

# What is an Activity? Appropriating an Activity-Centric System

Svetlana Yarosh<sup>1</sup>, Tara Matthews<sup>2</sup>, Thomas P. Moran<sup>2</sup>, and Barton Smith<sup>2</sup>

<sup>1</sup> Georgia Institute of Technology, Atlanta, Georgia, USA

<sup>2</sup> IBM Research, Almaden Research Center, San Jose, California, USA  
lana@cc.gatech.edu, {tmatthe, tpmoran, n6hdn}@us.ibm.com

**Abstract.** Activity-Centric Computing (ACC) systems seek to address the fragmentation of office work across tools and documents by allowing users to organize work around the computational construct of an *Activity*. Defining and structuring appropriate Activities within a system poses a challenge for users that must be overcome in order to benefit from ACC support. We know little about how knowledge workers appropriate the Activity construct. To address this, we studied users' appropriation of a production-quality ACC system, Lotus Activities, for everyday work by employees in a large corporation. We contribute to a better understanding of how users articulate their individual and collaborative work in the system by providing empirical evidence of their patterns of appropriation. We conclude by discussing how our findings can inform the design of other ACC systems for the workplace.

**Keywords:** Activity-Centric Computing, Appropriation, Office & Workplace

## 1 Introduction

Professionals whose work involves the creation, management, and distribution of information (known as *knowledge workers*) expend considerable effort managing work that has been divided across teams of contributors and a diverse collection of tools for handling documents, communicating, updating status, etc. [1,2]. Activity-Centric Computing (ACC) is an approach to address work fragmentation by allowing users to structure their work around the computational construct of an Activity [1,3-5]. This approach is loosely inspired by Activity Theory, a framework for studying socially situated phenomena originally developed by the Soviet school of Psychology. Adopting this framework for HCI, the term “activity” is used to refer to the basic unit of social analysis broadly defined as collective action towards a goal [6]. We use “Activity” (capital A) to refer to the computational construct and “activity” to refer to the social unit of analysis. Thus, an Activity is a locus for uniting information relevant to carrying out an activity and a persistent representation of its status, members, resources, constraints, and plans [1,3]. In this paper, we

present an empirical investigation of how users successfully structure their work activities as system Activities in an ACC system.

The ACC approach leverages the Activity construct to support the informal and emergent situational work of small groups, as has been validated through prototype deployments (e.g., [1,3]). ACC systems have several distinguishing characteristics. They focus work around activity goals, which separates them from communication systems like email. Unlike workflow systems, they allow the flexibility of on-the-fly (re)organization, are generic enough to represent a wide range of activity, and support lightweight creation and editing. To provide this generic and flexible support, ACC systems leave the responsibility of defining and structuring Activities to the user. However, prior studies have shown that this poses a major challenge: it is hard to delineate an activity's purpose, boundaries, and representation within a system, especially at an early stage [1,2].

Representing work in an ACC system involves thinking about how the Activity fits into existing working practices. Dourish [7] defines *appropriation* as the process of adopting and adapting systems to fit and evolve existing work practices. Workers must appropriate the Activity construct and the ACC system to benefit from its use. Yet, we know little about how they define, structure, and use an Activity in the process of doing real work. We address this gap by presenting empirical evidence of usage patterns and strategies emerging from the appropriation of one ACC system by some of its most active users.

We studied a production-quality ACC system, Lotus Activities. We present an investigation of 15 knowledge workers who have appropriated Lotus Activities for their everyday work. We focus on highly active users because they are most likely to have developed successful patterns and strategies of use that can inspire potential design directions of ACC systems. We address two questions to inform the design of ACC systems:

1. What use patterns do active users develop in appropriating Lotus Activities?
2. How do they divide and structure their work into effective Activities?

We begin by positioning our work in the context of other investigations of ACC systems. We describe Lotus Activities and its usage in one company. We then describe our interviews with 15 active users, and a content analysis of 120 of their Activities in the system. We present the patterns of use that emerged from this investigation. Finally, we discuss the appropriation of Lotus Activities and the Activity construct as evidenced by these patterns, highlighting the implications of our findings for ACC systems.

