

Temporal Task Footprinting: Identifying Routine Tasks by Their Temporal Patterns

Oliver Brdiczka

Palo Alto Research Center (PARC)
3333 Coyote Hill Rd, Palo Alto, USA
brdiczka@parc.com

Norman Makoto Su

Department of Informatics
University of California Irvine, USA
normsu@ics.uci.edu

James “Bo” Begole

Palo Alto Research Center (PARC)
3333 Coyote Hill Rd, Palo Alto, USA
begole@parc.com

ABSTRACT

This paper introduces a new representation for describing routine tasks, called *temporal task footprints*. Routines are characterized by their temporal regularity or rhythm. Temporal pattern analysis (T-patterns) can be used to isolate frequent recurrent patterns in routine tasks that appear repeatedly in the same temporal configuration. Using tf-idf statistics, each task can then be defined in terms of its temporal task footprint, a ranked list of temporal patterns along with their typical frequencies. Experimental evaluations using data of 29 days observing and logging 10 subjects showed that temporal task footprints of application windows, email and document usage outperform decision tree and SVMs in recognizing the subjects' tasks.

Categories and Subject Descriptors

H.1.2 User/Machine Systems: Human factors; H.5.2 User Interfaces: Theory and methods

General Terms

Algorithms, Experimentation, Human Factors, Measurement.

Keywords

T-patterns, Temporal patterns, routine task representation, task footprint.

1. INTRODUCTION

Knowledge workers are often involved in multiple tasks that they do on a regular basis and between which they need to switch frequently [9]. As a consequence, task management, maintenance of task context and recovery from interruptions become important problems that a knowledge worker needs to face on a daily basis. User interfaces and systems for managing tasks (e.g., [5]) have been proposed recently to address some of these issues. However, little work has been done on representing the regularity of tasks themselves, i.e. addressing what is commonly designated as “routine” work and its representation as well as automatic recognition.

This paper introduces *temporal task footprints*, a new representation for characterizing routine tasks by incorporating structural time information. A routine task refers to a recurrent activity that is composed of a set of actions to accomplish a job, problem or assignment. Temporal task footprints are not only a good means for recognizing and distinguishing different routine tasks, but can also serve as a *human understandable* representation of tasks for

constructing and sharing workflow knowledge in a company. The temporal task footprints constructed and evaluated in this paper are based on temporal patterns from application and document usage as well as from communication with co-workers.

Routines as a unit of analysis have been examined from an organizational perspective. Feldman and Pentland [6] state the “standard” definition of organizational routines as “repetition, a recognizable pattern of action, multiple participants, and interdependent actions.” Routines are repeated over time, seem to be a collection of activities one can categorize, involve multiple people, and are situated in the context created by one’s self and other people.

The temporal regularity (its rhythm) of routines, and how awareness of such rhythms can facilitate work has been analyzed in HCI and CSCW research. Begole et al. [1] point out that work patterns differ across time, location, and day of week. By examining past, recurring work rhythms, one can predict future presence based on current events. One can guess, for example, the amount of time needed for a certain individual to prepare and leave for an appointment, prepare for meetings, commute to work, return from lunch and other patterns not captured in an individual’s online schedule.

Reddy and Dourish [12] conducted ethnography at a hospital to examine how people use work rhythms to accomplish information seeking. For example, rhythms can provide valuable information between nurses and doctors. The regular rotation of doctors at intervals allows nurses to simply wait, rather than waste resources and time when he or she needs to seek a physician. Finally, specific usage of certain media such as email [3] have been observed to have rhythms (e.g., at the beginning of the day). However, the analyses that have examined the temporal aspects of routines have not addressed *how this temporal information can be used to describe and detect routine tasks*.

Much work has been done in academia concerning automatic task prediction using machine learning methods. The TaskTracer system [5] that has been built within the DARPA CALO project employed task prediction approaches that focus on the selection of different features that are observed and discriminative for each task. After feature selection, tasks are predicted using, for example, a hybrid approach of a naïve Bayesian classifier and support vector machines [14]. Even though recent approaches also focus on the discovery of work procedures or routines [15], the *temporal* information, such as significant duration between task steps that is characteristic for routine tasks, is not considered.

