
TV-ACTA: Embedding an Activity-Centered Interface for Task Management in Email

Victoria Bellotti*, Jim Thornton*, Alvin Chin[‡], Diane Schiano*, Nathan Good[†]

*Palo Alto Research Center
Palo Alto, CA 94304, USA
firstname.lastname@parc.com

[‡]Interactive Media Lab
University of Toronto
achin@cs.toronto.edu

[†]School of Information Management
Systems, UC Berkeley
ngood@sims.berkeley.edu

Abstract

In contextual computing, where cues beyond direct user input are used to trigger computation, one of the most daunting challenges is inferring what the user is doing. For the domain of task management, we have developed a new approach to reducing the problem of ambiguity of user action for intelligent systems. We introduce a construct we call an Activity, designed to reduce this ambiguity by providing a meaningful structure for task information that assists users with their work. We present ethnographic research and prototype evaluations to assess the value of the Activity construct from an end-user's perspective. Our findings suggest that the Activity structure is useful to people and therefore could be exploited for inference.

1 Introduction

Our work aims to tackle three challenges associated with task management (TM) and personal-information management (PIM).

Firstly we aim to confront the challenge of *cross-media PIM*; research suggests that integration of email, files and web bookmark organization is highly desirable [1,4,6]. Currently users have to invest at least three times as much effort as they should ideally have to, if they wish to keep consistent collections across all three types of media.

Secondly, we take on the challenge of supporting *complex interdependent tasks*. These are tasks that require people to monitor and wait for the work of others, and seem to be a major contributor people's sense of email overload [3].

Thirdly, our work contributes to a DARPA initiative (CALO: Cognitive Assistant that Learns and Organizes) to develop cognitive assistants that reason about and learn from user activity. Such assistants are intended to

support TM by providing reminders and automating simpler tedious tasks [10]. Our work is intended to mitigate the challenge inherent in this agenda of *enabling system inference* as to what the user is doing in TM terms as they manipulate information items.

We are addressing each of these three challenges by developing a construct we call an *Activity*. An Activity is a container for organizing heterogeneous content. It is presented to the user as a ready-made folder with various types of sub-folders, with labels like Contacts, Documents and Tasks, into which a user can drag emails, documents and links. Content is then processed and presented differently according to which sub-folder it is dragged to.

Activities come in various types (Meeting, Hiring, etc.), created from different pre-defined templates corresponding to common complex interdependent human activities such as organizing a meeting, hiring a new staff member, writing a report, or conducting an annual performance review. When an instance of an Activity type is created, it presents a set of sub-folders unique to its type, to provide the user with activity-specific resources for organizing messages, documents, contacts, schedules, and so on.

Apart from providing cross-media TM and PIM support for complex interdependent tasks [as defined in 3], Activities are also designed to facilitate automatic inference of what the user is doing; *by selecting a type of Activity and placing content within its pre-defined structure, the user is implicitly tagging content for system inference.*

2 Related Work

Recently, many researchers have begun to use lexical analysis and statistical modeling techniques to simplify task management. This work can be characterized as a desktop-based contextual computing approach to TM and PIM and includes the following thrusts.

Firstly, some have focused on reducing the overheads of filing incoming mail [7,21,26]. Secondly, others have used computation to anticipate users' goals and

provide mechanisms for system initiative in office tasks such as calendaring. [9,15,19]. Thirdly, we have seen attempts to identify task activity from analyzing the contents of emails, links and documents or collections or sequences of these that people are working on [12,20,25,30].

The approaches cited above all adopt a largely data-driven approach to inferring task-related properties of human action. We propose that, at least for GUI-based activities, this approach can be augmented with explicit, activity-based UIs for TM and PIM that can reduce uncertainty in making inferences about human behavior. By adding content to and manipulating it within a system-structured Activity, the user provides useful metadata to the system.

We contrast our work with content grouping tools [e.g., 17,28]. While grouping usefully allows users to gather together activity-related content and resources, it provides no system-interpretable semantics that can characterize the type of work the user is doing. To the system, any grouping is semantically equivalent to any other grouping.

Our work is somewhat related to the established domain of Workflow Management (WFM) technologies [14]. WFM system-supported processes are pre-defined to capture regularities of enterprise work and users are required to follow particular sequences of actions. However, work is only truly supported when it can be more or less routinely carried out the same way each time, leading to criticisms of inflexibility and attempts to soften the tyranny of centrally imposed structure on work practice [5, 11].

By contrast, activity-centered “ad hoc workflow” and groupware tools provide far greater freedom for users to create their own process representations through shared resources that can include user-generated knowledge about process. A good example is the Unified Activity Management (UAM) approach of Moran and his colleagues at IBM [22, 23].