## 2 Related Empirical Work

Several activity-centric systems have explored this paradigm prior to Lotus Activities. Christensen and Bardram [4] promoted an activity-centric approach, grounded in the evaluations of an off-the-desktop prototype with health care workers in a series of workshops. Bardram *et al.* [1] developed operating system extensions to provide personal computer users with an activity-centric experience. Though feedback for the activity-centric approach was positive, they found that users had difficulty in defining activity boundaries in

the system. The UMEA system [5] attempted to address the problem of partitioning work by inferring personal project categorizations from the user's interaction with other tools. While the automated approach reduced the work needed to define Activities, it introduced substantial overhead to remove incorrectly associated items from projects. Over a long-term deployment of Giornata [8], a system supporting reframing one's desktop into manually declared activities, Volda & Mynatt found that users had trouble defining Activities because of the interdependent and complex nature of knowledge work. Bellotti *et al.* [9] explored the use of an Activity construct for email and to-do management to find that the construct was flexible enough to support individual PIM needs, but that users could benefit from suggestions of potential Activity structures. Our work can be distinguished from these systems because it focuses on collaborative rather than individual Activity management, but all of these explorations reveal a common problem: dividing and structuring work into Activities is a challenge for users.

Most closely related to our work is a study of ActivityExplorer [3], an ACC system organized around flexible threaded collections of diverse media objects (e.g., files, chats, tasks). A deployment with 33 researchers and interns over the course of 3 months provided an initial validation for ACC as a design paradigm. Muller *et al.* [10] found that interns appropriated the system in surprising ways, e.g. as a community forum. We extend this line of work with a deeper exploration of emergent use patterns of a production-quality ACC system by knowledge professionals, who adopted the system of their own accord (i.e., our users were not recruited). Balakrishnan *et al.* [1] have recently examined the adoption of this same ACC system in the context of other tools available to teams. In contrast, we focus on another pervasive open question in activity-centric work: How do users successfully structure their work as Activities? We address this through an empirical investigation of active users of one ACC system.

### 3 Overview of Lotus Activities: An ACC System

Lotus Activities is a Web-based service. On the entry page the user is presented with the following text explaining the system:

*Organize all your people, work, and collaborations associated with a business activity in a single place. Share files and web links, post messages, track to-dos, and manage your deadlines with Activities. Access the web-based dashboard from anywhere, or use plug-ins to extend your existing collaboration tools.*

Each Activity is described by a title, a set of tags, a list of members, an optional due date, and content entries posted by the members. To initiate a new Activity the user only has to provide a title. Members can be added at any time: "owners" who have complete access; "authors" who can add entries but not edit those of others; and "readers" who can look but not contribute. Activity entries can be of various types: basic text posts, to-dos, and threads of comments. Any entry can have attachments, links, and tags. Only the

members of an Activity have access to its contents, unless the Activity is made “public” to all Lotus Activities users. Figure 1 shows a screenshot of an Activity.

After logging in, the user sees a “dashboard” page, listing her current Activities. To manage the list, the user can sort, view by tag, mark as complete, or hide any Activity. The user can also view a combined list of all to-dos across all Activities. Users can send email notification when adding or editing an entry. To organize entries, users can group them into sections (like folders). To enable work

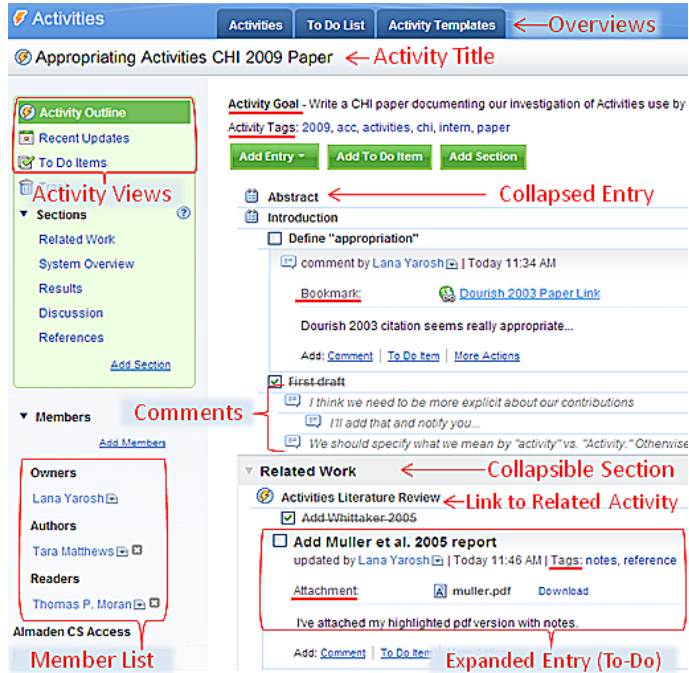


Fig. 1. Screenshot of the web interface to an Activity.

