A Context-based Approach to Discover Multitasking Behavior at Work

Marielba Zacarias
Instituto Superior Técnico de Lisboa
Universidade do Algarve,
CEO-INESC Lisboa, Portugal
(00351)289800900
mzacaria@ualg.pt

H. Sofia Pinto
Instituto Superior Técnico deLisboa
ALGOS, INESC-ID Lisboa, Portugal
(00351)213100338
sofia@algos.inesc-id.pt

José Tribolet
Instituto Superior Técnico de Lisboa,
CEO-INESC Lisboa, Portugal
(00351)213572957
jose.tribolet@inesc.pt

ABSTRACT
Despite the availability of several task and personal information management tools, an appropriate support to human multitasking at work is still lacking. Supporting multitasking behavior entails capturing and modeling this behavior. In this paper, we refine an approach to model multitasking behavior in organizations, through an ontology based on two interrelated primitives; *action* and *interaction contexts.* The main contributions of the proposed ontology are: (1) enable the discovery of scheduling heuristics combining personal and inter-personal elements, (2) enable bottom-up discovery of tasks and (3) suggest a flexible system architecture for multitasking support. The first two contributions are illustrated through a case study.

Categories and Subject Descriptors
H1.2 User/Machine Systems, *Human Factors*
H.4.1 Office Automation, *Task, Information & Time management*

General Terms
Design, Human Factors

Keywords
Multitasking Modeling, Context Modeling, Task Modeling, Task management, Personal Information Management.

1. INTRODUCTION AND MOTIVATION
The reality of multitasking at work is undeniable. Evidence can be observed through several case studies [1]. Furthermore, task interleaving is constantly increasing due to the improvement of personal and group information technologies. A diary study of task switching and interruptions of eleven experienced Microsoft Windows users, reported a 50 task switch average during a week, when performing their computer-assisted tasks [2]. Human multitasking capabilities and limitations have been studied in Cognitive Sciences and Experimental Psychology. The literature in multiple-task performance is extensive [3,4,5]. However, there is no consensus around multitasking benefits and costs for businesses. Whereas some view it as an opportunity to draw on human capabilities, build their skills and enhance their productivity [6], others consider multitasking counterproductive [9] due to switching costs, showed by several psychological experiments [10].

In any case, it is a fact that people at work typically handle several, independent and unrelated tasks. Due to scarce resources such as attention and short-term memory [15], it is necessary to focus on a single task at a time. Thus, current work dynamics force individuals to 'break' tasks and 'switch' among them according to criteria encompassing not only task-related factors, but also to resource availability or personal scheduling heuristics. Task-switching at work complicates information provision to workers -already a challenge due to information overload issues- because it requires *personalized* and *timely* mechanisms. Despite the availability of several task and personal information management tools, an appropriate support to human multitasking at work is still lacking [11].

To support multitasking behavior we must capture and model this behavior. In this paper, we propose an ontology to discover multitasking behavior in organizations and illustrate its benefits through a case study. The proposed ontology is based on a model defined in [7], which addresses multitasking in terms of two different but interrelated concepts; *action* and *interaction contexts.* At a particular moment, the specific set of resources used by an individual depend on a combination of personal, task, role, location or time-related factors that define specific *action contexts*. Modeling multitasking entails acknowledging (1) the several action contexts handled by a single individual and (2) how he or she handles these action contexts. Since we rarely work in isolation, multitasking behavior is also influenced by interactions. The importance of interactions on task switching is acknowledged in [2,16]. Drawing from speech act theory [13], interactions produce commitments that are handled according to inter-personal and social rules. We approach interaction contexts as sets of *commitments* between two or more personal action contexts. Commitments are a fundamental factor of action context switching. The relationship of action and interaction contexts with other common modeling primitives is also defined.

The main contributions of this ontology are: (1) enable the discovery of scheduling heuristics combining personal and inter-personal elements, (2) enable bottom-up discovery of tasks and (3) suggest a flexible system architecture for multitasking support. The present work illustrates benefits related to the first two contributions through a case study. The case study results show first, that the action context primitive enables to handle sets of resources as single entities. This approach facilitates the detection of context switches and context switching patterns. Moreover, the analysis of context switching patterns reveals some generic personal heuristics and underlying inter-personal rules. Second, it illustrates that grouping actions and interactions in their corresponding contexts facilitates the detection of recurrent action and interaction patterns and a bottom-up discovery of tasks.

