Unified Activity Management: Explicitly Representing Activity in Work-Support Systems

Thomas P. Moran
IBM Research
Almaden Research Center, San Jose, CA, USA
tpmoran@us.ibm.com

Most computer support of collaborative work is organized around tools for communicating and for sharing materials. Email is the prime communication tool, and it is probably the most used tool. It is reliable and familiar, but dealing with the flood of incoming email and organizing the collection of emails are major problems. Directories and "places" are the most common tools supporting the sharing of materials. These are useful, but have the overhead of setting them up and keeping them organized as they fill up. (In fact, most people share materials by attaching them to email.) Other tools support individual work – document creation, web browsing and searching, calendaring. All these tools are where most people spend most of their on-line time. These tools are independent and optimized to specific functions. However, the work people do, by themselves and with others, cuts across these tools. Thus it is left to people, both individually and collaboratively, to devise practices to coordinate the tools, which is both non-productive and error-prone. The hypothesis of this paper is that technology should support people focusing on to the work they are trying to accomplish and that the foundation of this kind of support is an explicit representation of the work.

Activities

People organize and think of their work as *activities*, which are personal objectives or social agreements to collaborate. Activities are supported in limited and ad hoc ways in various tools – todo lists, task lists, calendar meetings. There are tools, such as workflow and project management tools, that directly deal with managing work by formalizing an idealized conception of the work. They predefine the work and are used to control and account for it. These tools are heavyweight and rigid.

Most people do most of their work using lightweight tools, which do not provide adequate activity management support. A goal of much current collaboration research is to develop collaboration tools to support lightweight, on-the-fly, situated activities as well as tools to manage multiple activities. These explorations tend to focus on adding task structure to email (e.g., Bellotti et al, 2003; Whittaker, 2005) or capturing and bundling computational events into activities (Bardram 2005; Kaptelinin, 2003; Herlocker, 2005). These efforts are bottom-up approachEs to supporting activities, starting with the concrete actions people do and providing some structure to organize them.

In the *Unified Activity Management* (UAM, 2005) project at IBM Research we started from a somewhat different perspective. We hypothesized the need to explicitly represent activity as a computational construct and to provide an infrastructure to support it. This should be a generic representation, so that it can be used to integrate the variety of tools that people use. Thus we call it *Unified Activity*, since it is a representational schema that can unify different representations by capturing the essential elements of the idea of an activity.

Modes of Working

To understand where an activity construct fits in the set of system tools, we use a framework that characterizes how people work with these tools, shown in Figure 1. Computational tools afford different modes of working. We distinguish three modes.

The **Action** mode of working is the most concrete. Actions are limited pieceS of well-defined work at well-defined points in time. Actions are diagrammed as a line of small a's in Figure 1. Prototypical examples are writing/sending an email, making a phone call, editing a document, browsing the web, etc. Usually actions are by individuals, but they can be jointly carried out, such as an instant messaging session. Often there is an artifact resulting from the action. Some actions are online and recordable, but some are ephemeral, such as a hallway chat. There is limited structure between actions. There is temporal structure, so actions can be arranged in a timeline (as in Figure 1). There is also response structure: actions are often triggered by other actions (the arrows in Figure 1), for example replying to an email. What is characteristic of the action mode is that *people are spontaneous and reactive*. People just keep doing them without much thought. This is the value of the action mode – work keeps progressing. The problem with the action mode is that actions for many different purposes are intermixed in time (Czerwinski, Horvitz, and Wilhite, 2004; Gonzalez and Mark, 2004), and it is difficult to organize the actions or to see where the work is going.