reuse, any Activity can be converted to an Activity Template. When creating a new Activity, a user has the option to instantiate a Template. This new Activity will have the structure and content of the parent Template.

A total of 38,719 Activities (containing at least one entry) have been created by employees of a large, global company since May 2006. The average activity has 3.4 members ( $SD=9$ ) and contains 12.1 entries ( $SD=29.9$ ). There are 31,913 users who are members of at least one Activity. Most users are members of 10 or fewer: 7,286 users have 1 Activity; 7,873 have 2 to 5; 20,073 have 6 to 10; 2,234 have 11 to 50; and 296 have 51+.

About a third (33.5%) of Activities were modified at least once during the course of our study (May-July, 2008), indicating a large number of in-progress Activities, as well as a large history of completed (or abandoned) work. We conducted an in-depth qualitative investigation of how 15 active users employed Lotus Activities for their everyday work.

## 4 User Study Methods

We conducted interviews with 15 users and analyzed the content of 120 publicly-shared Activities to investigate patterns of use and strategies for appropriating Lotus Activities.

**Table 1.** Demographics and Lotus Activities use of each participant.

Code	Gender	Formal Job Title	# Activities	# Entries
P1	M	Techn. Enablement Spec.	61	999
P2	M	Technical Sales Manager	13	193
P3	M	Certified IT Specialist	23	570
P4	M	Software Developer	80	1579
P5	M	Software Sales Leader	17	281
P6	M	Sales & Distr. Consultant	45	826
P7	M	Technical Sales Spec.	17	212
P8	M	Support Ops. Analyst	16	76
P9	M	Research Staff Member	43	230
P10	M	CIO Architect	25	408
P11	F	Client Relationship Rep.	226	3559
P12	M	Senior IT Specialist	11	127
P13	M	DB Support Team Leader	14	91
P14	M	Software Architect	83	173
P15	F	Level 3 Service Specialist	73	649

#### 4.1 Participants

We focused our investigation on highly *active users*, as they represent workers who have appropriated the system in ways that are useful to their work. We solicited by email users who owned the most public Activities or authored the greatest number of entries in these. We continued interviewing users until we reached a point of data saturation (as defined in [12]). These participants represented the variety of job roles, work locations, and experience levels typical of their company (see Table 1). To understand how these users structured and divided their work in Lotus Activities, we studied their public Activities through a content analysis and their private Activities in interviews and system tours.

#### 4.2 Interviews and System Tours

We conducted 1-hour, semi-structured interviews by phone. Each participant was asked to provide an overview of their job responsibilities and current projects. We asked about their initial motivation for adopting Lotus Activities and current system use. Finally, we asked users to show us a few of their Activities using web-conferencing software and to discuss how they organized and managed their work in the system. These tours allowed us to look at Activities that were not publicly shared. We analyzed the interviews using a grounded theory [12]. Distinct themes began to emerge after two independent passes of open coding and memoing followed by axial coding.

**Table 2.** Activity patterns found, indicating the percent of the 120 public Activities for which the pattern was present (some Activities contained multiple patterns) and the number of participants (out of 15) who took part in the Activity of this pattern or discussed it in an interview.