Our own approach is similar to and distinguishable from Workflow and UAM in the following ways:

- Like WFM but unlike UAM, we anticipate regularities of commonly performed tasks.
- Unlike WFM but like UAM, we do not impose any sequences of actions on our users. Our structures are purely for organizing content, not enforcing procedures.
- Unlike WFM and UAM:
 - We do not require that users share any software with their collaborators.
 - Our software is deeply embedded in an existing common email client, (presently Microsoft Outlook™). This is because much knowledge work is received and executed through email [13]

and users are resistant to trying new clients (especially prototypes) because they need ongoing access to legacy content from their old client [1].

- Our software is specifically designed to explore the provision of pre-determined structures for user collections of content in order to assist machine inference about human activity.

3 Key Questions

Our design decisions raise two specific questions that we need to test in evaluation with users:

1. Will users adopt system-generated structures in preference to their idiosyncratic PIM styles?
2. Can we provide an incentive for users to label their content to facilitate machine inference?

These questions are addressed in the remainder of this paper, along with several other critical design issues. We begin by introducing our design ideas and then present findings from fieldwork that both motivated and evaluated the results of our design work.

4 Activity Constructs: Contextualizing Information and Human Action

We begin this section on our approach to supporting TM and PIM by explaining our current prototype system, which conveys the basic Activity idea. We then go on to discuss the broader design issues that we are exploring.

4.1 Implementing Activities

We have implemented a prototype called TV-ACTA that combines a to-do list, TaskVista (TV) with an Activity-Centered Task Assistant (ACTA) embedded in Outlook. At this stage TV-ACTA has no intelligent back-end; our principal goal so far has been to answer the two key questions introduced above before proceeding further.

TV (see Figure 1) is an electronic to-do list [2], which is intended to be used for email and non-email to-dos. The idea with respect to email is that short term to-dos reside as messages in the in-box as always, but when the user wishes to clean out her inbox, she can drag a message to TV where it may reside until it is dealt with. The user can also type in text to-dos in a field at the top of TV or drag items such as documents from her desktop to become to-dos. Clicking on a to-do opens an Outlook task pane, if it was typed in, or else opens an item that was dragged in.

The user creates an ACTA Activity directly from TV, or may convert an existing email, file, or typed-in to-do on the list into an Activity if it becomes clear that the ongoing task is becoming complex enough to justify creating a collection of content. The user can either use

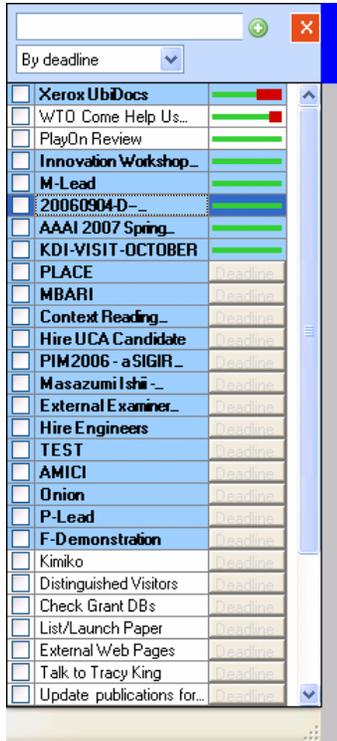


Figure 1. TV, an electronic to-do list from which Activities can be launched

a menu to choose from different Activity templates or drag a second item onto an existing to-do to create a Generic Activity which can later be converted to a specialized type if need be. A to-do turns blue when it becomes an Activity and a corresponding folder collection automatically appears in the Outlook folders list (see Figure 2) for the user to place content in. Just as a user might have different folder instances for different projects, the user creates different Activity instances for different work activities.

An instance of an ACTA Activity is presented as a folder with the same name as the TV item. It contains ready-made Components (presented as subfolders) that can be used to collect information related to a complex task. Figure 2 shows the folder list view of the three Activity types we have so far implemented; *Generic*, *Interview* and *Meeting* (all figures show real Activities created and used by one of the authors of this paper, so some names and addresses are grayed out for privacy reasons).

Various special purpose subfolders called Components (Figure 2) are drag-and-drop targets and containers for different types of information within an Activity:

- *Generic Activities* have only the common Components that are shared by all Activity types.