The Microsoft SWISH system [10] constructs a model of a user’s tasks based on tf-idf filtered terms in the title of application windows and the windows switching history. A number of clusters of these filtered terms are isolated, corresponding to the user’s tasks. This cluster representation is then used for recognition. Again, no

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUI’10, February 7–10, 2010, Hong Kong, China.

Copyright 2010 ACM 978-1-60558-515-4/10/02...\$10.00.

duration information between windows is used to characterize a task; windows switching analysis is purely based on building a connected graph from the *sequence* of switches (within a predefined time window T). We believe, however, that duration information is crucial when characterizing routine tasks because these tasks are precisely characterized by their temporal regularity and structure.

This paper intends to build a new task representation that is based on a distribution of temporal patterns characterizing a (routine) task. The employed temporal pattern analysis (T-patterns) is used to isolate frequent recurrent patterns in tasks that appear repeatedly in the same temporal configuration; in other words, the duration between each step remains relatively constant. The *tf-idf* [13] values of the resulting T-patterns with respect to the routine tasks allows for generating a ranked list of relevant patterns for each task, the temporal task footprint.

The remainder of our paper is structured as follows. First, we briefly introduce the temporal pattern detection (T-pattern analysis) as well as its parameters and outcomes. Based on the detected temporal patterns, the temporal task footprint is defined. Following that, the paper describes the collected data set and the experimental evaluation which has been conducted. A conclusion section sums up this paper.

2. TEMPORAL PATTERN DETECTION

We assume that routine tasks can be characterized by specific recurrent actions that are executed within nearly constant time intervals. In order to detect such patterns, we use a probabilistic temporal pattern detection method, called T-pattern detection [8]. The T-pattern detection algorithm identifies highly significant, hierarchically arranged temporal patterns that are composed of statistically related events that repeatedly appear in the same, relatively invariant, sequence and temporal distance.

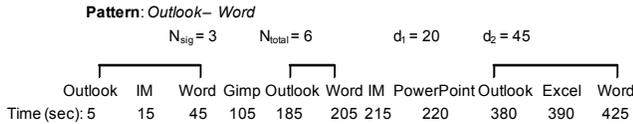


Fig. 1. T-pattern analysis detecting a pattern of the applications Microsoft Word following Microsoft Outlook from 20 to 45 sec.

T-patterns provide significant advantages over traditional sequence analyses by incorporating temporal distance. Indeed, traditional sequential pattern mining techniques [7] or compression-based algorithms (e.g., Lempel-Ziv-Welch) can discover sequential pattern. However, these methods do not take into account the temporal distance among elements of the patterns since the time delays are not modeled. Markov models [11] are not suitable for this problem either, as the first order Markovian assumption does not hold because patterns are constructed as long ($n > 2$) sequences. In addition, Markov model have problems handling patterns that have very long time intervals.

Our implementation of the T-pattern detection algorithm is based on the description by Magnusson [8]. The algorithm identifies a number of T-patterns (N_{sig}) that are significant per task. In addition, the significant minimal and maximal temporal length (d_1 and d_2) for each T-pattern is reported, that is, if A is an earlier component and B is a later component of the same recurring T-pattern, then, after an occurrence of A at t , there is an interval $[t + d_1; t + d_2]$ ($d_2 > d_1 > 0$) that contains at least one occurrence of B. Fig. 1 shows an example

where the use of desktop applications (Outlook, Word, IM, Gimp, etc.) forms the input events. In this example, the T-pattern Outlook-Word is found to have three significant occurrences ($N_{sig} = 3$), marked in the figure with intervals ranging from 20 to 45 seconds. There are additional occurrences of the pattern Outlook-Word (e.g. Outlook at 5 and Word at 205 seconds), but these have not been detected to be significantly recurring. The number of significant occurrences ($N_{sig} = 3$) is a subset of all occurrences ($N_{total} = 6$) of the pattern Outlook-Word.

The T-pattern algorithm can be run on a wide range of time stamped events. In general, these are actions that are executed by a human and reflect his or her behavior in time. In the remainder of this paper, we will focus on PC usage events, i.e. events coming from application windows, window positions, document usage, and email (sender, recipients).