Pattern	% Activities	# Participants
Set of Job-Specific Work Units	48 %	12
Create a Document	14 %	13
Organize an Event	9 %	10
Solve a Problem	48 %	9
Collect Information	19 %	13
Formalized Exchange	3 %	4
Personal Tasks	23 %	11
Team Space	5 %	11
Create a Tutorial	13 %	10

### 4.3 Content Analysis of Participants' Public Activities

Following the interviews, we created a list of all public Activities and Templates in which our participants authored entries or were listed as members. We did not look at any private Activities outside of the interview to preserve their privacy. Our participants had 152 public Activities/Templates, representing 22% of the 692 such Activities/Templates in the system. We excluded 7 Activities that represented blank or unedited instances of existing Templates and those where participant membership was in question. One public Template led to 25 public Activity instances. Including all 25 would have incorrectly overrepresented this type of Activity, so we only included the original Template of this set in the analysis. We analyzed the remaining 120 Activities/Templates to complement the interviews as a source of evidence on the participants' patterns and strategies of use. We performed initial open coding, focused on the patterns of use and strategies evidenced in the structure and content of these Activities and Templates. In the process of axial coding, we found that the use patterns evidenced in the interviews and the content analysis converged. We proceeded to assign one primary and up to two secondary pattern codes to each of the 120 public Activities/Templates.

## 5 Results: Observed Patterns of Use

In the course of our analysis, we observed that Lotus Activities users developed specific structures and uses for the generic Activity construct. Our interviews and content analysis revealed nine patterns of use, summarized in Table 2. The patterns were informed by our participants' public *and* private Activities, so they represent the types of patterns employed by active users, overall. As these patterns emerged, we considered the open question of how users go about dividing and structuring work into Activities. This section

provides the empirical groundwork needed for us to revisit this question in the discussion section, focusing on describing the patterns and strategies of use. We discuss the most interesting patterns individually, grouping others with similar structure.

### 5.1 A Set of Job-Specific Work Units Pattern

The “set of job-specific work units” pattern was characterized by a set of identically-structured Work Unit Activities, with a single Activity representing an active unit of work for which the user is responsible (e.g., a customer account, a software version). Multiple Work Unit Activities of the *same type* are often active at once, and they are used to track status and store content. P11, a sales representative, had over 50 active Units, each representing an account containing different sections for brands and opportunities. Other examples included work requests (P8, see Figure 2), proof-of-concepts (P5), sales plans (P2), customer reports (P15), invention disclosures (P1), and product-client pairs (P12).

Highly specialized tools have arisen to support specific types of common, repeated activities. For example, CRM tools like Sales Force<sup>1</sup> help manage customer accounts, and development tools like Rational Jazz<sup>2</sup> help users coordinate software debugging tasks. Lotus Activities does not replace these tools for knowledge workers, but rather supplements them with a space for ad-hoc collaboration and everyday planning. For example, P11 continues to use the company-prescribed CRM tool (Siebel) to account for her work, but also uses Lotus Activities to define a structure for each account that better matches her everyday needs:

*I still definitely use Siebel, because we have to. But, like most CRM systems, they are for reporting only... It's great for management, but it's not so great for the sales reps, who have to track their own opportunities and their own communications with the client.*

Users transferred work to Lotus Activities because they saw their current tools as inadequate in some way, borrowing structure from those prior tools. Participants cited ease of maintaining and sharing Unit status and resources as the main reason for adopting Lotus Activities. Before Lotus Activities, P11 supplemented Siebel with a complex combination of file folders, email folders, and spreadsheets for each account. Transferring her organizational system to Work Unit Activities allowed her to consolidate all of this information and easily keep team members updated of changes to accounts through notifications. Starting with the established structure in Siebel made it easier to decide on an initial Activity organization then tailor it to specific needs.

A common strategy users employed to create Work Units was to design a Template to formalize the structure and process for a single Unit. Because the Work Units were persistent structures to which users referred often, they were willing to spend more time organizing the information. Dividing work into multiple Activities required developing strate-

---

<sup>1</sup> [www.salesforce.com](http://www.salesforce.com)

<sup>2</sup> [www.ibm.com/rational/jazz](http://www.ibm.com/rational/jazz)