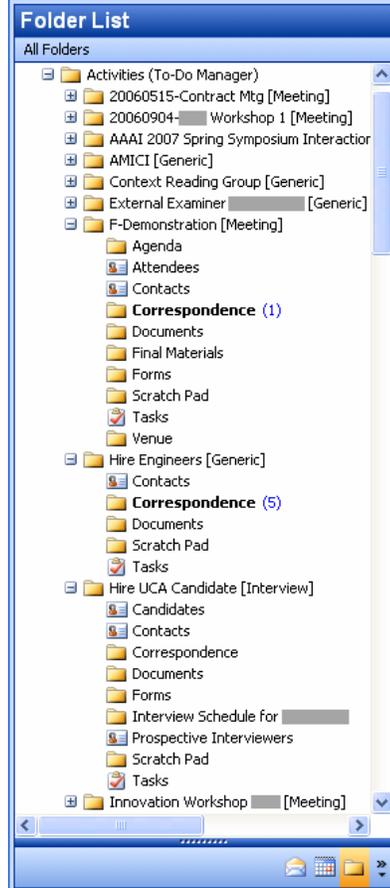


Figure 2. ACTA Activities components in Outlook folder list

- *Correspondence* is similar to a normal email folder and collects all email added to the Activity. When an email message is added to Correspondence, ACTA extracts contacts, documents and links to add to the *Contacts* and *Documents* Components.
- *Contacts* (Figure 3) collects email addresses as the user adds messages to the Activity and looks them up and synchronizes them with Outlook's contact database (the user can exclude or delete addresses if he chooses).
- *Documents* (Figure 4) collects documents and links from messages and these can be opened from this Component's viewer.
- *Scratch Pad* is used for typing notes particular to the Activity (e.g., references for papers to read).
- *Tasks* contains to-dos specific to the Activity and new tasks are created by dragging in email messages or typing into a "New" field.

Meeting Activities have some special Components:

- *Agenda* is a special-purpose drag-and-drop agenda creation and presenter assignment tool. It can be published to and re-synchronized with a shared Outlook calendar. It can also be included in an editable email message addressed to attendees or presenters at the touch of a button.

- *Attendees* is a place to drag contacts to appear in the Agenda.
- *Final Materials* is a place to drag presentations and handouts that will appear in the Agenda, from which they can be opened.
- *Forms* contains links to forms that are useful for organizing the Activity such as links to catering and expense forms (this Component is ultimately intended to be customizable).
- *Venue* presents a map view of the location (the address can be typed in here or in the agenda).

Interview Activities have other special Components:

- *Candidates* is for the contact listings of candidates for a position to which the Activity relates.
- *Interview Schedule* is created for a candidate when the user selects an option to "Schedule Candidate" in the Candidates Component. It is created from a set-up widget that allows parameters to be specified such

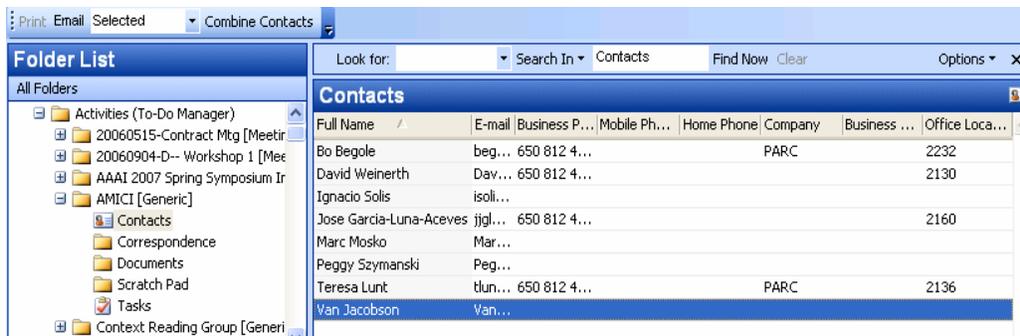


Figure 3. Contacts in an Activity, extracted from emails and matched with Outlook contacts

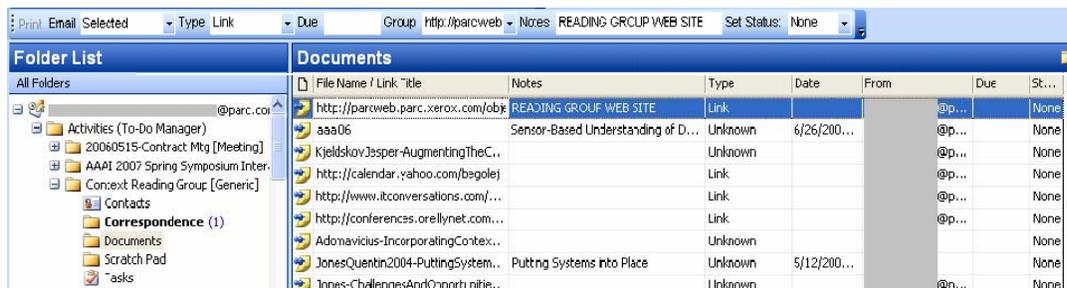


Figure 4. Documents in an Activity, extracted from emails or dragged from desktop or folder

as the date, start time and so on. Once the schedule is created, the user can drag interviewers into and between slots. The schedule can be published to a shared Outlook calendar if interviewers prefer to choose their own slot in the schedule. It can also be sent to interviewers in text form via a single button to create the addressed message.

- *Prospective Interviewers* is for contacts who might interview the Candidate. They appear in the Interview Schedule and can be dragged to its slots.

