New lenses to investigate media use: The layering process perspective

Roger Tagg  
*University of South Australia, roger.tagg@unisa.edu.au*

Prashant Gandhi  
*University of South Australia, prashant.gandhi@gmail.com*

Raaj Srinivasan Kumaar  
*University of South Australia, skraaj@gmail.com*

Follow this and additional works at: [http://aisel.aisnet.org/ecis2009](http://aisel.aisnet.org/ecis2009)

**Recommended Citation**  
Tagg, Roger; Gandhi, Prashant; and Srinivasan Kumaar, Raaj, "New lenses to investigate media use: The layering process perspective" (2009). ECIS 2009 Proceedings. 163.  
[http://aisel.aisnet.org/ecis2009/163](http://aisel.aisnet.org/ecis2009/163)

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Email has been recognized as one of the chief sources of the current epidemic of information overload. The work described in this paper is part of an ongoing project to re-focus the attention of information workers on “what needs to be done” rather than “what someone wants us to read”. The approach is to base a software agent that pre-processes incoming email on the user’s “personal ontology”. The ontology used in this work contained two facets, namely Work Topics and Task Types, and also included lexical “clues” whereby, if such a word or phrase is found in the text of a message, the system can infer with a certain probability that a particular ontology concept (class or instance) is relevant. The paper describes the prototype tools developed and some initial trials. Further trials and improvements are planned.

Keywords: Information Overload, Task Orientation, Ontology, Lexicon.
1 INTRODUCTION

The concept of information overload has been recognized for more than a decade (e.g. Whittaker et al 1996). Suggested causes range from trends in the organizational climate towards mistrust, risk aversion and micro-management, e.g. (Allen and Wilson 2003) to the availability of too much information and of the IT facilities themselves, e.g. (Alberts 1996).

Many authors, e.g. (Mulder et al 2006) have stated that email is the biggest contributor to information overload. A workshop discussion in (Whittaker and Sidner 2005) reported, for example, that in the previous three years, the number of email messages has increased eight fold. Some organizations have instituted email-free days (Wakefield 2008) and some individuals have given up email as a main channel of communication, e.g. (Suarez 2008).

The search to find ways to use IT to relieve information overload is becoming more urgent with every year that passes. Without relief, information workers will find that they have less and less time to carry out the functions for which they are actually employed, and decisions they make may be adversely affected (Sutcliffe 2006). The authors of the current paper, following a series of earlier experiments (Tagg 2007) under the theme “Virtual Private Secretary”, have over the year 2008 developed a prototype that aims to address individual overload, primarily that arising from email. The principles of our prototype are that:

- It should be activity-oriented, i.e. it should be concerned with what we as individuals most need to do, rather than what email senders want us to read.
- It should reflect the collaborative nature of most information work, but from a federal, bottom-up viewpoint rather than a top-down “one group, many members” perspective.
- It should make use of the information that a computer system can provide about an individual’s work topics, contacts, priorities etc.
- It should not be so complex that an ordinary user will be deterred by the effort of setting up and maintaining it.

The remainder of this paper is organized as follows. We first describe our rationale for the approach we have taken and the stages of the work we undertook. Next, we give the details of our prototype system for recognizing work priorities and tasks, and some comments on the evaluation we performed. Following this, we discuss our approach for users to maintain their personal ontology including the lexical clues. We then outline some of the related work that is addressing similar objectives to ourselves, and this is followed by some discussion and comparison. We then conclude with some suggestions for future work.

2 RESEARCH DESIGN

2.1 Justification for using a prototyping methodology

According to Clay Shirky, reported in (Wagner 2008), Information Overload has been with us for so long that it should be seen as a fact of life, rather than a new problem; instead, the real problem is that our filters are no longer adequate. The work described in this paper, therefore, concentrates on one activity where, in the authors’ experience, better filtering seems to be particularly needed; namely that of incoming email messages.