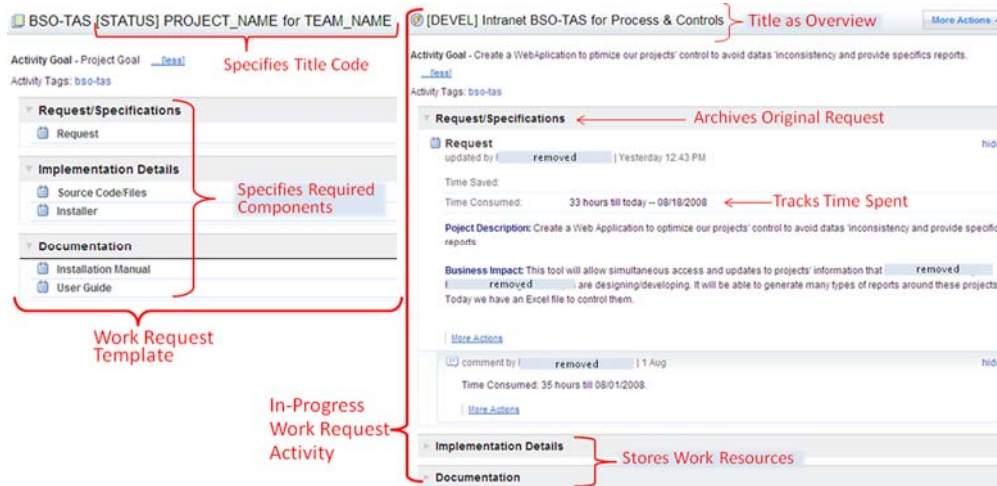


Fig. 2. Template for a work request work unit and an in-progress Activity of this type.

gies for understanding the current status across Units. Five participants developed conventions for including status codes in Activity titles, making status visible from the list of Activities on the dashboard.

Transferring process knowledge to others was another advantage of structuring Activities as sets of Work Units. Users appreciated that Activities exposed the *process* of carrying out the activity to new teammates by saving to-dos, intermediate drafts, and comments, as P15 related:

*I had to show [a new team member] how to properly do the [customer report]... I just pointed her to the Activity... I didn't have to do a lot of explaining of the [customer report] process. It was all kind of self-explaining.*

The Set of Work Units pattern demonstrates a non-trivial translation of work activities into the system, splitting multiple threads of a job responsibility across many highly-structured Work Unit Activities.

## 5.2 Complete Work Activity Patterns: Documents, Events, & Problem Solving

The next three patterns—Create a Document, Organize an Event, and Solve a Problem—are Activities bounded by a clear goal and a well-defined endpoint. Thirteen users employed Lotus Activities as a space for coordinating the creation of a document, such as a report or a presentation. Ten participants organized events like workshops, business conferences, and sales demonstrations. Nine users had Activities aimed at resolving a problem in software development, customer issues, and policy decisions. These patterns all involved taking notes or collecting ideas/information, brainstorming, and/or making deci-



sions. Having a shared space for coordinating production, planning, or problem solving was important to users because it helped improve awareness of the activity's status and reduce email volume during the process, as P12 explains about a document creation Activity: "...rather than sending these different files around, I can just stick it into an Activity and send a link around and... then everybody has access to the most current version."

### **5.3 Partial-Activity Patterns: Collecting Info & Exchanging Formalized Info**

The next two patterns, Collect Information and Exchange Formalized Information, served to manage content intended as a resource for a larger activity (which may or may not have been represented in Lotus Activities). The Collect Information pattern involved using an Activity as a space to collect sources of information (text, links, attachments, etc.) on a single topic. Used by 13 participants, Collections ranged from roughly task-focused (e.g., use cases for a software product) to loosely connected around a topic (e.g., list of Web 2.0 development resources). A common strategy for generating Collections was to add ideas in an ad-hoc manner, as P15 noted: "I just started putting it here whenever I learn something... Activities is a good place for me to use kind of as my brain."

In contrast, Formalized Information Exchanges, which involved requesting and managing information from multiple people, were planned and usually made into Templates. These Activities, used by 4 participants, included instructions requesting specific information. For example, P14 created a Template to collect demo proposals for a conference he was organizing, asking interested users to instantiate the Template and fill it out.

### **5.4 Cross-Activity Overview Patterns: Personal Tasks & Team Spaces**

Participants found it valuable to create or maintain overview perspectives that spanned multiple Activities. Managing Personal Tasks and Team Spaces represent two views of work that cut across Activities: one unifying individual work, the other unifying work across a team. About two-thirds of users reported having Activities which only they could access, containing personal to-dos. These Personal Task Management Activities were developed by users who wanted the Lotus Activities to-do list view to contain a complete overview of their commitments.

