows a list of all the attendees in the Attendees

Component. She creates items in the Agenda by dragging the presenter names into it, which creates a move-able agenda item.

The people who are presenting at the meeting eventually send their presentation slides to Alice and she adds them to the Presentations Component whereupon they also appear in a panel in the Agenda and can be dragged to the appropriate item. Alice is able to reorder items and change their durations causing all of the start times to update automatically. Once she has finished the Agenda, she emails a text version to all the attendees, with one menu item selection. On the day of the meeting she is able to run the entire meeting by accessing presentations from the Agenda.

#### ONGOING AND FUTURE WORK

We are now using ACTA for our own PIM and planning a full user-evaluation starting June 2006. Our aim is simply to show that ACTA is acceptable to a large proportion of users even without machine learning capabilities. If the user-interface is an improvement on the existing state of affairs then so much the better, but our ultimate goal is to enable machine learning to perform better with the additional context that ACTA can provide about the meaning of information items. Once machine learning capabilities are added to ACTA, it will be evaluated again to determine:

- Whether machine learning enhances ACTA's acceptability to end-users.
- Whether machine learning improves efficiency of TM and PIM.

#### REFERENCES

1. Bellotti, V., Dalal, D., Good, N., Flynn, P., Bobrow, D. G., and Ducheneaut, N. What a to-do: Studies of task management towards the design of a personal task list manager. *Proc. CHI 2004*, ACM Press (2004), 735-742.
2. Bellotti, V., Ducheneaut, N., Howard, M., Smith, I., and Grinter, R. E. Quality versus quantity: Email-centric task-management and its relationship with overload. In Bellotti, V., Moody, P. and Whittaker S. (Eds) *Human Computer-Interaction, Special Issue on Reinventing and Revisiting Email, 20, 1 & 2*. Lawrence Earlbaum Associates (2005), 89-138.
3. Boardman R., Sasse A. "Stuff goes into the computer and doesn't come out": a cross-tool study of personal information management. *Proc. CHI 2004*, ACM Press (2004), 583-590.
4. Boone, G. Concept Features in Re:Agent, an intelligent email agent. *Proc. Second International Conference on Autonomous Agents*, ACM Press (1998), 141-148.
5. Canny J. GaP: A Factor Model for Discrete Data. *Proc. SIGIR 2004*, ACM Press (2004), 122 - 129.
6. DARPA Solicitation: SOL BAA 02-21 - [http://www.darpa.mil/ipto/Solicitations/open/02-21\\_Mod6.htm](http://www.darpa.mil/ipto/Solicitations/open/02-21_Mod6.htm)
7. Dragunov, A., Dietterich, T., Johnsrude, K., McLaughlin, M., Lida L., Herlocker, J. TaskTracer: A desktop environment to support multi-tasking knowledge workers. *Proc. IUI 2005 International Conference on Intelligent User Interfaces*, ACM Press (2005), 75-82.
8. Ducheneaut, N. and Bellotti, V. 2001. E-mail as habitat: an exploration of embedded personal information management. *Interactions* 8, 5 (Sep. 2001), 30-38.
9. Gervasio, M., Moffitt, M., Pollack, M., Taylor, M. and Uribe, T. (2005). Active preference learning for personalized calendar scheduling assistance. *Proc. IUI 2005*, ACM Press (2005), 90-97.
10. Gwizdka, J. Supporting prospective information in email. *Ext. Abstracts CHI '01*. ACM Press, (2001), 135-136.
11. Henderson, A. Card, S. Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics* 5, 3, ACM Press (1987), 211-243.
12. Henderson, S. 2005. Genre, task, topic and time: facets of personal digital document management. In *Proc. CHINZ '05 ACM SIGCHI New Zealand Chapter's international Conference on Computer-Human interaction: Making CHI Natural. Vol. 94*. ACM Press (2005), 75-82.
13. Horvitz, E. Principles of mixed-initiative user interfaces. *Proc. CHI '99*, ACM Press (1999), 159-166.
14. Kushmerick, N. and Lau, T. Automated email activity management: an unsupervised learning approach. In *Proc. IUI '05 International Conference on Intelligent User Interfaces*. ACM Press, (2005), 67-74.
15. Mock, K. An experimental framework for email categorization and management. *Proc. SIGIR Conference on Research and Development in Information Retrieval*, ACM Press (2001), 392-393.
16. Segal, R., and Kephart, J. MailCat: An intelligent assistant for organizing email. *Proc. Third Annual Conference on Autonomous Agents*, ACM Press (1999), 276-282.
17. Smith, G., Baudisch, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D., Horvitz, E., and Andrews, D. GroupBar: The TaskBar evolved. *Proc. OZCHI 2003, CHISIG, Ergonomics Society of Australia*, (2003), 34-43.
18. Streblechenko, D. Outlook Redemption. <http://www.dimastr.com/redemption/>
19. Stumpf, S. Bao, X. Dragunov, A. Diettrich, T., Herlocker, J., Johnsrude, K. Li, L. and Shen, J. The TaskTracer system. *Proc. AAAI 2005*, AAAI Press (2005), 1712-1713.
20. Whittaker, S. and Sidner, C. Email overload: exploring personal information management of email. *Proc. CHI '96*. ACM Press (1996), 276-283.