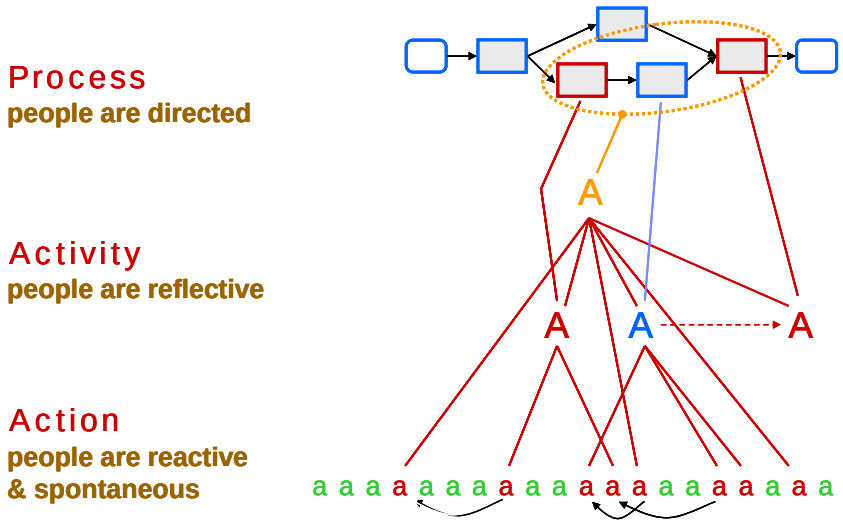

Figure 1. Modes of People Working

The **Process** mode is at the other extreme. Processes are preplanned by other people (process designers). They are usually represented in a flow structure, as shown in the top of Figure 1, and controlled by a computational system. People are notified when there are process steps that they must do. Thus *people are directed to take actions*; the initiative is with the processes, not the people. Processes can be conditional, but their flexibility is limited to the logic that can be programmed into them. Processes are oblivious to any actions taken outside the prescribed steps.

The **Activity** mode is in between. People set up, organize, coordinate, plan, and report on their work: activity management. The main characteristic of this mode is that *people are reflective about their work*. An explicit computational representation of the work can serve as the object of reflection. Figure 1 shows activities as capital A's in a hierarchic structure. These are computationally-represented activities. They can be articulated into subactivities to an arbitrary degree of decomposition. For practical purposes, activities are articulated only to the extent that is helpful for managing them. Activities provide contexts for organizing actions into meaningful structures (the lines between actions and activities in Figure 1). Activities are larger in scope than actions, and actions can be seen as the "terminal nodes" of activity structures. In the other direction, activities give the context that relates actions to processes. And activities also give visibility and rationale to processes.

Shared Checklist

What does an activity structure look like to the people involved? We have explored many different specific user interfaces, but a common metaphor for many of them is activity as a *shared checklist*, such as the example is shown in Figure 2. The internal representation of the activity of "organize workshop" is shown in the upper left of Figure 2. Tom is responsible for the overall activity. It shows that five subactivities are articulated, with different people doing each of them. (More subactivities can be

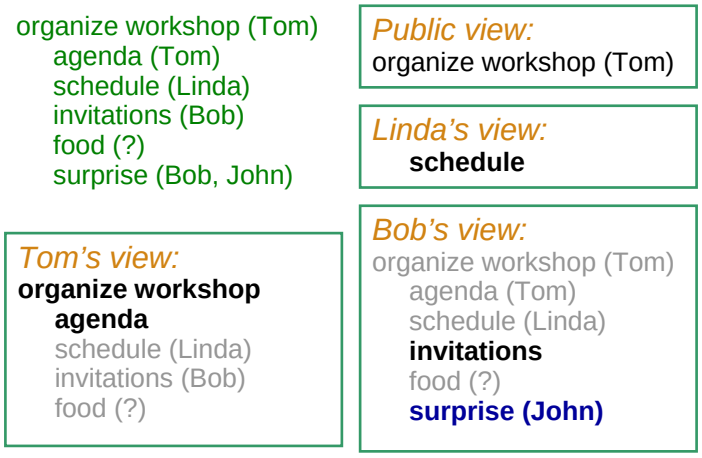

Figure 2. A Shared Checklist of an Activity with the different views of different participants

added, and the activity structure can be changed, as the activity proceeds.) Each participant has a view of the shared checklist that is appropriate for their involvement. Tom's view highlights the activities he is involved in. Linda has a very restricted view, since she has a limited role. Bob's view shows a subactivity that only he and John can see; at some point they can share it with the others. Finally, there is a view that tells the public that a workshop is being organized by Tom, but none of the details.

Unified Activity Metamodel and Metadata

Now that we have seen the shared checklist as one manifestation of activity structures to the participants, we can look in more detail at the internal representation of an activity. Unified Activity is a metamodel for describing activities. The representation can be seen as a graph, as shown in Figure 3.