Team Space Activities sought to unify content related to managing a team to provide an overview across projects and simplify bringing new members up-to-date. They were often organized into sections or discussion threads that included elements like meeting minutes, project timelines, accomplishments, and links to other team Activities. They also often included elements of other patterns, such as solving a problem, planning an event, or creating a document. These Activities represented a team-level view of past and ongoing work, usually maintained with considerable effort by the team's manager or coordinator.

## 5.5 Teaching & Reuse Pattern: Tutorials

While other Activity patterns focus on performing work, the Tutorial pattern was created explicitly to guide other users (10 participants had these). We saw two types of Tutorials: those intended to be read for educational purposes, and those intended to be instantiated to guide users through a process. P15 described the former: “I take entries from several different problems... [so that] others can now see how to solve this problem. Instead of coming back to me with questions, they can just refer to that.”

One strategy for creating a new Tutorial was copying a successfully completed Activity. Another strategy was reproducing the steps in existing policy guides or paper checklists as a Template. Tutorials created as Templates provided an advantage over static guides or paper checklists: they could be used to create an independent Activity and tailored to the needs of the user carrying it out, as noted by P7: “I wanted to give people the freedom to change [their Activity]... and let people have more control over it.”

It was difficult to make changes to tutorials over time, because instantiated Activities do not reflect Template updates. P12 was anxious about this limitation and “... felt like I really had to get it right before I put it out for people to use.”

Once content was created, authors often publicized their Tutorials by posting links to them on wikis, blogs, and other resources that were indexed for search across the enterprise. This step was necessary, because the search function within Lotus Activities was too unreliable for users to effectively discover Activities and Templates.

## 5.6 Relationship and Evolution between Patterns

Our content analysis enabled us to observe relationships between patterns commonly found together in the same Activity. Creating a Document and Collecting Information were often components of other patterns, such as Work Units, Team Spaces, Solving Problems, and Planning Events. Work Units and Team Spaces were most likely to include other patterns, because they served as organizing Activities.

Patterns that were loosely-organized and unbounded by time (e.g., Information Collection or Personal Task Management) could evolve into more structured Activities. For example, P7 described an Activity created with a client to “share information” and “reduce the amount of email” (an Information Collection) that, with time, began to represent a history of work with the client that included meeting notes and to-dos (i.e., a Team Space): “We brainstormed some work items... We would use the Activity to track them and define actions and dates so that we would end up with a kind of project timeline.” Next, P7 planned to remove client-specific information, make a Template of this Activity, and reuse it. Thus perhaps, we were seeing the evolution of a new Work Unit.

Tutorials were patterns that commonly evolved from other Activities by rearranging or merging content, adding directions, and/or making Templates from them. For example, P3 created a Tutorial guiding soon-to-be parents through company paperwork by making a

Template from a Personal Task Management Activity, where he had collected his own to-dos and resources as he was going through the process.

Activities often represented multiple patterns and could evolve over time, so deciding on a structure for an Activity was not about just identifying the “best” pattern to use. In the next section, we use these results to return to our driving questions, discussing how knowledge workers go about dividing and structuring work.

## 6 Discussion

In this section, we discuss how our participants appropriated the Activity construct, propose three strategies users employ to structure Activities, and identify emergent behaviors in their use of the system. We also highlight implications for design of ACC systems.

### 6.1 What is an Activity?

As a unit of analysis, an activity is defined as a coordinated set of actions by people towards a common objective, mediated by tools and subject to situational constraints (e.g. [6]). But in work settings, knowledge workers have objectives at different levels of granularity, some objectives being in service of other objectives (e.g. creating a demo as part of preparing for a review as part of managing a project). However, the Lotus Activities system does not represent in-service-of relationships between Activity constructs. Thus, the empirical question is: what granularity of objectives do people associate with Activity constructs. Our overall finding is that the Activity construct is used to support objectives at various levels of granularity. For some usage patterns, such as Creating a Document, Planning an Event, or Solving a Problem, Activity constructs clearly correspond to concrete, achievable objectives. On the other hand, the Work Units pattern reveals a single objective of managing all the work represented with multiple Activity constructs. Still other patterns, like Collecting Information or Exchanging Formalized Information, are in service of larger objectives represented as Activity constructs. Overview patterns—managing a Team Space or Personal Tasks—bring together the work of many objectives within a single Activity construct. Finally, multiple usage patterns may occur in the same Activity construct, which may evolve from one pattern to another. Thus, the analytic concept of an activity does not map in a simple way to the way people organize their work in an ACC system. Our findings identifying these patterns show that active users of Lotus Activities *were* able to identify effective divisions of their work in an ACC system, which confirms the value of a flexible, generic Activity construct. However, P13 emphasized that “the thinking is the startup cost” for appropriating Lotus Activities, and this cost forms the main barrier for new users. We believe that ACC system designers could aid novice users by providing explicit support for the common patterns identified in this study.