4.2 Implementation Issues

Our current TV-ACTA prototype is implemented as an add-in to Microsoft Outlook™, relying on its extensibility and customization features to enable the to-do list and embedded Activity components we have described. The visible UI additions are created in three ways. First, tabular views such as those in Figure 3 are created through Outlook's programmable view customization for folders plus custom toolbar controls. Second, some component folders, such as Venue, are implemented with an HTML page, exploiting Outlook's folder home page feature. Finally, we have added just a few custom window and dialog box types such as the TV window in Figure 1. Much of the behavior of TV-ACTA is triggered by Outlook event notifications. For example, upon user drag of a mail message into a component folder, TV-ACTA receives event notifications that cause it to do such things as saving contacts from the recipient addresses and extracting documents. In a variety of cases, TV-ACTA overrides

or alters the default behavior of Outlook, such as by handling the drag of a contact from one activity to another so that it is effectively a copy rather than the move that would be the native Outlook outcome. All persistent data is stored in the user's Exchange MAPI store, so features like offline operation work as expected.

Standard Outlook

items such as Tasks are used wherever that makes sense, but often TV-ACTA also stores data in custom fields or hidden items, in the normal manner of Outlook add-ins. TV-ACTA is written in C# plus small amounts of HTML and script code. In addition to Outlook it relies upon the .NET framework, the Redemption library for Outlook access [29], and some third-party GUI components.

The biggest implementation challenges involve working around the limitations of Outlook as a platform. Outlook does not allow customization of everything we would like and some useful customizations do not work cleanly. For example, we can display custom fields in a custom tabular view, but editing of those fields in-line rarely works as a user would expect. Another challenge is to exploit the available event notifications to figure out what is going on, control Outlook, and avoid infinite loops. Finally, there are many performance challenges especially because the simple ways to work with data using Outlook APIs are inefficient. These are all areas that need work to improve usability, and we have learned much from the experiences of users so far that will inform redesign.

The major design challenge is to support a wide variety of users in their varied, complex tasks. To make this feasible, we must create a compositional architecture for reusing functionality so that new Activities can be supported without significant programming. We are focusing on customizable component types as the basic building blocks. Our experience so far suggests that this

is a promising approach but much remains to be done to make it possible for end users to create their own new Activity structures.

4.3 General Design Issues

4.3.1 Embedding Cross-Media TM and PIM in Email

Our design is informed by research into TM and PIM that suggests that knowledge workers in many professions tend to spend a great deal of time working from their email and need TM and PIM resources to be tightly integrated with email work [1,4,6,13]. Based upon this, we integrated ACTA tightly with Outlook because it is a common email client with a relatively open architecture allowing add-ins and customizations. While we lost some flexibility in implementation, we were able to use Outlook's existing data structures such as contacts and tasks; for example TV stores its to-dos in Outlook's tasks database.

4.3.2 Complex Interdependent Tasks: From Email to To-Dos to Activities

We designed ACTA to support complex interdependent tasks, which are hard to keep track of in email [3]. They cannot be executed immediately because they are waiting on work to be done by others, which must be tracked and managed by the task's owner. To-do lists tend to be used more for postponed to-dos [2], whereas the email inbox is the de facto to-do list for tasks that came in email [1, 31]. A postponed to-do item may end up in either or both places depending on circumstances (for example, a user wishing to avoid losing an email that can't be dealt with immediately may move it to a to-do list from her overcrowded inbox).

We envisaged that the lifecycle of a to-do representing a complex interdependent task might be: To arrive in the inbox; to be postponed when it cannot be responded to; to then be moved to the to-do list; and finally to be promoted to an Activity if it becomes complex enough to demand an entire collection of messages and other content. This motivated the pairing of TV and ACTA in our design.

4.3.3 From Unstructured Work to Pre-Defined Structure

One of the two key questions associated with our work is, 'Will users adopt system-generated structures in preference to their idiosyncratic PIM styles?' While this is something we are trying to determine empirically, we take seriously the idea that prematurely imposing structure on work can make a system impossible to use. In the words of Shipman and Marshall, "*Users are hesitant about formalization because of a fear of prematurely committing to a specific perspective on their tasks.*" [27]. For this reason, we allow users to

gradually evolve to-dos into Generic Activities and then, if they choose, to type an Activity. The reward to the user for increasing the specificity and formality of definitions is to obtain more activity-specific resources, but with ACTA, they need not commit early or indeed ever.

4.3.4 Using Activities to Get Users to Tell us What They are Doing

The second key question our work raises is, 'Can we provide an incentive for users to label their content to facilitate machine inference?' Given the current tools, most people file useful electronic items for future use [6,13,24] and reuse email either in the inbox or stored in folders to support prospective tasks [3,16], although this is not well supported by current email clients [1,3,16,31]. When users create these folders and sub-folders they imbue them with personal meaning [18]. Every time they drag an item into one of these folders (or some other PIM or TM resource) the cognitive and manual effort that they invest in characterizing the item and dragging it somewhere is wasted from a system perspective. However, we believe this user effort could be 'hijacked' for content labeling.

This labeling problem is solved in a related system, TaskTracer, [12,30] by requiring users to enter a description term for their current task as they manipulate files. This explicit term (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 the content associated with a task, no matter where it is. However, this is at the expense of that user having to remember to keep explicitly telling the system what he is doing.