3. TEMPORAL TASK FOOTPRINT

By running the T-pattern analysis on one or several event traces that can be observed when a user works on a routine task, we can extract a number of significant patterns for this task. Table 1 shows a list of such patterns from PC application usage for the task “Expense Reporting.” The table reports the pattern, its significance value, number of occurrences (N_{sig}), and minimal as well as maximal intervals (d_1, d_2 in ms).

Pattern	Significance	N_{sig}	d_1	d_2
FIREFOX.EXE - MsgBox - EXCEL.EXE	3.495e-08	3	1985	16812
ACROBAT.EXE - OUTLOOK.EXE - ClipBoardChange	2.666e-07	2	874	3110
OUTLOOK.EXE - ClipBoardChange - WORD.EXE	1.332e-08	4	1709	13858

Table 1. List (excerpt) of significant T-patterns extracted for routine task “Expense Reporting”

The goal of the temporal task footprint is to represent the routine task by incorporating its temporal structure (i.e. its T-patterns) and to make it easily distinguishable from other similar routine tasks.

A basic representation of a routine task could consist of the raw frequencies of occurrence of its significant event patterns. However, this representation has the disadvantage that commonly occurring patterns may unnecessarily make all tasks look similar because their frequencies outweigh those that are characteristic for a particular routine task. To address this problem, we can make use of the term frequency-inverse document frequency measure *tf-idf* [13]. The *tf-idf* measure is a weight often used in information retrieval and text mining used to evaluate how important a word is to a document in a collection or corpus of documents. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus (e.g., “the” is weighted less than “proposal” in the corpus). In order to adapt *tf-idf* to our problem, we replace terms by pattern frequencies and document by task frequencies. Thus,

$$tf_{p,T} = \frac{n_{p,T}}{\sum_k n_{k,T}}$$

refers to the normalized occurrence

frequencies of pattern p in task T , while the *idf* measure for a pattern

$$idf_p = \log \frac{|T|}{(|T_p| + 1)}$$

is given by

where $|T|$ is the total number of tasks and $|T_p|$ the number of tasks where the pattern p has

been observed. The tf-idf measure for a pattern p and task T is defined as: $tfidf_{p,T} = tf_{p,T} * idf_p$. Intuitively, tf-idf scales down commonly occurring patterns and scales up patterns which rarely occur in tasks and therefore are *distinctive* for a task.

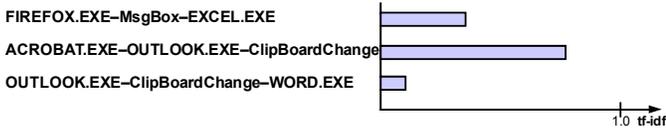


Fig. 2. Temporal task footprint for the task “Expense Reporting.”

A temporal footprint for a task T is finally defined as a list of temporal patterns p that are weighted by their tf-idf values $tfidf_{p,T}$. Fig. 2 shows an example task footprint for the data from Table 1. Even though the second pattern has the lowest occurrence frequency in the task, it has the highest tf-idf value because it seems to be unique to this task and has not been seen with any other task.

Given a collection of tasks represented by their temporal task footprints, a new candidate task C that has been observed can be recognized/assigned to a task in the collection by evaluating $\arg \max_i [tfidf_score(C, T_i)]$, where T_i refers to the routine task footprints in the collection. The $tfidf_score$ is defined by

$$tfidf_score(C, T_i) = \sum_{p=1}^N (tfidf_{p,T_i} \cdot \delta_{p,C}) \text{ where } N$$

refers to the numbers of patterns in the task collection footprint T_i . $\delta_{p,C}$ is 1 if pattern j has been seen in candidate task C and 0 otherwise. The basic idea here is to add up the $tfidf_{p,T_i}$ values only for those patterns p that are shared by T_i and the candidate task C . Finally, the routine task footprint with the highest $tfidf_score$ provides the label for the new observed candidate task. In sum, we have adapted tf-idf from measuring the importance of a word in a document, to measuring the importance of a T-pattern (e.g., a temporal pattern of application usage) in a task.