## 6.2 Structuring Activities

We have identified some common patterns of use for Lotus Activities, but users often structured their instances of these patterns in different ways. This presents a challenge to designers trying to provide guidance for defining Activities. However, users followed a common process in structuring their Activities: (1) they seeded structure from previous tools, (2) they evolved existing Activities, and (3) they reused successful ones. Understanding this process leads to design implications for ACC systems.

*1. Seeding.* Users seek out new tools when current ones fail in one way or another. Thus a user seeking to structure a piece of work in an ACC system is often transferring that work from another tool. We saw that users leveraged the structure of their old tools to seed new Activities. This strategy may explain why Work Units in a similar domain had different structures—users were migrating from *different tools*. P11 used an “account” as a Work Unit Activity to track sales opportunities, because this was how she previously structured her file and email folders. On the other hand, P12 transitioned from internal documentation forms that needed to be filled out for each customer-product pair, so his sales opportunity Work Unit Activities mirrored this structure. Users continued using company-mandated, team-specific, and generic tools in parallel with ACC, so it is critical for the designer to integrate ways to transition data between the previous tools used for the activity and Activity constructs of an ACC system.

*2. Evolving.* While initial Activity structure may be suggested by a prior tool, the Activity may still need significant modification to address a user’s needs, requiring incremental evolution over time. For example, P8’s team wanted to use Lotus Activities to document work requests in more detail than the spreadsheet they currently used (see Figure 2). They structured each row of the spreadsheet (each request) as a new Activity with the spreadsheet field data (project name, status, and target team) becoming the Activity title. The Activity started out with no entries. The team iterated multiple times on the internal structure, adding entries prompting for certain information and organizing them into sections. We were able to see this process by looking at the older versions of the work requests. Since evolution and reuse of activity-related documentation is a common practice [13], supporting this process of “incremental formalization” [14] can be one of the greatest benefits of an ACC system. To do so, ACC systems should provide effective tools for reorganizing existing Activities. Additionally, Activity Templates must be able to selectively propagate changes to existing Activity instances in order to support incremental evolution.

*3. Reusing.* It may take significant effort for a user to achieve a successful structure for a given Activity; but once it is identified, it can be easily reused. This explains why our users were likely to have many Activities with the same structure (i.e., Work Unit Activities). Tracking status across sets of Activities currently presents a challenge for Lotus Activities users, but designers can leverage the structured and repeated nature of these Activities to create overview perspectives, e.g., as visualizations, filtered or aggregated views, or overview Activities.

### 6.3 Emergent Behavior

Users exhibited emergent behaviors to realize unexpected benefits of using an ACC system: sharing knowledge and reflecting on work. In identifying these emergent behaviors, we suggest features to enhance the sharing and reflection benefits.

Activity constructs represent records of how activities were *carried out*, which can be educational for others. Lotus Activities lowered the overhead of creating sharable content, since it was created in the process of doing work. Unlike workflow systems, ACC systems can provide rich accounts of work as it was actually accomplished rather than as it was prescribed [15]. Among our participants, successful Activities were used to train new members, transition work between employees, and share ideas with other teams. However, we also saw that currently users could learn about these resources only through explicit advertising efforts (e.g., posting the link to a blog, mentioning in a presentation, etc.). To capitalize on knowledge sharing and work reuse opportunities, an ACC system should increase “discoverability” by providing comprehensive search facilities and by highlighting successful tutorials, e.g. with social software techniques such as ratings.

Some of our users found that the act of creating content in the ACC system helped them reflect on their work. Six users said that capturing work as they did it made reporting current status and accounting for their time easier. ACC systems should help users create accounts of their time and work progress for reflection or reporting purposes. Four users found that combining information from multiple Activities was a way to reflect on content that assisted them in identifying solutions to problems. This kind of improvisation is consistent with practices of other successful knowledge workers [16]. To support this behavior, reorganizing content across many Activities should be facilitated.