By contrast, our trick is to get users to implicitly inform the system of what they are doing whenever they drag an item to a specific Component in some Activity. In this way, they invest that decision-making and filing effort in fleshing out a system schema that can be used to reason about the information items and support user activity. So we provide a pre-designed organizational structure that is richer and more helpful for the task than generic folders, and that the system 'understands' as being a coherent user activity *of some type*. This requires no additional labeling step for the user.

Activities accept email messages, links and files. However, they are more than just heterogeneous folders. Components offer diverse structured views onto content that users have added. A user can also attach metadata (that an inferencing back-end might be able to use) while maintaining links to source items. For example, a user might drag an email message from someone accepting a meeting invitation into a Meeting Attendees Component, whereupon a new contact will be created for that sender and added to the Attendees list. She might mark the attendee's status as 'attending'.

But, since the contact item is still connected to the message from which it came, she can also simply open the original email from the attendee at any time to inspect what he actually said about attending.

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

User Semantics: Activity Components correspond to aspects of an activity that will be familiar to those experienced with the activity. Users who are unfamiliar with the activity can learn the importance of these elements by exploration. Thus Components represent reminders, educational tools, and highly relevant resources that help a user complete the activity at hand.

System Semantics: Activity Components correspond to elements of a system model of a type of activity. Each Component performs a different kind of function to process and present content and metadata. 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.
- Recording information (metadata) added by the user such as the status of a document.

Context: All Components are contextualized within a unique Activity instance so user action avoids errors of automated classification in the following situations:

- Multiple instances of the same type of activity exist with much overlap in content so that automatic classification is unable to distinguish between them.
- Items related to the same activity are highly dissimilar so that automatic classification is unable to group them.

4.3.5 Folders versus Tags

Despite the advent of powerful search tools, many users cling to folders in email and in their file storage systems because they fear they will not be able to easily recall or define all of their important content items in order to be sure of retrieving them when needed. Like [8] we advocate tagging schemes that allow items to appear in multiple categories at once. However, in our prototyping efforts, we determined that embedding of PIM resources in a common client was more important goal. So, since Outlook is folder-centric, we made a trade-off and stayed with simple folders rather than diverting effort to developing a tagging system.

5 Iterative User Studies

5.1 Motivating the Activity Construct Idea

In order to justify developing a working prototype of our software, we had three *motivating questions*:

1. Is the idea of a system-generated Activity acceptable to prospective users?
2. Does sufficient structure in ad hoc work processes exist to justify a one-size fits all Activity for each type of work that we might choose to support?
3. Do users require common activity-specific resources that an Activity could provide to encourage its use?

5.2 Preliminary Study

To address the first motivating question, we showed an early prototype of an ACTA Meeting Activity to Administrative Associates (AAs) at our organization. The AAs approved of the structure we had defined for organizing information and responded well to the fact that it did not require others to use the same system in order to be useful to them. This preliminary feedback encouraged us to continue our design efforts.

5.3 Study of Ad Hoc Activity Structure

In a second study, we addressed our second and third motivating questions by examining hiring; a complex activity involving coordination within and outside an organization and that might be expected to vary across organizations. We interviewed 15 people from all parts of our organization and three more from each of three other knowledge work organizations: An entertainment-software development company, a university department and a non-profit research organization.

No formal workflow tool was used for hiring anywhere but, despite variations in the process, we found many near-universal features that were rated as highly important by almost all interviewees, regardless of their organization or past roles in the process:

- **Common roles** including coordinator or host, candidate and interviewers.
- **Common documents** including resume, letters of reference and interview schedule.
- **Common resources** including email inbox, email announcements and online folders for content.
- **Common tasks** including telephone interviews, requesting references and scheduling interviews.

This suggests that ad hoc activities share properties across organizations. Of course, ultimately, the true test of the validity of this finding would be in testing our software in multiple organizations beyond our own.

5.4 Walkthrough Evaluation Study

After the ad hoc activity study, we developed a version of ACTA with many of the features described earlier, which was sufficiently robust to support a structured walkthrough of the Hiring Activity.

We conducted the walkthrough with 8 staff members from our organization; an AA (who had supported a hiring process 35 times), a researcher-and-hiring-coordinator (with 15 experiences) and six research managers (with 2-10 experiences of performing the host role). Two participants previously used a shared calendar tool allowing interviewers to assign their own interview slots and five created schedules themselves from a standard Word template shared via email.

Participants were asked to describe how they currently perform 8 tasks that our Hiring Activity supported and rate each in terms of how satisfied they were with their current tools where 1="very dissatisfied" and 5="very satisfied". They were then walked through the Activity tool prototype and shown how to perform the tasks with ACTA and asked to rate it on the same satisfaction scale. Overall the participants rated ACTA as a 4 (std dev 0.98) and their current tool as a 3.1 (std dev 1.45).

The main criticism of ACTA was that it did not have a shared calendar allowing people to sign themselves up into the interview schedule. For this reason, one of the two participants who previously used a shared calendar very much disliked our tool rating it on average 2.2 on a scale of 1 to 5. The other rated ACTA 4.4 overall but also criticized it for lack of shared calendaring.