The literature shows a wide range of theories for reducing information overload, ranging from intelligent software agents to organizational restructuring, personal goal re-appraisal and even speed reading. However so far there has been little evidence of success. One symptom is the lack of major improvements (Hall 2004) in task management facilities in everyday tools such as MS Outlook, or in
web browsers. We have therefore taken the approach of building a number of prototypes, based on relatively heuristic algorithms, to see what working with such prototypes feels like to users. A similar approach has been taken by other researchers in this field, for example the group at Xerox PARC (Bellotti et al 2007).

2.2 Review of alternative approaches to a prototype email pre-processor

Current commercial tools, such as Outlook, typically offer an email filtering facility using rules, with a wizard that helps the user express the desired rule. However Ducheneaut & Bellotti (2001) suggest that rule-based filters are not very widely used, suggesting that this feature is either not useful or is difficult to use. The same authors argue that filtering is mainly useful for filing schemes based on the identity of the sender, and offers weaker performance when required to deal with more complex categorization. Crawford, Kay and McCreath (2002) suggest that the process of composing rules is, for many users, cognitively demanding.

Another approach has been to extend the idea of spam filtering, where an approach based on Bayesian classifiers and machine learning is usually used. The application of this approach for categorizing emails into multiple categories (rather than just spam / not spam) can be found in studies conducted by Maes (1994), Segal and Kephart (1999), Mock (2001), and Crawford, Kay and McCreath (2002), amongst others.

Some other approaches address the recognition of tasks and priorities, rather than the categorization of emails. In current practice, the receivers of emails are highly dependent on the senders. Outlook, for example offers “follow up” and “high priority” flags; the sender can also send a formatted “task” as a message – though this does not seem widely used. Bellotti and her colleagues (2007) have approached tasks partly through recognizing threads of emails, coining the term “thrasks”. The more mundane reality, though, is that most users have to fill a form manually in order to turn a message into a task that would appear in a “to do” list.

Although historically, many proposed methods for email pre-processing have been based on Machine Learning, we have not followed this approach. Our reasons were:

- Machine Learning techniques depend on acquiring sufficient training data in order to build a competent classification system. The “slow start” problem, by which classifiers slowly build-up their competence as more examples are provided, may be more intense when a user has, say, tens of different categories rather than “spam/not spam”. A significant amount of training data, with user supervision, would therefore be needed in order to acquire strong classification results from the beginning.

- Making use of a personal ontology takes account of what is known in advance about the user’s work structure. Ontologies have recently gained increased attention within other research areas, and we believe that their application to email pre-processing is worth further exploration. However we recognize that, because user’s work structures change over time, a hybrid approach will be the most effective in the longer term.

2.3 History of previous prototypes developed within the authors’ research group

VPS (Virtual Private Secretary) is an ongoing project within this university’s Information Systems Laboratory, having started in 2005. Some of the earlier efforts on this project are described in (Tagg 2007). An overall view of the architecture of the current and planned components of the VPS project is shown in Figure 1. The upwards blue arrow represents the direction in which we think the user’s task interface ought to develop.
The numbers in the diagram refer to some of the prototypes developed within the VPS project up to the end of 2007:

1. An email pre-processor “TaskMail” that reformats emails, which contain one or more key words in a defined set, as tasks in a “to-do” list with links to the message body and any attachments.
2. A middleware architecture for aggregating tasks from different source applications such as Workflow, Project Management and ERP (Enterprise Resource Planning).
3. A “myTasks” portal that gives a unified to-do interface to tasks that come from different sources (e.g. workflow, project management, ERP system) via a unified TaskXML (Tagg 2007).
4. An Outlook add-on for importing tasks formatted in TaskXML into the Outlook task facility.
5. An “drag and drop” ontology editor called EzOntoEdit that allows users to import ontologies (OWL or CSV formats) into a personal OWL file and amend it graphically wherever possible (Einig et al 2006).
6. The VPS email categorization agent that is described in this paper, which uses the personal ontology maintained in 5 above.

In addition to the above, we also attempted in 2005 to support “sender-assisted” categorization, by means of a generator of “contact us” pages based on the personal ontology of the recipient, senders of messages being required to enter certain fields that give message context.

A further project in 2007 added into EzOntoEdit the facility for including “clues” in the user’s ontology. A clue is defined as a lexical string which, if detected in an incoming document, indicates (with a certain probability) that the document is relevant to a particular ontology category.

