

# Task Integration for Knowledge Workers: Especially those Involved in Multiple Collaborative Activities

Roger Tagg

School of Computer and Information Science, University of South Australia  
Mawson Lakes SA 5095, Australia

**Abstract.** One approach to overcoming the information overload that bedevils much collaborative work is to move towards an activity focus in the next generation of groupware tools. This implies a need to integrate and categorize all the activities, including tasks, which each user is required to do. This paper describes an ongoing research programme to address this need. An architectural model for future developments is proposed, and some prototypes developed by the author's research team are described.

## 1 Introduction

Knowledge work in the late 1990s and early 2000s has been characterised on the one hand by *collaboration between widely-distributed specialists* but also on the other hand by *information overload* [1, 2].

In this author's long practical experience, current IT support for knowledge workers often appears to contribute to, rather than mitigate, information overload. Two primary reasons are the slog of managing runaway volumes of email, and the lack of integration between the different IT tools that a single user is required to use.

One proposed approach to improving this situation is to align all information resources, within each user's interface, with the structure of that user's activities – rather than with the tools that created them. By doing this, the intention is to move the user's focus:

- *from* one (or more) "to read" inbox with one-at-a-time access to many separate IT tools
- *to* a single "to do" list that includes tasks drawn from all possible sources, e.g. application systems, projects, workflows, groups or requests from other individuals.

This paper describes an ongoing research programme along these lines, led by this author. Section 2 reviews related work towards re-orienting knowledge work to an activity focus. Section 3 proposes an architectural model for future developments, and highlights the key technologies required. Section 4 describes some exploratory prototypes that have been developed by the research team. Section 5 discusses the work that still remains to be done, and the means by which improved user support could become widely available.

## 2 Related Work

Attempts to improve IT support for distributed knowledge specialists date back to early work on Activity Theory and CSCW [3, 4]. The 1980s and 1990s saw the arrival of commercial software for Groupware and Workflow, and the increased attention given to Business Process Management (BPM).

Although some authors, e.g. [5], regard Workflow and BPM as constraining influences on the creative aspects of collaborative work, a balance has to be struck between unconstrained ad hoc interaction and managing the collaborative process (e.g. by remembering best practice patterns).

While earlier developments in IT support mainly addressed the group's needs, much recent work has shifted towards the individual user's perspective. The term PIM (Personal Information Management) has been introduced in this context by [6]. A study of practices and problems with modern desktop systems has been described in [7]. The observation is made that users' computing habits are often forced into grouping by the tool used or the file type, rather than the work context. A number of authors, e.g. [8, 9] have recognized these difficulties and have built and trialled prototypes, named TaskMaster and Activity Explorer respectively.

However a continuing challenge has been to extract, from free-form text, accurate enough context data to recognize the tasks that are implied and their place in the user's activity structure. One recent approach to this problem is that of SmartMail [10] from Microsoft Research.

A group from IBM [11] has proposed the concept of Unified Activity Management (UAM), based on the use of RDF (to represent the structure of activities) and OWL (to provide an ontology of common collaborative activity). The aim of UAM is to complement, rather than replace, workflow and process management, especially where multiple organizations are involved.

Another group, based at the University of Aarhus in Denmark [12], has trialled an extension to the Windows XP operating system for Activity Based Computing (ABC). Applications and documents that are part of the same activity appear as sub-windows of an activity "super-window", which can be brought to the front when the user is working on that activity. Applications can also be shared both synchronously and asynchronously.

## 3 An Architectural Model for Task Integration

The model proposed in this paper is shown in Figure 1. It follows similar principles to that of [11], but takes on board two extra considerations. One is the concept of many-to-many group membership [13], which recognizes the trend that many users have to work in many groups simultaneously.

The other consideration is that each user maintains, usually tacitly, a personal ontology structure of the concepts he or she is involved with. This structure can include both topics of interest or aspects of the user's work. With present technology, users represent their personal ontology with local or network file folder structures, personal or shared email archive structures and web browser bookmarks; but these

structures often conflict and duplicate each other. The user is also influenced by the ontologies of the groups to which he or she belongs, as well as "best practice" in the domain of interest.