The key question the walk-through addressed was whether an Activity-based model for managing laborious tasks appealed to prospective users. Despite lacking a shared calendar, ACTA seemed preferable to existing resources. This justified developing a working prototype and the feedback also inspired us to enable users to publish to and synchronize with a shared calendar in our next prototype.

5.5 Evaluating TV-ACTA in Daily Use

A working prototype of TV and ACTA was developed for in situ evaluation. It was pre-tested in daily use by three of the authors, one of whom used it for several months and despite various bugs and limitations it was finally deemed good enough to test our design ideas.

Twelve diverse volunteers were recruited to use and give us feedback on our TV-ACTA prototype (Table 1):

Role	Admin	Attorney	BizDev	Liaison	Librarian	Manager	Publicist
No.	4	1	1	1	2	2	1

Table 1. Twelve participants who used and gave feedback about TV-ACTA

The four "Admins" (AAs) have very diverse roles, such as personal assistant, lab coordinator, event coordinator and legal assistant. The "Liaison" assists with customer relations. One "Manager" manages several software engineers who work with an external customer; the other is head of a 40-person non-research organization. Only two of the participants have any programming skills that would suggest technical sophistication.

5.5.1 Daily Use Evaluation Procedure

Volunteers were trained on how to use TV-ACTA at installation time. They were also given a manual. During the usage period, a researcher would stop by occasionally to determine if there were any ongoing problems, answer questions and give advice and tips.

After at least 3 weeks, (in most cases 4) a semi-structured debriefing interview was conducted with each of the participants that tried out the software. The first part involved an open ended section in which users were asked to discuss usability, intuitiveness and usefulness of the software. The second involved a formal series of questions including a series of feature rating questions using a 7-point Likert scale; 1=Extreme Dislike; 7=Extreme Like (plus notations for forgetting or never trying features).

5.6 Findings

We begin this section with two pragmatic issues which impacted our evaluation. We then turn to more general findings organized in terms of the issues presented earlier in the "General Design Issues" section.

5.6.1 Acclimatization

On average, participants had 7.5 active TV to-dos (StdDev 5.2) and 2.8 with deadlines (StdDev 2.8) and 5.1 Activities (StdDev 6.6) at the time of the debriefing interview (not including deleted, completed or archived to-dos and Activities). However, several reported not having had enough time to become really familiar with all the new features of TV-ACTA. By comparison, participants estimated creating on average only 2.1 normal Outlook folders during that time.

This suggests that we should have continued the evaluation for far longer with refresher training sessions to allow time for more Activity creation. Seven participants explicitly stated they would like to keep using the software and only two requested that we uninstall it after the debriefing.

5.6.2 Trade-offs of Integration

While we benefited from integrating our software with Outlook, some of its features worked against us. Firstly, a common default view of Outlook mail folders, "Mail," does not show people Components (Contacts, Attendees, Interviewers) in the folder list. Despite

warnings 2 participants forgot about the invisible people Components. Also, one user was using Outlook's Favorite Folders features (visible only in the incompatible "Mail" folders view) very much like Activities for ongoing work. Our Activities would not automatically show up in that set, so she found them hard to use.

These flaws underline the challenges of embedding new software in an existing highly-developed application.

5.6.3 Usefulness of TV-ACTA.

On average, users rated the usefulness of both TV-ACTA and (for comparison) Outlook as being the same; 6.1 – Considerable Usefulness. They also guessed that they used TV and or ACTA 6.4 times a day on average (when asked to give a rough number estimate). The following findings sections provide more detail by addressing each of the issues we presented in the General Design Issues section of this paper.

5.6.4 Embedding Cross-Media TM and PIM in Email

Part of the final debriefing interview covered users' reactions to the general idea of embedding TM and PIM resources in their email. This was rated on average 6.5 (between Liking Considerably and Liking Extremely) by the twelve participants; as the Business Developer put it, *"I used to save it in a directory and then I would still save the e-mail. But because this allowed me to just drag and drop and it separated the document and I didn't know where it went, I just knew I could get back to it under an activity folder."* In particular, he liked that ACTA's embedding in Outlook ensured everything was synchronized with the server so he didn't have to back things up any other way.

5.6.5 Complex Interdependent Tasks: From Email to To-Dos to Activities

Our design anticipated that users would create TV to-dos principally from email messages, then promote to-dos to Activities as they evolved into more complex interdependent tasks in email. However, users estimated that they created only about 25% of their to-dos by drag-and-drop from email, so the rest were created by typing into TV. It is not clear, why typing was preferred but it could be that the inbox was still functioning as a to-do list for email and the TV to-dos were not represented by any email at creation time.

One participant only used TV and four participants only created one Activity. On the other hand one Administrative Assistant had 9 Activities and only 3 to-dos and the Business Developer stated that he was really only using his 11 Activities and that his 29 to-dos were less useful.

Just as there is variation in how people organize their TM and PIM, there were a variety of ways in which TV-ACTA was used. This suggests that we should develop our software to be as flexible as possible.