The remaining of this paper is structured as follows; section 2 summarizes work on tools for multitasking support; sections 3 and 4 summarize theories and models supporting this approach. Section 5 defines action and interaction contexts and illustrates these definitions with examples drawn from a case study, section 6 summarizes case study results on personal action contexts and section 7 summarizes related work on modeling approaches for user interface design addressing multitasking. Section 8 gives our conclusions and future directions.

2. MULTITASKING SUPPORT

There are several research prototype tools intended to enhance user support regarding multitasking issues. GroupBar [27] enables to organize project-related documents, e-mails and other windows together in the windows XP toolbar. GroupBar allows users to drag and drop taskbar "tile" on top of each other, forming groups of items in the bar that can then be operated on as a unit. Also, once the user lays out their work in a preferred configuration, GroupBar saves and restores these layouts. ROOMS is another project-oriented tool [28], which allows the user to set up specialized workspaces or "rooms" containing the resources necessary to carry out different types of activities. The use of dedicated work spaces has been limited due to the configuration overhead posed on their users [29]. The system UMEA aims at overcoming these limitations through a systems design which (a) organizes resources into project-related pools consisting of documents, folders, URLs and contacts, (b) monitors user activities and tracks resource usage in each project, (c) automatically organizes and updates these resources to make them easily available to the user when he or she resumes each project. Communication-based environments tools (such as e-mail applications) organize resources around contacts, communication threads or messages. E-mail conduit's function has lead to exploit inboxes for task management [16]. Moreover, its inter-personal nature poses interesting challenges for task management which have not been addressed. Reported limitations of current tools are (1) lack of more intelligent and flexible means for associating user actions and interactions with tasks or projects and enabling an automatic switching between them and (2) exclusion of inter-personal elements. The value of context identification heuristics in overcoming the first limitation has been acknowledged in [29].

From our point of view, the discovery of scheduling heuristics including personal and inter-personal factors are another valuable aid in enhancing current capabilities of associating individual actions and interactions with their respective contexts. Another limitation of most current tools is their inflexible design, which forces to organize actions, interactions or resources according to single pre-defined schemes (e.g. tasks, projects or communication threads). We argue that limitation is due to the lack of an underlying theoretical model of human multitasking. These models may suggest systems architectures capable of providing a more adequate multitasking support.

3. HUMAN MULTITASKING MODELS

This section summarizes two representative cognitive models described in [10]: the attention-to-action (ATA) model and the frontal-lobe executive (FLE) model. These theories show how multitasking may be modeled through a three-layered model were task execution is separated from executive control processes that handle task interleaving and from processes that handle new situations or complex situations.

Attention-to-Action Model: The ATA model has three subcomponents: *action schemas*, *contention scheduling*, and a *supervisory attentional system)*. **Action schemas** are specialized routines for performing individual tasks that involve well-learned perceptual-motor and cognitive skills. Each action schema has a current degree of activation that may be increased by either specific perceptual 'trigger' stimuli or outputs from other related schemas. When its activation exceeds a preset threshold, an action schema may direct a person's behavior toward performing some task. Moreover, on occasion, multiple schemas may be activated simultaneously by different trigger stimuli, creating error-prone conflicts if they entail mutually exclusive responses (e.g., typing on a keyboard and answering a telephone concurrently). To help resolve such conflicts, the ATA model uses **contention scheduling**. Contention scheduling allows task priorities and environmental cues to be assessed on a decentralized basis without explicit top-down executive control. However, this may not always suffice to handle conflicts when new tasks, unusual task combinations, or complex behaviors are involved. Consequently, the ATA model also has a **Supervisory Attentional System (SAS)**. The SAS guides behavior in a top-down manner. It helps organize complex actions and perform novel tasks by activating or inhibiting particular action schemas, superseding the bottom-up influences of contention scheduling and better accommodating a person's overall capacities and goals. Figure 1 illustrates the ATA model.

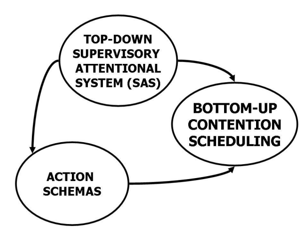

Figure 1. ATA model

Frontal-Lobe Executive Model : Assumptions similar to those of the ATA model have been embodied in the Frontal Lobe Executive (FLE) model. It has three main components: *goal lists, means-ends analysis procedures*, and *action structures*. **Goal lists** represent a person's current set of prioritized intentions. **Means-ends analysis**, like the SAS, updates the contents and order of goals in working memory, considering of how well they are being achieved over time. Supplementing such functions, the **action structures** of the FLE model constitute a large store of procedural knowledge for goal-directed behaviors embodied as sets of condition-action production rules. The conditions of these rules refer to goals and perceptual stimuli; the actions involve responses to achieve the goals (e.g., *"if the goal is to do task a and the stimulus is s, then produce response r"*). Action structures are analogous to the ATA model's action schemas. Figure 2 illustrates the FLE model.