The left hand side of this diagram represents things that happen on the user or client side, whereas the right side represents the user's "outside world" of the groups he or she participates in, and of other individuals.

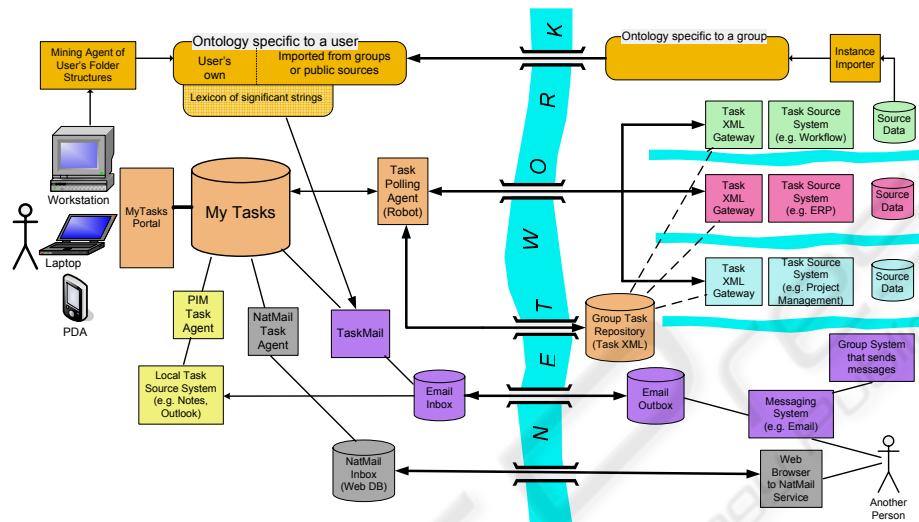


Fig. 1. Overview of Proposed Architecture and Prototype Development.

The winding "rivers" represent the separation between the user's client and various servers. The bridges represent the networks that support data communications. Across the top is the meta-data, represented by ontologies, which are partly specific to each user and partly group-oriented. Messaging is represented by the two lowest "roadways" across the main river. The main channels for task integration are shown above these routes and below the ontologies.

### 3.1 Unified Model of Tasks

There have been ongoing attempts, among some commercial groupware vendors, to establish a standard for interchange of calendar requests and appointments between heterogeneous packages. Notable among these are iCalendar [14] and CalDAV [15]. iCalendar includes a component VTODO which represents an assigned action item. However this does not hold details of the workflow template, or the project or sub-project, from which it comes – and it is not yet included in the more recent CalDAV.

Since many general groupware and "to-do list" packages already support tasks in some form, a comparable standard for tasks is surely desirable. At the time of writing, the only ongoing effort appears to be GroupDAV [16], which is supported by a number of independent "open groupware" vendors.

The project management (PM) community has also proposed a standard PMXML [17] that includes task information. But according to [18], it has not gained wide

acceptance, and MS Project formats are nearest to a *de facto* standard. In any case, a PM-oriented standard is not ideal for the paradigm of multiple business cases following a template, which is inherent in workflow systems.

Therefore, a key part of the model proposed in this paper is a unified task model as shown in Figure 2 below. This covers both recurring, workflow-style tasks that follow a template (left hand branch) and project-style tasks that are defined newly each time and have more emphasis on scheduling and resources (right hand branch). The lower central object class represents the basic Task Instances, while the bottom right class covers Events that happen in the life of a task instance. The class at the top represents the Source Systems where tasks are managed and coordinated from a group viewpoint. This model is still very much a draft and work is in progress to develop it further.