5.6.6 From Unstructured Work to Pre-Defined Structure

One of the most crucial questions in our debriefing interview was, 'How do you feel about automatic creation of Component folders?' We wondered if participants might reject structure imposed on their folder organization. However, the average rating was very high; 6.2 (Between Liking Considerably and Liking Extremely). Users also approved the types of Components we created (5.7) and, oddly, did not suggest other types that we might add.

However, the Liaison, who had been organizing a meeting, wanted to be able to discriminate between different types of attendees within a single event:

Liaison: *"I guess I used a lot correspondence... and also the contacts. And I am planning to use agenda but not yet. And I just made some random folders in here [in the Activity] by myself and that's actually good."*

Interviewer: *"Why did you add 'Dinner'?"*

Liaison: *"Because I wanted to keep track of who's attending and if I need to add more people I can just refer to it for example, I can just refer to this folder quickly."*

The Attendees component did not have a feature for subdividing attendees or linking them to different parts of an Activity. So the Liaison had to create a normal folder inside the Activity to capture this distinction.

5.6.7 Using Activities to Get Users to Tell us What They are Doing

A core goal in this research was to determine if users would willingly type Activities even though this was primarily for the benefit of the system. We only had time to create three types, but one of these was only general purpose and would thus not inform our system of the context of PIM. In fact, while most Activities observed at debriefing were Generic, 25% were Meetings and 5% were Hiring, 30% being typed seemed a reasonable number, given the limited choice.

Clearly, then, users are apparently willing to type their Activity collectives, but the question is then, were some activities going unlabeled?

It turned out that during the evaluation period only 4 participants needed to organize a meeting and only 2 needed to create a candidate schedule. Given these small numbers and the fact that 30% of the Activities being typed, we infer that when users found a match

with the type of activity they engaged in, they were often willing to assign a type to an Activity.

Due to the relatively infrequent need for the few typed Activities we had time to implement, the Components in the typed Activities were under-utilized. In fact, during development of our software, our lab switched from having managers organize candidate interviews to having one Administrative Associate (AA) do the job which lost us a lot of potential users. However it also meant that we had one avid user who created two Interview Activities and used them a great deal.

Despite such setbacks, we were able to obtain some feedback from actual use of typed Activities. Overall, due to some bugs, the Meeting Activity was rated only 5.3 (between Liking Somewhat and Liking Considerably) by 4 participants who used it, but the Interview Activity was rated 7 (Liking Extremely, in spite of its bugs) by 2 participants who used it.

Most criticisms were attributable purely to the bugs with users telling us they would otherwise have made much greater use of the resources and expressing a lot of interest in trying future versions of the software.

5.6.8 Folders versus Tags

Because we wanted Activities to behave as much as possible like normal mail folders, we designed drag-and-drop into an Activity to move items out of the inbox. To our surprise, this attempt at consistency with folders backfired on us. Several participants became confused because they expected items to still be in the inbox because they were still working on them.

“...correspondence, putting the documents all in one place. I like that and once I got used to the fact that they moved from my mail because my first instinct was you do the search and wait a second, I know I just got that. Oh, it's over there. So that, once you get used to that, I think that's fine. But it kind of had me.” – Manager.

Our software relies on a change in the way that users think about using folders. While this worked well for most, some found it tricky to get used to. A future system, based on tagging to allow items to reside in the inbox, might be a better alternative for some and is a direction we would like to pursue in the future.

5.7 Summary of Findings

Recalling the two key questions introduced earlier in this paper, our findings suggest that, even with a buggy prototype:

- Users are happy to use system generated structures for content even with no ‘smart’ behavior as a reward.

- Users are willing to assign types to their work activities, in essence, to tell us what kind of work they are doing, when we ‘piggyback’ content labeling on existing filing practice.

The goal of our research efforts, given our limited resources, was simply to develop a prototype that was just good enough to verify these two points before investing more resources in furthering our research agenda. We believe we have accomplished this goal. We also obtained feedback on how to improve TV-ACTA to make it more useful, usable and less buggy.

If we develop more Activity types such as ‘Writing,’ ‘Purchasing,’ ‘Hosting a Visit,’ ‘Travel’ and so on, we suspect the percentage of classified Activities will rise. Since we reuse existing Components across Activities it is not particularly expensive to generate a new type. We plan to refine our Activity designs and conduct further studies to identify further useful types.

However, the Holy Grail of our research is to provide context for reasoning and learning software to support automation of routine PIM (e.g., classification of content, finding related information or contacts, or suggesting to-dos). Such automation will further encourage use of TV-ACTA. Our findings suggest that our approach is acceptable even though it imposes structure and depends on users typing their collections. Users apparently find enough value in the existing design to persuade them to ‘play along’ with these impositions even without the benefits of automation.