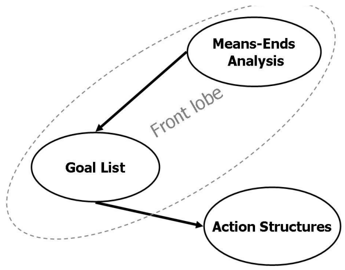

Figure 2. FLE model

4. CONTEXT MODELING APPROACHES

The concept of context is essential to our modeling approach. The notion and models of context vary according to the area of application. This section briefly summarizes engineering, cognitive and sociological approaches to context.

4.1 Engineering approaches

Context in the Operating Systems field refer to the context of processes [13]. Contexts are regarded as a *state* and are implemented with tables maintained by the operating system that have an entry for each process (fig. 3). This entry contains information about the process' state (running, blocked or waiting), its program counter, stack pointer, memory allocation, the status of its open files, its accounting and scheduling information and everything that must be saved when the process is switched back from running to ready or blocked state so it can be restarted later as if it had never been stopped.

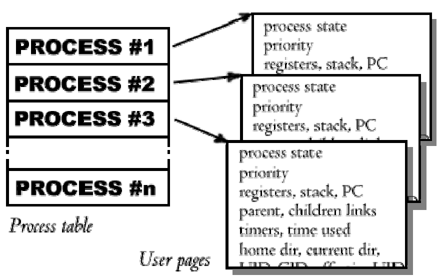

Figure 3. Process Context tables

The Artificial Intelligence field has developed an extensive research on context. In this field, context is viewed as a collection of things (sentences, propositions, assumptions, properties, procedures, rules, facts, concepts, constraints, sentences, etc) associated to some specific situation (environment, domain, task, agents, interactions, conversations, etc). This consensus is reflected in the "box metaphor" [20] (fig.4). The intuition is that context can then be seen as a container where its content depends on some set of situational parameters or dimensions. Dimensions such as time, location, culture, topic, granularity and modality have been proposed as defining elements of context space [22]. A proposal for a workflow context space in [23] includes function, behavior, causality, organization, information, operation and history parameters.

Context-aware applications have also modeled contexts as a function of localization, user identity, activity and time parameters [24]**.** However, more recently this field and the CSCW field are acknowledging the need of developing richer context models providing other information than time and location [14, 26]. The need to model user contexts and interaction contexts for improving user support is also acknowledged in [25].

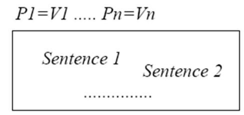

Figure 4. The box metaphor

4.2 A Cognitive Approach

B. Kokinov [17] developed a dynamic approach to context modeling to understand how human cognitive processes are influenced by context and how to model this influence in computer simulations. This work defines context as *the set of all entities that influence human (or system's) behavior on a particular occasion*. This context model assumes that mental representations involved in the current context are being formed by the interaction between at least three processes: *perception* that builds new representations of the current environment; *memory* that reactivates or builds representations of old experiences; and *reasoning* that constructs representations of generated goals, inferred facts, induced rules, etc. It is also assumed that context in turn influences perception, memory, and reasoning processes. The main principles of the dynamic theory of context are: (1) context is a state of the mind, (2) context has no clear-cut boundaries, (3) context consists of all associatively relevant elements and (4) context is dynamic.

4.3 Sociological Approaches

Sociological approaches typically regard context as networks of interacting entities (people, agents or actors and artifacts). These approaches focus on the structural properties of contexts, resulting from recurrent interactions among entities. Whereas some focus on the network elements, others focus on its emergent properties. In the latter case, the context itself is regarded as an entity which both supports and regulates interactions among its members [19]. Activity Theory (AT) [30] and Actor-Network Theory (ANT) [21] have been widely used in modeling social contexts. Both theories approach contexts as networks. Whereas ANT has been mostly used for a 'macro-modeling' of contexts, AT has been used in addressing finer-grained context models.

5. THE CONTEXT PRIMITIVES

Organizational activities have been modeled with a variety of concepts such as tasks, actions, interactions, roles, actors, goals, events, time and resources (e.g. tools, information, skills or people). We agree with [11] in the need of different primitives, capable of tying together other sets of common primitives used in modeling user behavior. In [7] we proposed *action* and *interaction contexts* as basic primitives to capture and model multitasking. However, these primitives need refinement.