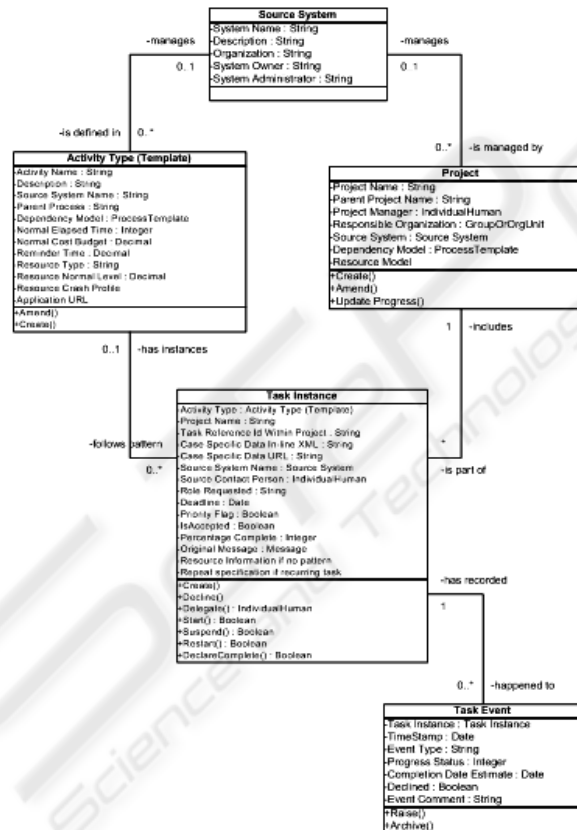


Fig. 2. Proposed Unified Task Model (UML format).

### 3.2 Categorization of Tasks

One objective of a new architecture must be to present the user with a categorization of his/her tasks at the right level of granularity. This author's own experience suggests that a knowledge worker may prefer not to have more than 5-6 major work categories

active at any one time, although there may be 10-12 sub-categories within each main category. This suggests a need to carefully personalize a user's ontology if it is to be used to drive an automatic categorization of tasks. Each user would be responsible for maintaining the top 2 layers of his or her ontology so as to correctly represent his/her current activity structure.

Some tasks may need to appear in more than one of the user's work categories. For example, I may run two courses and a student wants to switch from one to the other. This means one may have to create "alias" task instances. Categorization may also have several dimensions (e.g. relating to person as well as project) and there may be a choice of which the main category is. As well as this, some automatic categorizations may be fuzzy, with significant probabilities that a task could be categorized in more than one way, even in the same dimension.

### 3.3 Identification of Task Instances

For task instances which are based on a template (e.g. those arising from a workflow or ERP system) there are two further issues. One is the storing of template information on the user (client) side, so that only task *instance* data (the large central entity in Figure 2) has to be transmitted. The other issue is over the identification of the individual *business case*. Two tasks from different sources may not use the same identifiers (e.g. one from a workflow system and another via email), but they may in fact relate to the same customer transaction. To cope with this, one needs to include all possible identifiers of individual business instances in the user's personal ontology. Our approach to this is to download these identifiers from a shared database (e.g. tables of students with names, numbers and email addresses). This is represented in Figure 1 by the software component "instance importer" on the very top right.

If there can be fuzzy or inexact matching of text strings to ontology concepts, we envisage an intelligent agent that would evaluate the likelihood of two task instances from different sources actually applying to the same business case.

### 3.4 Determination of Context

The algorithms by which tasks that are implicit in message bodies and attached files can be categorized automatically do not yet represent a mature technology. If a human personal assistant was doing this job, she/he would have to gradually learn the clues that identify the right category, and even then she/he would often have to ask her/his principal for resolution. Two alternative approaches are statistical text mining and matching against an ontology.

The problem with using text mining alone is that no advantage is taken of the user's model of his/her knowledge structure, which is to some extent already there – or can be deduced fairly reliably from metadata on the user's client computer and his/her network resources. But the advantage is that the algorithm, if used continuously, can react to changes in classification that inevitably occur as time proceeds.

The problem with using the ontology approach alone is that the user's knowledge structure, unless checked regularly against current message patterns, is in danger of

becoming out of date. Also, the clues that appear in messages and text documents (mainly text strings) do not necessarily match the names of the concepts or instances in the ontology. This is why an add-on Lexicon facility, as shown in Figure 1 under the upper left ontology, is proposed. This facility would need to access the probability that a given text string indicates a given concept or instance.

A hybrid approach seems most appropriate, for example [19], in which text mining is used to improve ontologies.

Of course, it may not be immediately obvious in a text extract that a task for the recipient is implied. Senders do not often put an explicit "Action" clue in their emails. A phrase such as *must be completed by dd/mm/yy* is suggestive, but it is not always easy to work out whether the recipient really must do the task. However without reliable determination of context, spammers will quickly find ways of ensuring that their spam gets included in their victims' "to-do" lists.