6 CONCLUSIONS

Evidence from use of TV-ACTA supports the idea of imposing artificial structure on users’ TM and PIM and the idea of getting users to assign system types to their activities to provide context for an intelligent back-end. We plan to integrate TV-ACTA with back-end inferencing software from the DARPA CALO program and to conduct evaluations to see the extent to which additional context provided by TV-ACTA improves inferencing beyond what is possible with current non-Activity-centric resources.

7 ACKNOWLEDGMENTS

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. We thank the people who took part in our studies. We also thank Jonathan Cohen, Paul Rasmussen and Timothy Chang, our engineers.

References

1. Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. Taking email to task: The design and evaluation of a task management centered email tool. In *Proc. CHI 2003*, ACM Press (2003), 297-304.

2. 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.
3. 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.
4. Bergman, O., Beyth-Marom, R., and Nachmias, R. The project fragmentation problem in personal information management. *Proc. CHI'06*, ACM Press (2006), 271-274.
5. Bernstein, A. How can cooperative work tools support dynamic group process? Bridging the specificity frontier. *Proc. CSCW'00*, ACM Press, (2000), 279-288.
6. Boardman R., Sasse A. "Stuff goes into the computer and doesn't come out": a cross-tool study of personal information management. *Proc. CHI'04*, ACM Press (2004), 583-590.
7. Boone, G. Concept Features in Re:Agent, an intelligent email agent. *Proc. Second International Conference on Autonomous Agents*, ACM Press (1998), 141-148.
8. Cutrell, E., Robbins, D., Dumais, S., and Sarin, R. 2006. Fast, flexible filtering with phlat. *Proc. CHI'06*. ACM Press, (2006) 261-270.
9. Dabbish, L., Kraut, R., Fussell, S., and Kiesler, S. Understanding email use: predicting action on a message. *Proc. CHI '05*, ACM Press (2005), 691-700.
10. DARPA Solicitation: SOL BAA 02-21 - http://www.darpa.mil/ipto/Solicitations/open/02-21_Mod6.htm
11. Dourish, P., Holmes, J., MacLean, A., Marqvardsen, P., and Zbyslaw, A. 1996. Freeflow: mediating between representation and action in workflow systems. *Proc CHI'96, ACM Press (1996)*, 190-198.
12. Dragunov, A., Dieterich, T., Johnsrude, K., McLaughlin, M., Lida L., Herlocker, J. TaskTracer: A desktop environment to support multi-tasking knowledge workers. *Proc. IUI'05 International Conference on Intelligent User Interfaces*, ACM Press (2005), 75-82.
13. Ducheneaut, N. and Bellotti, V. 2001. E-mail as habitat: an exploration of embedded personal information management. *Interactions* 8, 5 (Sep. 2001), 30-38.
14. Georgakopoulos D., Hornick, M., Sheth A.: An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases* 3(2) (1995) 119-153.
15. Gervasio, M., Moffitt, M., Pollack, M., Taylor, M. and Uribe, T. (2005). Active preference learning for personalized calendar scheduling assistance. *Proc. IUI'05*, ACM Press (2005), 90-97.
16. Gwizdka, J. Supporting prospective information in email. *Ext. Abstracts CHI '01*. ACM Press, (2001), 135-136.
17. 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.
18. 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.
19. Horvitz, E. Principles of mixed-initiative user interfaces. *Proc. CHI'99*, ACM Press (1999), 159-166.
20. 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.
21. 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.
22. Moran, T. P., Cozzi, A., and Farrell, S. P. 2005. Unified activity management: supporting people in e-business. *Commun. ACM* 48, 12 (2005), 67-70.
23. Muller, M. J., Geyer, W., Brownholtz, B., Wilcox, E., and Millen, D. R. 2004. One-hundred days in an activity-centric collaboration environment based on shared objects. *Proc CHI'04*, ACM Press, (2004), 375-382.
24. Nardi, B. and Barreau, D. "Finding and reminding" revisited: appropriate metaphors for file organization at the desktop. *SIGCHI Bull.* 29, 1 (Jan. 1997), 76-78.
25. Richards, B., Kay, J., Quigley, A.: Activity modelling using email and web page classification. 10th International Conference on User Modeling (UM'05) Workshop 4, UM'05, Colin de la Higuera, Thierry Artières, NA, pp 60-67. 2005.
26. Segal, R., and Kephart, J. MailCat: An intelligent assistant for organizing email. *Proc. Third Annual Conference on Autonomous Agents*, ACM Press (1999), 276-282.
27. Shipman, F and Marshall, C. "Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer-Supported Cooperative Work*, 8, (4) (1999) 333-352.
28. 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.
29. Streblechenko, D. Outlook Redemption. <http://www.dimastr.com/redemption/>
30. Stumpf, S. Bao, X. Dragunov, A. Dietrich, T., Herlocker, J., Johnsrude, K. Li, L. and Shen, J. The TaskTracer system. *Proc. AAAI 2005*, AAAI Press (2005), 1712-1713.
31. Whittaker, S. and Sidner, C. Email overload: exploring personal information management of email. *Proc. CHI '96*. ACM Press (1996), 276-283.