

Task Context for Knowledge Workers

Mik Kersten and Gail C. Murphy*

Tasktop Technologies Incorporated and *University of British Columbia
mik.kersten@tasktop.com and murphy@cs.ubc.ca

Abstract

Knowledge workers work on many different tasks and must often switch between those tasks. In earlier work, we have shown the benefits of automatically capturing contexts for tasks for a specific category of knowledge worker, software programmers. Captured contexts facilitate task switches and reduce information overload by enabling the display of only the information relevant to the task-at-hand. In this paper, we describe the results of two studies of the use of captured contexts for a broad range of knowledge workers. The first study we describe is a field study of eight knowledge workers who used the model in their daily work for up to 25 days on tasks involving both file and web documents. We found that these knowledge workers need information to decay from their context and that our model is adequate at automatically trimming contexts. The second study is a case study of the use of contexts to support the operations of a software development company. We analyzed task contexts from hundreds of days of work from three users and found similar trends of information decaying from contexts. Results from each study also shed more light on the nature of mixed artifact task contexts.

Introduction

Knowledge workers work on many different tasks per day (González and Mark 2004). Many of these tasks require locating just a few relevant documents needed to complete a task amongst the vast number of documents available on hard drives and on the web. When a worker switches between tasks, she typically loses the collection of information gathered for a task, requiring additional navigation and searching to re-locate relevant information when a task is resumed.

To help these knowledge workers remain cognitively focused on a task, it is desirable for a mechanism to determine, as unobtrusively as possible, which documents are associated with each task. An interface can then use this information to facilitate switching between tasks.

UMEA and TaskTracer are examples of such a semi-automated approach; each uses a knowledge worker's interaction with the desktop as a basis for associating documents with tasks. In UMEA, a knowledge worker indicates the start of a task and subsequent interaction with particular document types, mostly MS Office documents, is then associated with the task (Kaptelinin 2003). TaskTracer takes a similar approach over more document types, including web documents (Dragunov et al. 2005). Other work tries to reduce the need for a worker to be aware of the underlying technology: work on TaskTracer applies machine learning to infer task boundaries (Stumpf et al. 2005) and CAAD applies pattern mining to find artifacts to group automatically as contexts for inferred tasks (Rattenbury and Canny 2007). In all but CAAD, documents accessed by a worker become associated and visible for a task unless the worker applies specific commands to delete a document. In CAAD, despite the use of a relevance weighting to determine which information to display for a task, the second most frequent action by users is to delete information from the information automatically determined to be part of a task.

A task context model we have developed and validated on programmers also uses interaction to associate information with tasks indicated by the programmer (Kersten and Murphy 2006). Our model differs from other work in using a degree-of-interest weighting for each piece of information based on the frequency and recency of a user's interaction with the information. Over the course of a task, the relevance of various pieces of information rise and fall. One way we use this weighting is to filter what information is displayed as part of a task. Information of high importance to a task can be decorated to be highly visible in a view; information accessed as part of exploration or a misstep can gradually decay out of view. In a field study of 16 programmers who used this task context model, we found that decay was important in the acceptance of the approach because it helped optimize the use of screen real estate for highly important information and it did not constrain how information was accessed, as

irrelevant information would decay from view. Our task context model is now used by hundreds of thousands of programmers daily through the Eclipse Mylyn framework and project.

Programming is a particular kind of knowledge work that involves interaction with highly structured information. We conjectured in 2007 that our task context model could also help knowledge workers who interact with less structured information, such as office and web documents. We performed a field study in 2007 in which eight knowledge workers used a task-focused desktop that included our task context model to capture local file, shared file, and web documents accessed as part of a task. We found that these workers did revisit tasks, that there was a need for documents to decay out of the visible task context, and that our model does a reasonable job of managing that decay, although further tuning could be beneficial.