An activity is represented as an association of properties and as relationships to other entities. An activity has informal statements describing the objective and status; the rest is relationships. An activity involves various Actors in roles (e.g., responsible, advisor) (Harrison, Cozzi, & Moran, 2005). It uses various resources (reference materials, tools, and other artifacts) and produces results (which can be artifacts. An activity is also bound by Events, such as a meeting being a deadline. An activity has an associated action log (cf. Figures 1 and 3), where actions are instances of Operators which change the activity representation. Thus, an activity representation associates actors (people), resources, and events with a description of how they work together. For example (see Figure 2), Linda schedules a workshop, with Tom's advice, using a calendaring and reservation system, and producing a scheduled event.

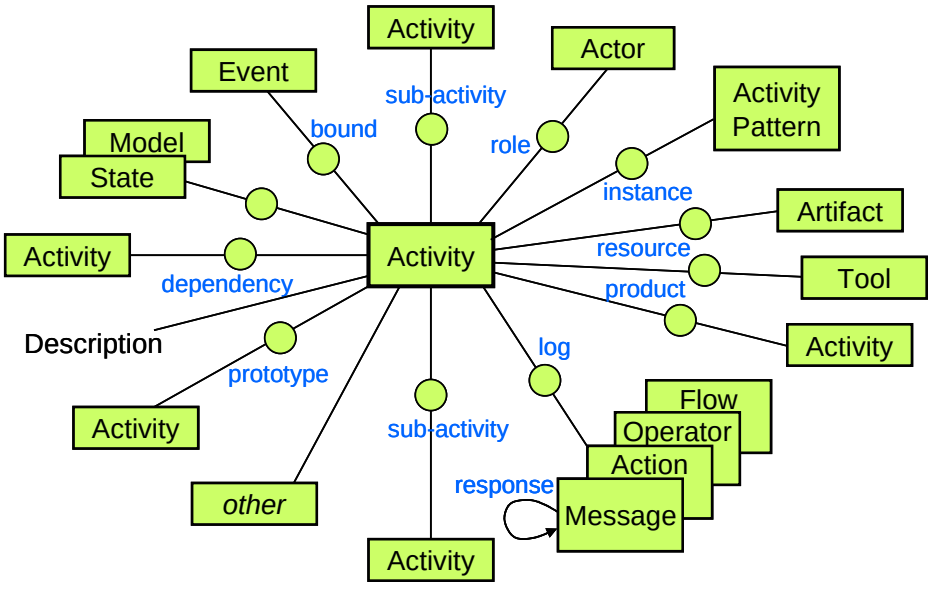

Figure 3. Unified Activity Representation as a graph

Activities do not exist in isolation, but are related to other activities in many different ways. Figure 3 shows a few activity-activity relationships. The principal relationship is sub/super-activity. An activity can be decomposed into subactivities, and an activity can be in the context of one or more superactivities. An activity can be dependent (e.g., as consumer) on another activity (as producer). An activity, such as a meeting, can produce other activities (action items).

A major benefit of representing activities is that they are then re-usable. An activity can use a previous activity as a prototype, either by reusing the structure of the previous activity or by borrowing actors or resources. An Activity Pattern is an activity description that has been refined, generalized, and parameterized, as a representation of best practices. So, new activities can be created as instances of patterns.

Thus we see that Activity is represented as a web of relationships to other entities and to other activities. It is the "glue" that ties together information that is already represented in other systems (email, calendars, content repositories, personnel directories, etc.) around the generic activity semantics. We encode Activity descriptions in RDF, the "semantic web" standard of the W3C (RDF, 2005). This is an ideal platform to provide the flexibility, extensibility, and data integration necessary to support the inherent variability and adaptability of the activities. And we are using OWL (2005) to formally specify the ontology of unified activity to explore it as a standard.

To fully realize the concept of activity as a central organizing principle of work support, it must be implemented as a part of the infrastructure of work support, not just as another application with its own data type. Thus, the central architectural component is the RDF-based *Activity Metadata Repository*, which maintains the activity representation and provides APIs (from formal web services to less formal REST-based http protocols). Many different applications will use these APIs to expose activities in various forms and settings, not only as checklists, but also as email organizers, "how-to" aids, management "dashboards," etc.

Nature of Unified Activity

While Unified Activity is a metamodel for describing informal work, activity descriptions are not considered to be models of the activities that people carry out. Rather, *activity descriptions are social artifacts that support the management and execution of work*, that is, activity descriptions are themselves resources for collaborative work. Activity descriptions provide a *common ground* (Whittaker, Moran, & Farrell, 2005) that supports the coordination, *awareness* (Carroll, 2005), and *social legibility* (Dourish, 2005) of collaborative activities.