2.4 Attempting to ease the burden on the user in maintaining a personal ontology

However initial evaluation of the clue-enhanced EzOntoEdit ontology editor suggested that users would probably be discouraged by the effort required to fill in the forms needed to specify these clues. We were also concerned about the user’s workload in reacting to changes in the pattern of his or her work.
We therefore investigated the use of an automated text analysis tool, Leximancer (Smith 2003) which has been used in a number of domains to suggest both “Themes” (what a document is about) and “Concepts” (words or phrases most strongly associated with a Theme).

We applied Leximancer to collections of emails in two different ways:

- Documents created by concatenating all the emails - received or sent - that had been already archived by a user, into each of around 20 major categories;
- A single concatenated document of all the emails in the same user’s inbox that had not yet been archived.

The first approach aimed to detect significant differences in the frequency of words or phrases in the different categories, while the second approach tried to deduce themes *ab initio*; the desired number of themes being set as a parameter. While the results made some sense, we judged that:

- The influence of the chosen stop word list was very high.
- The effect of email senders including the text of previous emails in a thread was variable and unpredictable.
- Files of concatenated emails behaved quite differently from the single longer documents for which Leximancer had been designed.

We therefore decided to abandon this approach for the time being. This research is described in more detail in (Tagg, Lalwani and Srinivasan Kumaar 2008).

### 2.5  Research design approach eventually taken

We therefore split our work for the remainder of 2008 into two parts:

- To build a new prototype email pre-processor, this time one that separated the ontology classes and instances (in two dimensions, “work topics” and “task types”) from the lexical clues used; and
- To develop a better, more user-friendly approach to capturing the lexical clues from the user’s normal activities.

We would later bring these two parts together and test their effectiveness in predicting the work topic and “taskiness” – following a term coined in (Macbeth 2008) - against the manual judgment of users.

### 3  THE VPS CATEGORIZATION AGENT

#### 3.1  Mode of operation of the prototype

The mode of operation of our prototype is described in Figure 2 below. The letters show the high-level sequence of operation.

- Ontology maintenance (A1-A3) is discussed in section 4 below.
- The OWL ontology is converted to Oracle (B) for the agent’s internal purpose, although the definitive version is maintained in OWL XML format.
- Messages arrive in the user’s mailbox asynchronously (C), however …
- The email agent only reads the inbox at set intervals (D), during our trials, this was 5 minutes.
- Pre-processing (E) involves stripping off message bodies and attachments into separate web-accessible files, and tokenizing the message body and other fields.
- The context recognition stage (F) uses the ontology to tag the messages in one or more “work topic” or “task type” categories, the default being “uncategorized”.
- The user views the categorized task list, derived from a Tasks database (G1) in a “MyTasks” portal, with links to the original message text bodies and attachments on a web server (G2).
Our approach to categorization is a very simple and heuristic one.

1. Scan an incoming document for occurrences of any of the clue strings (words or phrases) in the user’s ontology.
2. Apply the concept of Indication Strength, which is the subjective probability that presence of the clue’s string indicates relevance to an ontology concept.
3. The Weighted Indication Strength (WIS) for each matched clue is then calculated based on multipliers that reflect the number of occurrences of the clue. These range from 1.2 (or a 20% raise of the indication strength) for 2 occurrences of the clue, to 1.75 for 10 occurrences or more.
4. Aggregate all the clues that refer to the same ontology category (class or instance) and add their WIS values.
5. If the total of the WIS values matches or exceeds the threshold value (default is 1.0), insert a row in the “categorized” table, and include URI links to the original message and any attachments.
6. Repeat for each category that is indicated by clues in this document.
7. Repeat for all documents received since the last clock event (currently, at 5 minute intervals).

The procedure is performed twice; the first time with respect to Work Topics, the second with respect to Task Types. The Task Types we used were as in Table 1 below. In our trials, we removed the middle “invitation” priority level for simplicity, and reallocated the Task Types as shown by the arrows.