Since 2007, the operations of a software development company with which the authors are principles has used a task-focused desktop approach to manage the operations of the company. To gain more understanding of the shape of task contexts over time, we performed a case study of three knowledge workers use of the operations repository for Tasktop Technologies. Analyzing how these knowledge workers used task contexts over years, we found additional data supporting the need to decay information from task contexts, we learned these workers used between three and four task contexts a day on average and that many task contexts were re-accessed over many days. We also learned more about the need for both file and web documents in task contexts.

This paper makes three contributions. First it demonstrates that contexts for tasks involving file and web documents can be trimmed automatically based on the frequency and recency of access to a document. Second, it provides initial information about how knowledge workers use both file and web documents as part of tasks. Third, it provides data about how often a small group of knowledge workers use task contexts over years.

Related Work

Early work in supporting task models and task switching focused on the definition of explicit structure for the tasks (e.g., (Card and Henderson 1986)). Other systems have applied filters across information to impose task organizations (e.g., (Bellotii et al. 2003)). Our approach is most similar to those that make the association of information with a task and the determination of the tasks themselves as unobtrusive as possible (e.g., UMEA (Kaptelinin 2003), TaskTracer (Dragunov et al. 2005) and CAAD (Rattenbury and Canny 2007)). In UMEA and

TaskTracer, the information associated with a task increases with interaction unless a user takes explicit action to delete information from the task. Various mechanisms for determining relevance of a piece of information with a task is discussed for these two systems, but little empirical evidence of experience with these mechanisms is provided. The CAAD system attempts to infer both task boundaries and the information associated with a task based on the patterns in which information is accessed together by a user. Data from a short-term 10-person study of the use of CAAD suggests that while the discovered artifact groupings provide utility for users, the most common edit event applied by a user to a grouping was to delete an artifact. Our work differs from these previous efforts in auto-trimming what information is shown to users about a task. Our approach computes degree-of-interest values based on the frequency and recency with which a user interacts with information during a task to determine which information is still interesting for a task and which can be elided in displays of a task's context.

Lettkeman and colleagues also considered trimming the documents associated with a task automatically (Lettkeman et al. 2006). They tested within the TaskTracer environment whether machine learning could predict which web pages accessed as part of a task were likely to be revisited. Their work did not consider file interactions as part of tasks. In the future it would be interesting to compare the precision of a decay feature based on interaction with a statistical approach.

Task-Focused Desktop

Our task-focused desktop includes a task list populated by shared tasks queried from a task repository, from email or created by a user locally (Figure 1-1), a viewing and editing pane for resources associated with a task (Figure 1-2), and a tree view that provides access to local files, shared files and web pages accessed as part of a task (Figure 1-3). The tree view can show all available elements or be filtered to show only those in the task context as in Figure 1. When a knowledge worker starts working on a task, he activates the task by pressing the round button to the left of the task name. This activation initiates monitoring to enable the association of interaction with the active task. The more a document is worked with, the higher the interest value associated with the document. When a document passes a threshold, it becomes a landmark and appears bold in the tree view (e.g., mylar-chi-study.doc in Figure 1-3). Documents that are not accessed gradually decay until they pass a threshold that removes them from the display. A web browser is available from the toolbar; the browser appears in the viewing and editing pane, allowing searches to desired web