In order to be effective as social artifacts for work, activity descriptions do not have to be formal and complete. In fact, we should expect them to be rather "ragged," for activity descriptions are created pragmatically, balancing costs and benefits. Activity descriptions that are created on the fly are loose, terse, and partial. Those that are instantiated from Activity Patterns will have a head start. In either case, they will evolve and change to adapt to the circumstances of the particular work situations.

Activity Patterns are social artifacts with a larger scope. They embody effective practices and useful advice. While each activity description that starts as an instance of a Pattern can be situationally adapted, Activity Patterns have a different cycle of evolution. How patterns evolve is a matter of social arrangements, incentives, and organizational culture (e.g., Nardi, 1993).

Unified Activity and Activity Theory

Activity is a central theoretical construct in HCI/CSCW research and theory. This is the case not only for Activity Theory (e.g., Nardi, 1996), but also for distributed cognition (e.g., Hutchins, 1994), situated action (Suchman, 1987), and others (e.g., Carroll, 2003). It is interesting to compare how a Unified Activity, a practical construct, is necessarily both similar to and different from theory. For this discussion, let us focus on Activity Theory (AT) as the point of comparison for Unified Activity (UA).

UA was created to fill a missing gap in work-support systems. The term "activity" was chosen carefully – out the zeitgeist of CSCW research – to characterize what work-support should support. UA was not specifically inspired by AT, but UA and AT have many similar concepts. AT provides an analytic account of real human collaborative activities, giving us distinctions and concepts with which to "parse" what we observe people doing. Although UA is meant as a schema for producing social artifacts, its metamodel (Figure 3) can also be seen as providing concepts to "parse" what people do. We discuss several examples.

Activity and Consciousness. AT sees activity as conscious behavior. UA is also concerned with conscious behavior. Further, UA provides activity descriptions as artifacts for *reflection* on activity. In AT terms, UA provides for *externalization* of people's subjective organization of what they are doing.

Defining Activity. AT sees activity as sustained actions towards a (subjectively defined) *objective*. The notion of an objective is broad, encompassing pretty much any motivation that gives direction to the activity. UA also considers that activity is motivated, but does not formalize this aspect, except to give a person the option of describing an objective. UA has a different slant, seeing activity as a coherent set of actions and potential actions. In UA the concept of *coherence* plays a more defining role. Coherence is based on the *need to manage* a set of actions as an activity. There are different kinds of coherence, such as:

- *Conceptual coherence* manages actions by some concept (a goal or objective).
- *Contextual coherence* manages actions around a context, such as a meeting, where disparate actions are brought together to take advantage of the meeting context. Another example is email (people say they "do email"), which is a tool context in which actions are taken because they are convenient to while being in the context of the tool. AT sees this as activity mediated by a cultural tool.
- *Categorical coherence* manages actions of the same type, even if otherwise scattered. For example, a set of trips can be managed for purposes of budgeting.

- *Role coherence* manages actions around a job role, for example, all the things one has to do to function as a teacher.
- *Perspective coherence* manages actions around a particular point of view on how an activity is organized. For example, reorganizing a project in mid-stream is, in effect, creating a new activity. A complex project might also be usefully managed from multiple perspectives.

Activity and Object. AT sees activity as object-directed, in that the objective is often a concrete object. UA tends to stress the doing more than the product. UA allows several artifacts to be associated with an activity as resources. UA distinguishes different kinds of resources: those that are referenced (read only), those that are used (tools), those that are worked on (edited), and those that are produced (the outputs). We have observed that people often write down activities as nouns designating the things that are to be produced (e.g. "agenda"), where the verb (e.g. "create agenda") is either vague, obvious, or hard to specify (Muller and Wu, 2005). UA currently deals with the notion of an object by the *represents* relationship: and object represents an activity. (Note that the object in this sense of a representative can also be a person, e.g. "Dave" for "call Dave.") UAM prototype clients allow one to easily convert ("activate") an object into an activity: dragging a document icon into an activity region creates an activity with the document as representative. This allows the system to "wrap" activity metadata around any object. (Actually, there has been considerable discussion in the UAM project in trying to resolve the UA notion of an activity with the notion in a companion project, Activity Explorer [Muller et al., 2004], which views activities as shared document objects.)