4. EXPERIMENTAL EVALUATION

4.1 Data set

Data was collected in situ by human observation and computer logging of 10 employees working at a research and development company. The human observation allowed us to establish the correct task labels for our time-stamped computer logging data. Table 2 shows the distribution of the informants and their different functions in the company. All but one informant (who was observed for only two whole days due to scheduling constraints) were observed for three whole work days. The observed days were not contiguous; each was separated by a period of one week to one month. The days were selected to contain at least one recurring task. However, the selected recurrent tasks only constituted a small subset of the complete task set that was observed for each user. The remaining tasks could recur during all days of observation (but we did not influence their recurrence in any way). The observer would meet the informant upon their arrival to work and follow the informant as closely as possible until the end of the business day. Using a paper notepad with an electronic LiveScribe pen (www.livescribe.com), the observer would record and label, to the second, user tasks and their start/end times (e.g., 8:15:32am start “Expense Reporting”). A total of 572 different tasks have been observed and labeled, most of which were not been shared across informants. Prior to and during

the observation, logging software was installed on each participant’s PC to record application usage.

Accounting Staff	3
IT Staff	2
Library Service Staff	2
Intellectual Property Staff	1
Lab Administration	1
Researcher	1

Table 2. Distribution of informants within the company

The application usage record included: application window, application window position, active document and email sender-recipient. Application window refers to the current application (e.g. EXCEL.EXE) and the current window information (e.g. ConsoleWindow or MessageBox or InternetExplorerPanel1). Note that one normally has several window classes per application; this provides a higher granularity observation of the user’s actions. The positions of the application windows (x, y, width, height) were discretized using EM clustering [2]. The cluster numbers of the current application window positions are the input events for the T-pattern analysis. Active document refers to the document that the user is currently working on. Email sender and recipient refer to the event when the user receives, sends or selects an email. Sender and recipient ID of this email are then the input events of the T-pattern analysis.

4.2 Evaluation

The aim of this evaluation was to show that the temporal duration information incorporated in T-patterns and subsequently in temporal task footprints is beneficial for task recognition. Therefore, temporal task footprints have been constructed for the collected data set and compared to the recognition performance of state-of-the-art machine learning algorithms that utilize non-temporal features when recognizing tasks. T-pattern analysis was conducted separately for each task using the data fields of the logging data trace.

We conducted a 3 fold cross-validation, taking one third of a participant’s data (= 1 day) for training/constructing the task footprints and two thirds (= 2 days) for testing. In addition to our temporal task footprints, decision trees (J48 [16]) and support vector machines (LibSVM library [4] using radial basis kernel) were employed to learn the task labels from the respective inputs (application window classes, window positions, document usage, email). The obtained classification accuracies are depicted in Table 3. Except for window position patterns, temporal task footprints outperform both decision trees and support vector machines when recognizing the task labels. This is due to the fact that temporal task footprints leverage the temporal information (duration information encoded by the T-patterns) that is not used by J48 and SVMs. This temporal structure information *significantly* improves task recognition except for window positions and sizes. Duration information between different positions and sizes of application windows does not seem to provide much additional information that might improve classification. The best performance for task recognition is obtained when using patterns from document usage. Apparently, sequences of documents and specific durations associated with these are very discriminative between different tasks.

	Temporal Task		
	Footprints	J48	SVM
Application Window Class	51.34%	38.33%	36.93%
Window Positions	26.29%	29.01%	31.44%
Document Usage	63.36%	40.38%	37.38%
Email (sender, recipients)	35.72%	19.21%	24.03%

Table 3. Task recognition accuracies for temporal task footprints, decision trees (J48) and support vector machines (SVM).

The overall low recognition accuracies can be explained by the fact that an ethnographer collected and transcribed the users' tasks in extreme detail. The resulting ground truth therefore contains a large number of task labels and distinguishes tasks at a much lower granularity than most previous work. To illustrate, our data set involved 10 people with 572 different task classes, while TaskTracer [14] used a data set of 8 people with only 107 task classes. Our data set was intended to be very complete and thorough, providing thus some kind of baseline for comparison. We believe that a less fine-grained distinction of tasks (i.e., creating 5-10 routine tasks to be trained and learned per user in a system) will yield better results.