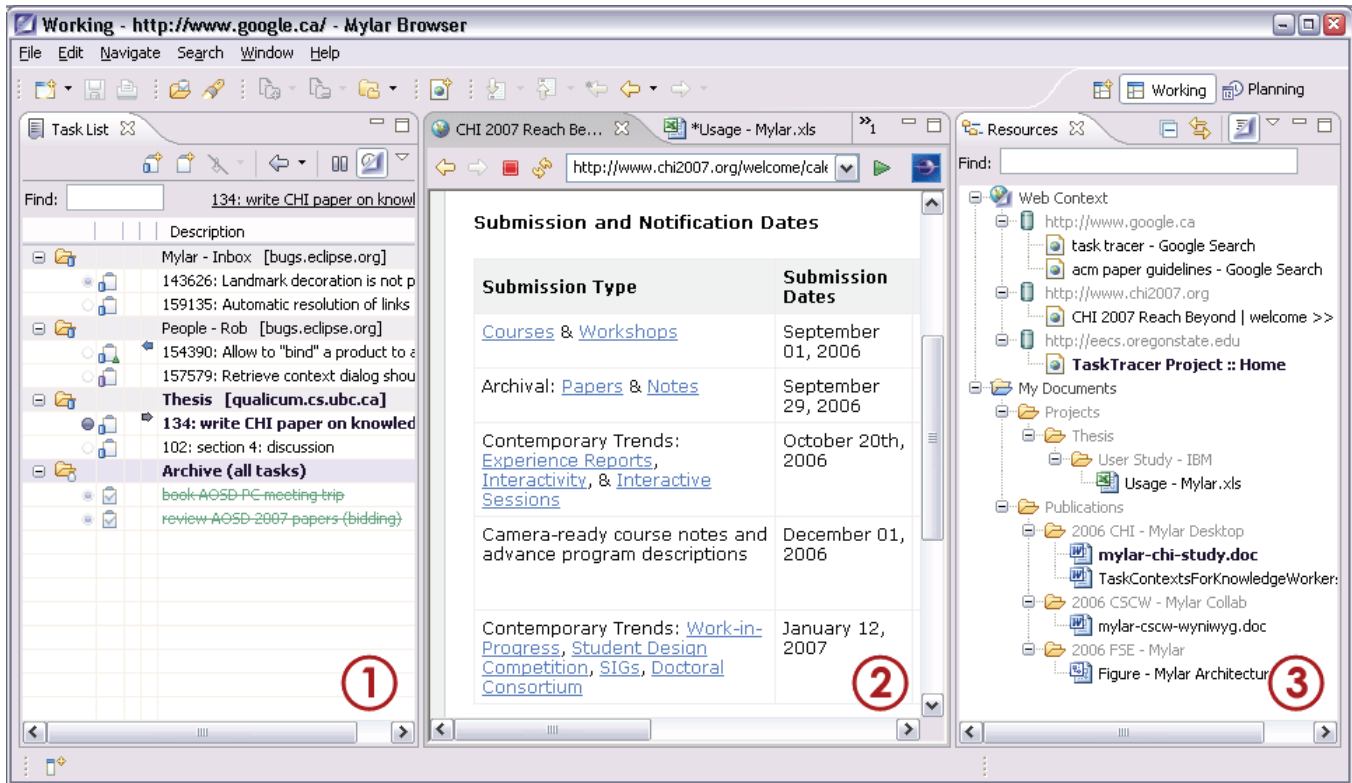


Figure 1 A Task-Focused Desktop includes 1) a task list, 2) a view/edit pane and 3) the resources in the context.

pages. MS Office documents can be viewed and edited either within the viewing and editing pane or in a separate window. Other kinds of documents accessed appeared in a separate window.

Three kinds of events contribute to the interest value of a document in the task-focused desktop: selections, edits and commands. Each event contributes a different scaled value to a document's degree-of-interest value based on the event's kind. A detailed description of our model, interaction history collection and monitoring is provided elsewhere (Kersten and Murphy 2006).

Only one task may be activated at a time in the task-focused desktop. A task is activated with one-click on a button associated with the task in the task list or the task's editor. A task activation closes all documents associated with a currently activated task and refocuses the tree view on all documents associated with the newly activated task. Documents opened in external windows are not closed automatically on a task switch.

Field Study

We performed a field study with an early version of the task-focused desktop in 2007 to determine if our model captures the context of tasks for knowledge workers working with semi-structured documents. We chose a field study because our model is intended to help workers with

tasks that are more long-lived than a simple edit of one or two documents or a single browsing session. We wanted to gain knowledge about three questions: 1) do knowledge workers find it useful to create contexts for their tasks that mix file and web documents, 2) do they revisit those tasks, and 4) does a decay based on frequency and recency of accesses retain the 'right' information in a context?