Activity Hierarchy. AT sees activity as functionally structured in a hierarchy: activity, action, operation. Activity is oriented to high level motives, and action is goal-directed behavior in service of an activity. Both activity and action are conscious. An operation is an action that is routine and unconsciously carried out. UA sees activity in a different hierarchy: process, activity, action. UA is not concerned with the unconscious and so doesn't represent an operational level. Because it is a representation, UA is concerned with formal process representations and how UA relates to them. AT is not concerned with process in this sense. AT observes that behavior can shift between levels: operations can become actions, and actions can become activities. UA also supports similar shifts in its hierarchy: actions that need to be managed can be easily converted into activities, and activities can be evolved over time into patterns and into processes.

Activity vs Action. AT sees activity as the highest level motivation for people. Any behavior that is subservient to an activity is an action. Actions are goal-driven and can be structured in a goal hierarchically. UA, on the other hand, sees activities as being hierarchically structured, whereas actions are the lowest-level, goal-driven behaviors. UA sees actions as communicative and reactive behaviors, structured by temporal and *respond-to* relations (Figure 1). In UA activities can vary greatly in temporal scope, from minutes to months. They are all activities and can be related to each other as subactivities or supporting activities. The UA stance is that people can be motivated by different time horizons, and the basic structure of activity representations (Figure 3) is the same at different levels.

Activity as Context. AT is basically a theory of the context of human behavior. UA is a framework for providing context support for collaborative (and personal) work. It

does this by linking together people, resources, communications, and activity properties to give an individual a context for what to do next. But also it does this by relating activities to each other. Activities are not disjoint from each other. Activities exist in a network of related activities. Thus activities serve as context for each other. The main context-defining relation in UA is the sub/superactivity relation: an activity is carried out in the context of its superactivity. But activities do not only nest neatly in a strict hierarchy, they "overlap" to various degrees. A principal way UA represents this is by "multiple parenting" (as we loosely talk about it). Not only can an activity have multiple subactivities (supporting activities), but also it can have multiple superactivities (supported activities). Thus any activity can be carried out in multiple contexts. For example, preparing this paper simultaneously serves the contexts of the Activity Workshop at ECSCW, the travel planning for the trip to Europe, the publication strategy for the UAM project, and the semantic modeling activity of the UAM project.

Subjective vs Shared. AT, being a psychological theory, see activity as subjectively defined by the people engaged (the subjects). Each person has their own motivations, and so a collaborative activity is intersubjective – some kind of meshing of motivations. UA starts as a shared representation. When the participants have different views, then UA promotes negotiation of a common ground. Different people will have different involvements in different parts of the activity structure, and this is one way to represent their different motivations. We are exploring how much further we should go by allow personal views, annotations, and properties.

Development. A major tenet of AT is that activity undergoes constant development from practice and experience over time. UA also sees activities as evolving, and it provides the Activity Pattern as the main mechanism for supporting, capturing, and disseminating improvements with experience.

Can Activities be Explicitly Represented?

Analytic accounts of activity, such as ethnographies, are difficult to come by, requiring enormous effort from skilled practitioners. While practical descriptions are less demanding, they do require attention and effort. Most people prefer action to reflection. There is a reason they avoid representing their activity. Activities are elusive, ephemeral, tacit, dynamic, and often difficult to articulate and categorize.

Explicitly representing activities has benefits at many levels. For individuals, activity descriptions aggregate people, resources, and tools around activities; and thus they are "at hand." With Unified Activity, individuals can manage diverse activities from diverse sources. For group or team work, creating and maintaining a shared activity representation can aid understanding, negotiation, consensus, coordination, and activity awareness. Awareness can be beyond activity participants; there can be awareness by management, and awareness across teams so they can learn from each other. For the organization or enterprise, activity patterns are a way to capture best practices in a bottom-up fashion. Analyzing how activity instances deviate from patterns can be a source of organizational adaptation. Finally, activity descriptions can provide activity-based audit trails.

The potential benefits are powerful, but most of them are not immediate, which is a challenge. How can we mitigate the difficulties of representing activities? There are several simultaneous techniques:

1. Representing an activity should be as easy as jotting a post-it. Fluid user interfaces are needed.

2. Perhaps more important is that activity creation should be contextual – in the habitats of work. This means that activity interfaces should sit alongside or be embedded in the common applications, such as email. Common computational objects, such as documents, should be able to be "activated" by wrapping them in activities that represent the development and use of those objects.