Automatic recognition of significant dates and times is itself a problem. To be useful, a large library of text patterns such as *by next Friday* is needed. Spreadsheet packages such as MS Excel go some way in this direction; e.g. if a user types something that looks like a date, it is assumed to be a date.

## 4 Prototypes Developed

A number of small prototypes have been developed by the author and his research team in the last 3 years, primarily as part of Masters theses and postgraduate projects. In this paper, three of these are described.

### 4.1 "My Tasks" Prototype

This prototype, the most recently completed, is described in a student project report [20]. The prototype offers the user a web portal that consists of a table of "to-do" items that are available to the user from a number of server-side "source systems" in which this user is registered. A client-side file maintains this user's login details for each source system and his/her preferences for the frequency and volume with which new tasks are sought or completion status is sent back to the source system. To achieve this prototype, we have defined a fledgling "Task XML". On the server side, we are building gateways that translate between Task XML and the native task format of the source system. Currently, another project group is building gateways for the Chameleon workflow management system [21], for SAP Workflow and for Microsoft Project.

The coverage of this prototype is indicated in Figure 1 by the horizontal "trident". On the left is the user portal, which looks very much like a typical inbox, except that it can be sorted by additional attributes such as priority, deadline or category. *My Tasks* is a database of the user's current and past (until archived) tasks. Currently this is held in MySQL. The agent named *Robot* converts between Task XML and MySQL, and also stages requests for "any new tasks?" as well as responses like "I've finished that one". It uses the user's logins and the source system's Web Service API to invoke

a function in the source system. A Task XML gateway has to be built for each software product that runs a source system.

In the case of a user who accesses tasks from totally autonomous systems in different legal entities, the tasks are only integrated in the robot's buffers. However certain groups may take the initiative of setting up a shared task repository, so that all group members can access tasks from the same server-side source that is already in Task XML.

#### 4.2 Task Mail

This prototype [22] attempts to pre-process incoming email into potential tasks by matching text strings in parts of the message header and body. It does this by adding a number of "keyword tags" to the message, which represent matched ontology concepts and instances. It also strips off attachments and creates an inbox-style relational table of task instances, in which the derived task becomes a row in a task database, whose attributes include URI links to the text of the original messages and the attachments.

TaskMail is shown in Figure 1 by the sequence leading from the bottom right, across the second lowest bridge, then diagonally up to My Tasks. A branch is shown heading horizontally to the left to cater for tasks that are sent as attachments to emails, as can be done between two systems that use MS Outlook.

The current version of TaskMail is not yet integrated with My Tasks, and it is also limited in that its ontology is a simple flat table of strings containing the name of the concept to which each string is associated. This means a purely deterministic relationship between strings and concepts, and no inter-concept relationships.

#### 4.3 EzOntoEdit

This prototype was described at last year's ICEIS conference [23]. The purpose is to provide a user-friendly, graphical tool for users to import ontologies and to adjust them. The original intention was to allow the user to tune the behaviour of the TaskMail pre-processor described above.

EzOntoEdit's role in the architecture in Figure 1 is the maintenance by the user of the assembly "Ontology specific to a user" at the top of the figure, just to the left of the "river".

This turned out to be a large project and the working version did not contain all the desired features. There is currently no separate lexicon, or provision for "synonym" text strings. However, one notable feature of the prototype was that versions were built simultaneously that can run on Windows, Linux or Mac OSX.

#### 4.4 NatMail

NatMail (named after the student who developed it) is shown by the route that crosses the lowest bridge on Figure 1. This prototype [24] took a different approach, namely that one way of enabling incoming messages to be reliably classified and pre-

processed is to require the sender to do the classifying. A website was set up, into which "contact us" pages could be defined using a Wizard. These pages would all have an area for a free text message and scope for attachments, but they would also have a number of mandatory fields, each representing a "dimension" of classifying information as defined by the receiving user.

The NatMail task agent moves input received through the web server into the My Tasks database. The original intention was to build a system that could be tailored using a Wizard for any CRM (Customer Relationship Management) situation.