Participants

We targeted knowledge workers within and related to our university. We advertised the study to approximately two dozen individuals who had heard of the programming tool (but whose work does not include programming), resulting in eight participants who had the tool installed on their work computer and who answered usage questions via email and in person: P1 (technology assessment), P2 (Masters student), P3 (teacher), P4 (instructor), P5 (CEO), P6 (Project coordinator), P7 (Student coordinator), and P8 (Communications coordinator).

Method and Data

Participants were asked to use the tool as they saw fit for their daily work. Each was given a 30 minute tutorial on how to use the tool by the first author. Questions that arose about the usage of the tool during the study were answered by email and in person. Participants were provided updates

of the tool to fix bugs. As participants joined the study over a period of six weeks, the period over which we collected data for each participant varied.

Results

Table 1 summarizes the task activity of participants during the study. The broad spectrum in the number of active workdays is in part due to participants joining near the end of the study (P2, P4, P5), and in one case due to a lack of use of the tool (P7). The total tasks activated column reports the tasks defined within the task-focused desktop for which contexts were activated. The tasks/per day column reports how many unique tasks were activated per day on average. The activations/day column shows how many times per day work occurred on a task on average. For all but P7, the activations/day values are higher than the tasks/day, indicating that participants returned to tasks on which they had worked earlier that day. This data indicates that subjects do revisit tasks and that they used task context for the purpose of restoring context when multitasking.

Table 1 Activity of Tasks with Context in Field Study

	Active workdays	Total tasks activated	Tasks/day	Activations/day
P1	25	26	5.0	7.0
P2	5	5	2.8	9.0
P3	8	22	5.6	11.0
P4	4	9	2.8	3.8
P5	4	5	2.5	6.8
P6	26	41	1.8	2.2
P7	1	1	1.0	1.0
P8	14	29	3.4	3.8

Table 2 describes the content of the tasks with contexts across participants. The data indicates that all but one participant (P7) had tasks involving both web and file documents. This data also highlights the wide variation in context size from those containing just a few to those with hundreds of documents.

Table 3 describes the content of the participants' task contexts at the end of the study by the type of document. The L column presents the percentage of web or file documents with landmark (very high) interest; the I column presents the percentage with interest that keeps the document visible in the desktop; and D represents the percentage of documents that have decayed out of view. Six of the participants generated enough interaction to

result in substantial decay in their web documents, while five participants experienced decay in their file documents. For P1 and P5, on average, both more web and file documents decayed than were retained. The high-values of decay (over 20% for most participants) indicate the need to weigh the relevance of elements in the context to avoid overloading the user with information. We asked each participant if the context model showed too much information (three participants), too little information (two participants) or just about the right information as they worked (three participants). In general, the participants' comments indicate that initial exploration tends to result in too many web documents in the context at the start of a task. Only one participant commented that web documents would sometimes disappear too quickly when re-activating a context. Two participants commented that in the presence of substantial web navigation files decayed too quickly. This situation occurred because more interaction events were generated navigating web documents than files.

Table 2 Contents of Task Contexts in Field Study

	Contexts with web			Contexts with files			Tasks with both
	Tasks	Avg size	Max size	Tasks	Avg size	Max size	
P1	7	69.1	239	16	42.7	317	6
P2	4	5.3	8	5	4.0	14	2
P3	9	17.8	96	2	3.0	4	1
P4	2	3.0	4	1	3.0	3	0
P5	5	32.2	125	4	68.8	225	4
P6	5	5.2	12	2	2.0	2	1
P7	1	5.0	5	0	0	0	0
P8	9	4.9	13	3	4.7	9	2

Table 3 Context Model Accuracy in Field Study

	Web (%)			Files (%)		
	L	I	D	L	I	D
S1	1	10	89	3	8	89
S2	0	73	27	5	50	45
S3	1	30	69	45	55	0
S4	0	100	0	0	75	25
S5	2	13	85	2	19	79
S6	4	96	0	0	1	0
S7	20	60	20	0	0	0
S8	3	52	48	3	34	63