<table>
<thead>
<tr>
<th>Priority -&gt;</th>
<th>High</th>
<th>Invitation</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“Definite task”</td>
<td>“Conditional task”</td>
<td>“For information only”</td>
</tr>
<tr>
<td></td>
<td>“Reply only” (explicit request)</td>
<td>“Questionnaire”</td>
<td>“Private communication”</td>
</tr>
<tr>
<td></td>
<td>“Vote”</td>
<td>“Conference invitation”</td>
<td>“Uncategorized”</td>
</tr>
</tbody>
</table>

Table 1. Task Types used in the prototype.
The user’s ontology can be amended between clock events, and if it has changed, the new ontology will automatically be used the next time the in-box is processed.

The myTasks portal displays a scrollable table of inferred tasks, sorted either by Task Type within Work Topic, or vice versa. By default, only the high-priority tasks are displayed, but the user can choose to look at the low-priority list if he or she has time to do so. There is also an option to display the original email message body for a task on a table that has been highlighted by the user.

3.2 Evaluation of the prototype

In the time available we were only able to test the system against one user’s manual categorization. However this was done twice, once for an early version of the prototype and again for a later version when the task inference had been improved (by adding to the ontology some clues indicating template-based task-requesting emails that were repeatedly received). Both trials involved around 100 emails forwarded from the user’s live inbox. For Work Topics, performance was consistent in both versions with precision over 80% and recall 67%. Task Type categorization was less successful – precision was only 43% in the first trial and 62% in the second; recall was around 55% in both.

On reflection we felt that we still need to improve the ability to accurately infer tasks from the text of emails. Three particular problems appeared to contribute to these results

- a proportion of definite tasks were not recognized as such, due to the seemingly inexhaustible circumlocutions in the English language of asking for something to be done
- we did not have in the prototype a means of recognizing dates, which appear to be a good indicator of taskiness
- our “Conditional Task” category was too wide and did not distinguish between conditions that were quite likely to be true and those that were extremely speculative.

3.3 Results of an informal questionnaire carried out during the project

Halfway through the project, we demonstrated the prototype at a “project fair” to a mixture of students, academic staff and project sponsors from industry, and asked them to complete a questionnaire; 43 people responded, out of 70-80 people attending. We do not claim that this is a representative sample of users in the wider world, but we were interested to gauge initial reactions from people who saw our prototype working.

Over half the respondents reported that they regularly juggled 6 or more work streams. Deadlines were the chief driver to carry out tasks. People received 75% of their tasks through email, and they frequently “batch” tasks so that they get a run of doing similar tasks together. 80% declared an interest in trying a tool such as our prototype, although most said that they would like to keep the ability to look at the traditional email inbox view. The feedback suggested that we need to look more closely at peoples’ personal information habits and preferences.

4 GENERATING A PERSONAL ONTOLOGY AND CAPTURING LEXICAL CLUES IN VPS

This section details the work done on the ontology maintenance side of the project. Most authors, e.g. (Sheth et al, 2003) take the view that building an ontology, especially a complex one, is not a task for the average end user. We therefore need to find a mechanism by which the user can, relatively easily, generate and maintain a simple personal ontology and capture the lexical clues that indicate that a document relates to an ontology category. The EzOntoEdit tool (Einig et al 2006) mentioned above is oriented to importing a pre-existing ontology from a work group or a domain, then tuning it to one’s personal preference, rather than creating it from scratch.
The problem remains of how to capture the lexical clues. Figure 3 shows a range of modes of capturing clues and the ontology structures to which they relate.

It has been rightly pointed out that, by adopting an ontology-based approach, we might be simply transferring the user’s workload to manually maintaining his or her ontology, instead of ploughing through a mass of emails.

While totally automatic maintenance of lexical clues seems unrealistic, it might be possible to do regular text mining of samples of recent documents and messages, and for the user to interpret those results. However our own experiences with Leximancer (Tagg et al 2008) suggested that such a system might turn out to be very complex. Semi-automatic approaches have been proposed, e.g. (Wang et al 2006, Fortuna et al 2006) but there are not yet many reviews of their user acceptability over an extended period. In our study, we decided on an intermediate - but fundamentally semi-manual - approach, assisted only by the importing of “blueprints” or “stereotypes” of the clues used by users doing similar tasks, and by the process of “snipping”.