5. CONCLUSION AND FUTURE WORK

This paper introduced a new concept for characterizing routine tasks, called temporal task footprints. Temporal task footprints are based on the assumption that routine tasks are characterized by temporal regularity and repetition of steps that are executed. T-pattern analysis is used to extract these repetitive temporal patterns that are characterized by a relatively constant temporal configuration. The temporal task footprint is further defined by the tf-idf values of the extracted temporal patterns with respect to the corresponding routine tasks. While these footprints can be calculated for a wide variety of tasks and events that are observed, this paper has focused on events from PC usage and routine tasks that have been executed on PC. Our experimental evaluation involving observation and logging data of 29 days in total from 10 different subjects has shown that temporal task footprints are effective in recognizing routine tasks and outperform standard machine learning methods in this context. This is partly due to the fact that temporal task footprints encode temporal duration information of the observed tasks by incorporating T-patterns. This information is quite discriminative, especially when distinguishing routine tasks that are based on a similar range of events that are observed.

Because temporal task footprints provide a human-readable format for understanding patterns in routine tasks, their utility may go beyond simple task recognition. Temporal task footprints could provide a new way to describe and ultimately share information about routine work. The incorporation of temporal information (T-patterns) in a task representation provides an efficient means to recognize and re-use work routines. Each routine task pattern that is identified may further contain links from its events to related content (e.g., document files or emails). Users can then review their own and their colleagues' patterns, edit and add content (e.g., exchange template files), and execute and re-apply these temporal patterns. In particular, the obtained pattern representations and visualizations can contribute to the re-use of knowledge workflows in a company when training new employees or sharing new work processes with co-workers. By recognizing which routine a user probably engages in, a collection of related patterns can be dynamically recommended. Hence, this pool of temporal task

footprints along with the contained, human-readable, temporal patterns can serve as a data base for defining and sharing workflow information within an organization.

6. REFERENCES

- [1] J. B. Begole, J.C. Tang, R. B. Smith, N. Yankelovich. Work rhythms: Analyzing visualizations of awareness histories of distributed groups. In Proc. of CSCW'02, 334-343, 2002.
- [2] J. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models, Technical report TR-97-021, U.C. Berkeley, April 1998.
- [3] V. Bellotti, B. Dalal, N. Good, P. Flynn, D. Bobrow, N. Ducheneaut. What a to-do: Studies of task management towards the design of a personal task list manager. In Proc. of CHI'04, 735-742, 2004.
- [4] C.C. Chang and C. J. Lin. LIBSVM: a library for support vector machines, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [5] N. Dragunov, T.G. Dietterich, K. Johnsrude, M. McLaughlin, L. Li, J.L. Herlocker. TaskTracer: a desktop environment to support multi-tasking knowledge workers. In Proc. of IUI '05, 75-82, 2005.
- [6] M.S. Feldman, B. T. Pentland. Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly* 48, 1 (Mar.), 94-118, 2003.
- [7] J. Han, H. Cheng, D. Xin, X. Yan. Frequent pattern mining: current status and future directions. In *Data Mining and Knowledge Discovery* 15:55-86, 2007.
- [8] M. S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments, & Computers*, 32(1):93-110, 2000.
- [9] G. Mark, V.M. Gonzalez, J. Harris. No task left behind?: examining the nature of fragmented work. In Proc. of CHI'05, 321-330, 2005.
- [10] N. Oliver, G. Smith, C. Thakkar, A. C. Surendran. SWISH: semantic analysis of window titles and switching history. In Proc. of IUI '06, 194-201, 2006.
- [11] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In Proc. IEEE 2: 257-286, 1989.
- [12] M. Reddy, P. Dourish. A finger on the pulse: Temporal rhythms and information seeking in medical work. In Proc. of CSCW'02, 344-353, 2002.
- [13] G. Salton, M. McGill (eds). *Introduction to Modern Information Retrieval*. McGraw-Hill. 1983.
- [14] J. Shen, L. Li, T.G. Dietterich, J. L. Herlocker. A hybrid learning system for recognizing user tasks from desk activities and email messages. In Proc. of IUI'06, 86-92, 2006.
- [15] J. Shen, E. Fitzhenry, T.G. Dietterich. Discovering frequent work procedures from resource connections. In Proc. of IUI'09, 277-286, 2009.
- [16] H. Witten, E. Frank. *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, San Francisco, 2005. Software available at <http://www.cs.waikato.ac.nz/ml/weka/>