Fields that can be set as mandatory can include a boolean "Action Required?", a "Deadline Date/Time", and "Requested priority 1-10". These help to make explicit, and categorize, any task inherent in a message.

The disadvantage, for collaborative teams that work in a peer-to-peer mode, is that forcing messages into a format is not in tune with current computer etiquette. A survey of local companies carried out at the same time as this project returned a resounding "no thanks". However a number of large organizations - e.g. banks, airlines and government - and even some conferences - use this approach rather than free-form email. The common feature is that these organizations know that individuals need their service – thus it is accepted practice for a dominant partner in a relationship.

#### **4.5 General Assessment**

In retrospect, although these prototypes were all successful to some extent, they were not able to be adopted as they stand. One reason is that as separate tools, they do not offer enough advantage to a real user to make a major shift in their computing practice, even for the purposes of a trial. Also, the students working in our group do so on a fairly short-term basis, either one semester or one year. They often run out of time to properly document their work. Their tools were only tested on artificial data, although TaskMail was used on messages diverted from this author's own inbox. In any case, to carry out trials on human subjects, ethics approval is required, and this can take almost as long as the project itself.

This problem of getting realistic trials is not however limited to university research teams. Even Bellotti at Xerox PARC [8] and Muller and Geyer at IBM [9] had to do their trials on groups of "interns". It seems that to stand a chance of a realistic trial, any developer of new IT cooperation support software has to build up a package of critical mass that can establish its advantages clearly with users – who will probably not want to go back to whatever they were using before!

### **5 Future Work**

With a new intake of Masters thesis and project students, we are continuing research and development in this general area. One thesis student who is continuing into 2007 is furthering work on the Task XML.

There are three areas of planned future work with our current prototypes:

- a) to enhance their usability;



- b) to integrate them into a single user interface; and
- c) to link them with more related systems, not just for the source of tasks but also for instance importing.

A future gateway is planned for Microsoft Outlook tasks, both those that exist in shared Exchange folders and those that appear in the user's own Outlook .pst files.

Our plans for EzOntoEdit are to extend the current prototype to accommodate:

- Import of ontology concepts and instances by mining of the user's (or group's) folder structures;
- Import of ontology instances from the rows of database tables in a shared or local database;
- Development of an experimental lexicon-ontology link.

Besides continuing to develop additional Agents that were proposed in [25], we are also planning to develop prototypes of more advanced user interfaces, for example the Dashboard introduced in the same 2003 paper, and for mobile devices.

## 6 Conclusions

It may be that only very big players stand a chance of succeeding with task integration. However although the main vendors of groupware, including Microsoft and IBM, all have projects addressing some of these problems, the average user is still to see many signs of the brave new world. It may be that there are more serious problems like spam, phishing and viruses that may threaten the whole groupware market - as well as user confidence. Also, vendors may be nervous about abandoning the vehicles that have given them so much desktop dominance for so long. Maybe we have to wait for a new player to make a bold move, but even they would need to capture some big-name user organizations before the mould can be broken.

## References

1. Whittaker, S. and Sidner, C., Email Overload: Exploring Personal Information Management of Email, In *ACM Conference on Computer-Human Interaction (CHI)*. <http://dis.shef.ac.uk/steve-whittaker/emlch96.pdf> (1996)
2. Kirsh, D., A Few Thoughts on Cognitive Overload, *Intellectica*, <http://icl-server.ucsd.edu/~kirsh/Articles/Overload/published.html> (2000)
3. Wikipedia, Scandinavian Activity Theory, [http://en.wikipedia.org/wiki/Scandinavian\\_activity\\_theory](http://en.wikipedia.org/wiki/Scandinavian_activity_theory) (2006)
4. Wikipedia, Computer Supported Cooperative Work, <http://en.wikipedia.org/wiki/CSCW> (2006)
5. Hoffmann, O., Copley, D., Copley, A., Nguyen, L. and Swatman, P.A., Creativity, Requirements, and World Views, *Aust J of Info Systems*, Vol 13, No 1: pp159-175 (2005)
6. Jones, W. and Bruce, H., A report on the NSF-sponsored workshop on Personal Information Management, Seattle, USA, <http://pim.ischool.washington.edu/final%20PIM%20report.pdf> (2005)
7. Ravasio, P., Schaer, S. and Krueger, H., In Pursuit of Desktop Evolution: User Problems and Practices with Modern Desktop Systems, In *ACM Transactions on Computer-Human Interaction (CHI)*, 11(2) (2004).