## 7 Conclusion

For any ACC system user, developing effective ways to structure Activities will be a challenge. The usage patterns, strategies employed, and challenges faced by active users have revealed some ways to mitigate this adoption barrier in ACC systems. First, the designer could aid novice users by providing explicit support for the common patterns identified in this study. Second, the designer could aid in the transition to the ACC system by suggesting ready-made structures similar to those present in the existing tools favored by the user for targeted activities. Third, the designer could make it easier to continually improve and reuse the structure of Activities by supporting lightweight reorganization and by allowing selective propagation of Template changes to instantiated Activities.

We have shown that ACC systems have the potential to support knowledge workers by providing a shared Activity constructs for coordinating work units, producing deliverable outcomes, gathering information, and sharing knowledge. By investigating both the successful usage patterns and the challenges users faced while structuring their work, we believe that future ACC systems, as well as future versions of Lotus Activities, will over-

come one of the major adoption barriers for ACC systems: helping users understand what an Activity is and how it can help them be more effective workers.

**Acknowledgments.** We thank: James Lin, Jeffrey Pierce, and Aruna Balakrishnan for advice during this research; the Lotus Activities team—Miguel Estrada, David Brooks, Marty Moore, Bob Stachel, Suzanna Minassian, and Scott Prager—for ongoing support; and Michael Muller and Gregory Abowd for comments on drafts of this paper.

## References

- [1] Bardram, J., Bunde-Pedersen, J. and Soegaard, M. "Support for activity-based computing in a personal computing operating system," *Proc. of CHI*: 2006, pp. 211-220.
- [2] González, V.M. and Mark, G. "Constant, constant, multi-tasking craziness: managing multiple working spheres," *Proc. of CHI*: 2004, pp. 113-120.
- [3] 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," *Proc. of CHI*: 2004, pp. 375-382.
- [4] Christensen, H.B. and Bardram, J. "Supporting Human Activities - Exploring Activity-Centered Computing," *Proc. of UbiComp*: 2002, pp. 107-116.
- [5] Kaptelinin, V. "UMEA: translating interaction histories into project contexts," *Proc. of CHI*: 2003, pp. 353-360.
- [6] Kuuti, K. "The concept of activity as a basic unit of analysis for CSCW research," *Proc. of ECSCW*: 1991, pp. 249-264.
- [7] Dourish, P. "The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents," *Proc. of CSCW*: 2003, pp. 465-490.
- [8] Voida, S. and Mynatt, E.D. "It feels better than filing: everyday work experiences in an activity-based computing system," *Proc. of CHI*: 2009, pp. 259-268.
- [9] Bellotti, V., Thornton, J., Chin, A., Schiano, D., and Good, N. "TV-ACTA: embedding an activity-centered interface for task management in email," *Proc. of Conference on Email and Anti-Spam*: 2007.
- [10] Muller, M.J., Minassian, S.O., Geyer, W., Millen, D.R., Brownholtz, E. and Wilcox, E. "Studying appropriation in activity-centric collaboration.," *International Reports on Socio-Informatics*, vol. 2, 2005, pp. 50-58.
- [11] Balakrishnan, A., Matthews, T. and Moran, T.P. "Adopting an Activity-Centric System in an Ecology of Workplace Tools," IBM Almaden Research Center: 2009. *Working Paper*.
- [12] Seidman, I. *Interviewing as Qualitative Research: A Guide for Researchers in Education And the Social Sciences*, New York: Teachers College Press, 1998.
- [13] Moran, T.P., Matthews, T., Vega, L., Smith, B., Lin, J. and Dill, S. "Ownership & Evolution of Local Process Representations," *Proc. of INTERACT*: 2009. To Appear.
- [14] Shipman, F.M. and McCall, R. "Supporting knowledge-base evolution with incremental formalization," *Proc. of CHI*: 1994, pp. 285-291.
- [15] Dourish, P. "Process descriptions as organisational accounting devices: the dual use of workflow technologies," *Proc. of GROUP*: 2001, pp. 52-60.
- [16] Schön, D.A. *The Reflective Practitioner*, Basic Books, 1983.