Figure 3. Various Modes of Generating Lexical Clues for an Ontology.

Figure 4. SnipCat main window.
“Snipping” in our prototype is derived from the ideas of (Snipit 2008) and from MS Internet Explorer 8 Accelerators (Microsoft 2008). A user highlights a word, phrase or name from any document he or she is working on (e.g. an email, Word document, spreadsheet, PDF file etc) and on the “Copy” command that text appears automatically in the prototype’s window. In Figure 4 above, the user has just highlighted and copied the word Andrew; that text appears in the top left window of our SnipCat (short for Snip Categorization) tool.

The Categories window allows the user to browse the ontology hierarchy and highlight the appropriate category. The Location window allows the user to specify where the text should appear; the options are From, Subject, Body and Anywhere. Indication Strength allows for “maybe” clues – the default is a probability of 1 (slider at extreme right). In the top right window the existing clues for the highlighted category Campus Related are shown. If the green tick is clicked, a Clue instance with Andrew, Subject and 0.9 will be added. Note that “Discriminator” is not the same as Clue, since Andrew might have other meanings, for example if it is in the From location.

One can have a clue that points to a whole Class instead of a particular Instance; currently this is done by highlighting the standard Instance None. Likewise, Other indicates assignment of the string to the next higher level Class.

SnipCat also allows for “Bulk Snipping”. If that button is pressed, the user is prompted for the path of a tabular file (currently MS Excel but potentially also a database), the Sheet (or Table) and the Name of a Column. A typical use would be to import list of contacts such as email addresses, surnames etc. These strings are all set up as clues to the ontology concept chosen.

SnipCat also allows for additions to the ontology, but only at the Instance level. Changes to classes currently require the user to switch to the EzOntoEdit tool mentioned in section 2 above, or some other OWL editor.

SnipCat was tried by 5 users. All found the method fairly easy to use, although to really judge whether or not people would use it, trials would have to continue over a long period. There was a wide range of complexity in peoples’ ontologies; the smallest had fewer than 10 classes with 20 instances and 30 clues, while the largest had:

- 6 work topic classes with 24 subclasses and 98 instances
- 8 task type classes, no subclasses and 27 instances
- 261 clues, split into 25% in the location From, 30% in Subject, 16% in Body and the rest in Anywhere
- The distribution of clues by indication strength (probability) was 67% for a probability of 1.0, 25% for 0.7 to 0.99, 8% for less than 0.7.

5 RELATED WORK

In recent years a number of major prototypes, several of them European funded, have been built addressing similar issues. The systems listed below all follow, to some extent, an ontology-based approach.

- DELOS (Katifori et al 2008) incorporates a PIMS (Personal Interaction Management System) based on a Personal Ontology. This group has explored alternative ontology visualization methods. Their ontology is significantly fuller in detail than that described in the current paper, and they recognize that it is critical to find suitable views to prevent the task of maintaining the ontology from becoming too complex. However, they do not appear to use a clear separation of ontology concepts and clues.
- NEPOMUK (Sauermann et al 2008) includes the concept of a Semantic Desktop, driven by a PIMO (Personal Information Model), which appears to be a form of Personal Ontology. A number of standard classes are provided, and users can add more using a tool. However automatic categorization, in particular of emails, does not seem to be included.
• Perfect Digital Assistant (PDA) (Schraefel et al 2008) has been developed by a consortium involving MIT, University of Southampton and Nokia Research Centre. As in our VPS project, it aims to simulate the work of a personal assistant. It is built on a RDF representation of personal knowledge, which is partly fed by activity log mining, but does not yet appear to support email categorization.

• DYONIPOS (Rath et al 2008) also builds up a user model by mining logs from the whole range of a user’s frequently-used applications and operating system actions. The resulting context “ontology” is expressed as a hierarchical structure of Tasks, Event Blocks and Events. Tasks are recognized automatically while the user is working.