8. Bellotti, V., Ducheneaut, N., Howard, M. and Smith, I, The design and evaluation of a task management centered email tool, In *Proc ACM Conference on Computer-Human Interaction (CHI)*, Fort Lauderdale, Florida (2003)
9. Muller, M., Geyer, W., Brownholtz, B., Wilcox, E., and Millen, D., One-Hundred Days in an Activity-Centric Collaboration Environment based on Shared Objects, In *Proc ACM Conference on Computer-Human Interaction (CHI)*, Vienna, Austria (2004)
10. Corston-Oliver, S., E. Ringger, M. Gamon and R. Campbell. 2004. Task-focused summarization of email. In *Proc of Text Summarization Branches Out Workshop*, ACL (2004)
11. Moran, T., Cozzi, A. and Farrell, S., Unified Activity Management: Supporting People in E-Business, *Communications of the ACM* Vol. 48, No. 12 (2005)
12. Bardram, J., Bunde-Pedersen, J. and Soegaard, M., Support for Activity-Based Computing in a Personal Computer Operating System, In *Proc ACM Conference on Computer-Human Interaction (CHI)*, Montreal, Canada (2006)
13. Tagg, R. Activity-Centric Collaboration with Many to Many Group Membership, In *Proc COLLECTeR Europe Workshop*, Basel, Switzerland (2006)
14. Dawson, F and Stenerson, , D., Internet Calendaring and Scheduling Core Object Specification, Internet Engineering Task Force RFC 2445, <http://www.ietf.org/rfc/rfc2445.txt> (1998)
15. Dusseault, L. and Whitehead, J., Open Calendar Sharing and Scheduling with CalDAV, *IEEE Internet Computing*, vol. 9, no. 2, pp. 81-89 (2005)
16. Hess, H. Storage of Groupware Objects in WebDAV (GroupDAV), GroupDAV website, <http://www.groupdav.org/draft-hess-groupdav-01.html> (2004)
17. Volz, R (2002): PMXML - a XML standard for Project Management, [http://www.vrtpj.com/content/istandards/pmxml\\_en.html](http://www.vrtpj.com/content/istandards/pmxml_en.html) (2002)
18. Curran, K. Flannagan, L. and Callan, M., PMXML: An XML Vocabulary Intended for the Exchange of Task Planning and Tracking Information, *Information Technology Journal* 3 (2), pp 192-5 (2004).
19. Dittenbach, M., Berger, H. and Merkl, D., Improving Domain Ontologies by Mining Semantics from Text, *Proc. 1st Asia-Pacific Conference on Conceptual Modelling*, Dunedin, New Zealand, <http://www.crpit.com/confpapers/CRPITV31Dittenbach.pdf> (2004)
20. Bui, N. and Bhatt, B. Enhancing a Current Prototype on an Integrated Task Interface for Personal and Group Information Management, Masters Project Report, University of South Australia, available from author (2006)
21. O'Hagan, T. Chameleon Workflow Demos, University of Queensland website, <http://www.itee.uq.edu.au/~tohagan/Chameleon/> (2005)
22. Punekar, S (2005) Ontology Based Add-In for Automatic Classification of Incoming E-Mail, Masters Thesis, University of South Australia, available from author (2005)
23. Einig, M., Tagg, R. and Peters, G. Managing the knowledge needed to support an electronic personal assistant, In *Proc. Intl Conference on Enterprise Information Systems(ICEIS)*, Paphos, Cyprus (2006)
24. Mahalingam, N., Potential for Process Knowledge Management Techniques in Administration, Masters Thesis, University of South Australia, available from author (2005)
25. Tagg, R. Software Agents to Support Administration in Asynchronous Team Environments, In *Proc. Intl Conference on Enterprise Information Systems(ICEIS)*, Angers, France (2003)