Threats

The participants in our study may not be representative of any interesting class of knowledge workers. Given that the participants fulfill a variety of roles, the results may generalize to an interesting population although further study with a better sampling approach across a wider population is warranted. The participants also may have only used the desktop for tasks that were well-suited to the tool. We do not know what percentage of their daily tasks for which they used the tool. The heavy use of the tool by some participants suggests that it does fit some work patterns. One participant (P1) even stopped using MS Outlook task management in favour of the task-focused desktop. Future studies should consider which tasks our task context model supports and which it does not and why.

Case Study

To provide insight into the nature of task contexts over a longer usage period, we performed a study of the use of task-focused technology to run the operations of a software company, Tasktop Technologies, of which the authors are co-founders. In this organization, a shared task repository is used to communicate and collaborate about all aspects of business operations, including aspects of sales, business development, finance, human resources, marketing and general administration. Operational tasks can be defined in a shared task repository, in which case the task is visible to others, can be assigned to different individuals, can be used to host discussions between collaborators, and can be used to share contexts consisting of file and web documents. Operational tasks can also be defined personally either as named local tasks or as tasks based on email messages. The shared task repository has been operational for three years and eight months and currently hosts 8042 tasks of which 5707 (70.9%) have been marked as completed. The average number of comments left on tasks as people collaborate upon the tasks is 8.14 ± 13.63 . In this shared task repository, 2404 (28.6%) of the tasks have some sharing of files and or web documents between users; this percentage does not include artifacts captured as contexts personally by individuals.

To gain insight into how individuals use task contexts over a longer period of time than in our previous study, we selected a convenience sample of three individuals using this shared task repository. We analyzed data collected as these individuals worked over periods ranging from just over 300 to almost 600 days to answer the following questions:

- How often do the individuals activate tasks? Activating a task triggers the capture of task contexts suitable for focusing the user interface to show only those file and web documents associated with a task, for

immediate re-access to those documents upon reactivation of a task and for enabling sharing of contexts with other collaborators.

- How often do the individuals reactivate a task? Reactivating a task is an indication of a desire to re-access the context associated with a task.

- Does decay of files or web documents from task contexts occur? The decay of documents indicates a need for trimming contexts automatically.

- What is the mixture of files and web documents captured in task contexts? This mixture provides data towards the nature of task contexts used by knowledge workers.

Table 4 summarizes the data about task reactivations from the three individuals. The data for P1 and P2 represents work associated with the operations of the software business. The data for P3 includes work both for the operations of the software business and for academic work. In Table 4, the column labeled “# Days” refers to the total number of days in the analyzed data an individual activated at least one task. There may be more days that the individual considered information on tasks without activating the capture and recall of tasks contexts. The second column, labeled “# Task Act.” Indicates the number of unique tasks the individual has activated. The third column, labeled “# Act.” presents the total number of activations of tasks. The last four columns presents the average and maximum number of unique tasks activated in a given day, providing a lower bound on the number of tasks on which an individual worked and the average and maximum number of days on which a task was reactivated, which indicates the longevity of tasks. The data in Table 4 shows a range in the number of task activations per day; interestingly, each individual had at least one day in which a large number (i.e., greater than 15) tasks were activated. All individuals also had tasks for which they continued to work and refer and access previously captured task contexts.

Table 4 Use of Tasks in Case Study

	# Days	# Task Act.	# Act.	Avg. Act. /Day	Max Act. /Day	Avg. Re-act.	Max Re-act.
P1	311	617	3819	8	24	4	85
P2	308	225	1217	2	16	4	93
P3	598	717	2796	3	19	3	148

Table 5 displays data about the kinds of task contexts used by the individuals. The table presents the average and maximum number of interesting files per context compared to the average number of files per context. An interesting

file is one whose degree-of-interest value is larger than a threshold used to trim the display of files from the view of a user. For each user, the average number of files per context exceeds the average interesting files per context, suggesting decay was effective in trimming the individual's task context. The table also presents the average and maximum number of interesting web pages per context compared to the average number of web pages. Similar to the files, the data shows decay was useful in trimming web documents from an individual's task context. This data also sheds light on the mixture of files and web pages in individuals' task contexts. Two individuals (P1 and P3) had more mixed use of files and web pages than the third individual (P2). The mixed use suggests there is a need to support modeling of multiple artifact types across task contexts.