3. Collaborative work often elicits reflection on activity to plan and coordinate, and activity descriptions are a natural side effect. Sharing activity descriptions means that not everyone has to do the work of managing an activity.

4. Activity descriptions will often start as instantiations from activity patterns. Good patterns give a head start on organizing recurrent work. The benefit is immediate.

5. Activities will be created by formal processes, in e-commerce and in corporate environments.

6. In the business environment, compliance with regulations requires better documentation of work, and activity descriptions can serve these requirements.

7. Some "smart" system capabilities could help organize work into activities. Various technologies, such as text and social network analytics and computational learning, can be applied to recognizing the coherence in actions and produce suggested activity descriptions. Much current activity-centered computing research is in this vein (e.g., Herlocker, 2005). This kind of capability is inherently noisy and thus must be carefully applied in a supporting role in activity management.

All of these factors must come into play if work support systems are to be more activity-centric. There has to be a network effect. Explicit activity objects must become a common interchange among different systems and between systems and people. Once this happens then the cost of creating activity descriptions will not be an issue – it will simply be the way one works.

Acknowledgement. Thanks to Michael Muller for valuable advice and critique of this paper.

References

1. Bardram, J. E. (2005). Activity-Based Computing: Support for Mobility and Collaboration in Ubiquitous Computing. *Personal and Ubiquitous Computing*, 2005.
2. Bardram, J. E. (2005). Activity-based computing – lessons learned and open questions. *ECSCW 2005, Workshop on Activity*.
3. Bellotti, V., et al. (2003). Taking email to task: the design and evaluation of a task management centered email tool. *CHI 2003*.
4. Carroll, J. M. (Ed.) (2003). *HCI Models, Theories, and Frameworks: Towards a multidisciplinary science*. Morgan-Kaufmann.
5. Carroll, J. M., et al. (2005). Supporting common ground in geo-collaboration. *ECSCW 2005, Workshop on Activity*.
6. Czerwinski, M., Horvitz, E., and Wilhite, S. (2004). A diary study of task switching and interruptions. *CHI 2004*.
7. Dourish, P., et al. (2005). Activities and two forms of social legibility. *ECSCW 2005, Workshop on Activity.*
8. Gonzalez, V., and Mark, G. (2004). Constant constant multitasking craziness: Managing multiple working spheres. *CHI 2004*.
9. Harrison, B. L., Cozzi, A., and Moran, T. P. (2005). Roles and Relationships in a Unified Activity Management System. *Group 2005*.
10. Herlocker, J. L. (2005). TaskTracer: An activity-oriented personal information overlay. *ECSCW 2005, Workshop on Activity.*
11. Hutchins, E. (1994). *Cognition in the wild*. MIT Press.
12. Kaptelinin, V. (2003). UMEA: Translating interaction histories into project contexts. *CHI 2003*.
13. Moran, T. P. (2003). Activity: Analysis, Design, and Management. *Symposium on the Foundations of Interaction Design*, Interaction Design Institute, Ivrea, Italy.
14. Muller, M. J., Geyer, W., Brownholtz, B. Wilcox, E., Millen, D. (2004). One hundred days in an activity-centric collaboration environment based on shared objects. *CHI 2004*.
15. Muller, M. J., and Wu, M. (2005). Multiple users' perspectives on collaborative activities in complex work. *ECSCW 2005, Workshop on Activity*.
16. Nardi, B. A. (1993). *A small matter of programming: Perspectives on end-user computing*. MIT Press.
17. Nardi, B. A., Ed. (1996). *Context and Consciousness: Activity theory and human-computer interaction*. MIT Press.
18. OWL (2005). Web Ontology Language: www.w3.org/TR/owl-features
19. RDF (2005). Resource Definition Framework: www.w3.org/RDF
20. Suchman, L. A. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge University Press.
21. UAM (2005). Unified Activity Management project. www.research.ibm.com/uam
22. Whittaker, S. (2005). Supporting collaborative task management in e-mail. *Human-Computer Interaction, 20.*
23. Whittaker, S., Moran, T. P., & Farrell, S. P. (2005). Why Email is not Enough: Combining Communication and Shared Representation to Support Activity Management. *ECSCW 2005, Workshop on Activity.*