• The Semantic LS system (Krishnan et al 2008) is being developed by IIT Kanpur, India but with European funding. It is targeted at a P2P cooperation model, but for a private group rather than an open internet. A DKS (Domain Knowledge Model) is maintained as a taxonomy in OWL. It does not yet address email documents.

The following noteworthy systems, which are not ontology-based, should also be mentioned.

• TV-ACTA (Bellotti et al 2007) is a development of work that has been going on at Xerox PARC since around 2001. TV (Task Vista) is a “to-do” list, while ACTA is an Activity Centered Task Assistant that works as an Outlook add-in. The system takes account of repeated “workflow-like” activity patterns. Earlier work (Taskmaster) looked at threads of messages related to tasks.

• SCOUT (Sow et al 2006) was developed by a group within IBM. This appears to be primarily rule-based; the “context” referred to is primarily physical location. Their evaluation showed good results, but they only analyzed those messages that were generated automatically by software, which were only a small percentage of the total.

• SmartMail (Corston-Oliver et al 2004), is a Microsoft Research development that “presents the user with a task-focused summary of a message. The summary consists of a list of action items extracted from the message. The user can add these action items to their ‘to do’ list.” Natural language processing is used.

• Chandler (Chandler 2008) is a cross-platform personal information management tool that streamlines the links between emails, calendar appointments and tasks. It was originally designed by Mitch Kapor of Lotus 1-2-3 fame, and supported by the Open Source Applications Foundation. It appears to be one integrated PIM that is generally available.

It seems clear that the major software vendors are seeking a way forward in this area, but may be averse to the risk of prematurely launching a major enhancement to popular systems such as Lotus Domino/Notes and MS Exchange/Outlook. The typical user has yet to see any significant products addressing any of context recognition, task recognition or reducing the fragmentation that arises from having to use different applications for each different file type.

6 CONCLUSIONS AND FUTURE WORK

Most of the prototypes emerging from the projects listed above have been wider-ranging and are further developed than our own. We have limited ourselves to addressing email overload problems, partly because of resource and time limitations, and partly because we feel that email is the single most pressing problem.

Our own prototypes have many shortcomings; in particular, our ontology structure as it exists today is little more than a taxonomy, and is-a and part-of relationships are not always strictly distinguished. Our recognition algorithm is very simplistic and needs testing with more variations over more data. Its performance in inferring tasks from emails is so far only moderate.

Our evaluation process was very limited, due mainly to the strict deadlines imposed by the student authors’ thesis and project timetables. However it is worth commenting that it seems very difficult to obtain reliable evaluation of how tools - like those proposed here – would be received by real users in overload situations. Many related projects have used teams of student interns, but these may not be
representative. However the administrative and ethical challenges of testing, using active employees in a cross section of industries, seem daunting.

Our contributions have been:
- to investigate the idea of clue-enhanced personal ontologies
- to test the principle of using these for pre-processing (as a personal assistant might do) emails and potentially other documents
- to explore some of the issues in inferring tasks from incoming emails
- to test an alternative approach to capturing clues and instances in personal ontologies

We recognize that we have only scratched the surface of what is needed if these ideas were to become part of a widely usable tool with “critical mass”. Some of our current ideas for future development are:
- General improvements to our personal ontology facilities, e.g. incorporating WordNet
- Building a library of blueprints (both basic ontology and lexical clues) for common human roles
- Recognition of dates in text, especially where these are deadlines and therefore indicate a task
- Recognition of the relevance of a message to a specific activity within a work topic
- Use of the context recognition system as a component in a system for annotating observed activities in a grounded theoretic approach to group work (Blackburn et al 2007)
- Integration of non-email sources of tasks into the task database, e.g. “actions” from minutes of meetings that are stored in shared folders
- Investigation into improving the contribution of email senders towards categorization for the message receivers.
- An investigation of habits and preferences in personal information management for a range of users in different work environments.

However we feel that we have made a start on a potentially valuable approach to context recognition that strikes a balance between, on the one hand, a high-technology approach that is hard to implement and maintain, and on the other a more easily manageable – but possibly simplistic - solution.

References
System, CHI 2008 Workshop on Personal Information Management, Florence, Italy