Table 5 The Nature of Task Contexts in Case Study

	Avg. Int. File/ Con.	Max Int. File/ Con.	Avg. File/ Con.	Avg. Int. Web/ Con.	Max Int. Web/ Con.	Avg. Web/ Con.
P1	4	102	12	5	46	10
P2	10	114	16	0	25	1
P3	5	116	13	1	28	2

Threats

This study reports on a convenience sample of users. These users may not be representative of a broad spectrum of knowledge workers. The data was analyzed retrospectively so it may be more indicative of common work patterns as the individuals were not aware the data had been stored in a way amenable to later analysis. The many changes that occurred in the tooling used to collect the data over a three year time period may have led to noise in the data analyzed.

SUMMARY

Through a field study and a longitudinal case study, we have demonstrated that the daily work of knowledge workers performing various knowledge worker roles includes tasks that involve both file and web documents, that these workers revisit tasks within a single work day, and that there is a need to automatically trim the contents of tasks as they work. The task context model used in the study performs such trimming by decaying documents from view based on the frequency and recency of accesses to a document. This model seems to perform adequately; all participants in our field study said they would continue using the tool. Our results suggest that an interaction-based

weighting of information provides an alternative to machine learning approaches for ensuring a task context captures the information most relevant to a task. Further study is needed to determine which approach works best and in under what conditions. This study also provides initial data about the breakdown of task contents between files and web documents. We are not aware of any existing studies that have provided such a breakdown.

Acknowledgements

We thank the participants in our studies and gratefully acknowledge funding support from NSERC and IBM.

References

- Bellotti, V.; Ducheneaut, N.; Howard, M. and Smith, I. 2003. Taking Email to Task: The Design and Evaluation of a Task Management Centered Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 345-352. New York, NY: ACM.
- Card, S.K. and Henderson Jr., A. 1986. A Multiple Virtual-Workspace Interface to Support User Task Switching. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics*, 53-59. New York, NY: ACM.
- Dragunov, A.N.; Dietterich, T.G.; Johnsrude, K.; McLaughlin, M.; Ji, L. and Herlocker, J.L. 2005. TaskTracer: A Desktop Environment to Support Multi-Tasking Knowledge Workers. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, 75-82. New York, NY: ACM.
- González, V.M. and Mark, G. 2004. Constant, Constant, Multi-tasking Craziness: Managing Multiple Working Spheres. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 113-120. New York, NY: ACM.
- Kaptelinin, V. 2003. UMEA: Translating Interaction Histories into Project Contexts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 353-360. New York, NY: ACM.
- Kersten, M. and Murphy, G.C. 2006. Using Task Contexts to Improve Programmer Productivity. In *Proceedings of the 14th ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, 1-11. New York, NY: ACM.
- Lettkeman, A.T.; Stumpf, S.; Irvine, J. and Herlocker, J. Predicting Task-Specific Webpages for Revisting. 2006. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 1369-1374. Menlo Park, CA; AAAI Press.
- Stumpf, S.; Bao, X.; Dragunov, A.; Dietterich, T.; Herlocker, J.; Li, L. and Shen, J. 2005. Predicting User Tasks: I Know What You're Doing. In *Proceedings of the 20th National Conference on Artificial Intelligence Workshop on Human Comprehensible Machine Learning*. Menlo Park, CA: AAAI Press.
- Rattenbury, T. and Canny, J. CAAD: An Automatic Task Support System. 2007. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 687-696. New York, NY: ACM.