In this section, we provide a description of the structure and state variables of action and interaction contexts. In terms of its structure, we follow sociological and cognitive approaches and regard contexts as networks of people and/or artifacts. In terms of its state, we use the operating systems notion and regard contexts as a set of state variables. These variables describe not only the state of individual network elements, but also of the state of the network emergent properties.

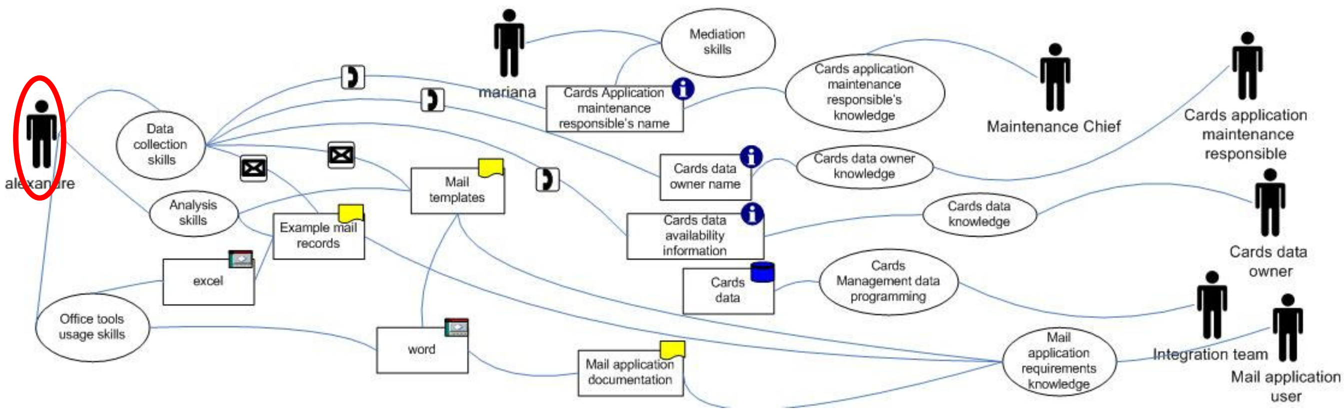

Figure 5. The "Cards and Mail Data Collection" Action Context of Alexandre

5.1 Actions and Interactions

To capture and model multitasking we must identify "broken" tasks. This entails looking inside tasks and working on the basis of smaller execution units. We define **actions** as atomic acts that change the state of some information or application resource (e.g. document is finished, sent, etc.). Unlike tasks, actions do not have clearly identified goals (e.g. update, print document). Tasks are composed of several actions. **Interactions** are actions involving two kinds of actors (persons); a **sender** and a **receiver** (e.g. request meeting, ask questions). An interaction is performed by one person and intends to change the state of another person.

5.2 Personal Action Contexts

Each individual at work uses some particular set of resources. These resources may be related to task (procedures, practices or routines), information, application or technological items. Resources may also include personal competencies, habits, preferences or rules. Cognitive limitations force individuals to focus on one sub-set at a time, forcing to a continuous suspension and reactivation of concurrent sub-sets.

(Personal) **action contexts** define the *sub-sets* of relevant resources (along with its state) and the relationships among them (and the state of this relationship) for an individual *during particular time intervals*. Each person handles several *action contexts* that assemble together task, domain, tool and personal-related knowledge. Action contexts become 'entities' that are created, activated, modified, suspended, resumed or terminated according to criteria that may encompass task or resource-related factors, as well as personal scheduling heuristics.

Figure 5 shows an example of an action context that belongs to an individual (Alexandre), which is a programmer from our case study. This action context shows the network of relevant resources for Alexandre, when collecting data required for an application to manage the correspondence with clients (Mail Application). As depicted in figure 5, this action context encompasses the following resources; *formal information items* (yellow folders and blue database symbol): *mail records* and *template* and *mail application documentation cards database*; informal information items (white i's in blue circle): name of responsible of cards application maintenance; *cards data owner name* and *cards data availability information;* application items (red and gray boxes): *MS outlook, Word* and *Excel* and *human resources* (names in figure 5), which provide two kinds of competencies (depicted with ellipsis): skills *–data collection, analysis, mediation and MS office usage skills-* and knowledge about cards data and mail applications.

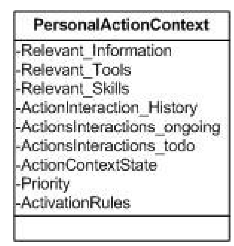

Figure 6. State Variables of Action Contexts

Personal action contexts are created and continually updated by actions and interactions. Hence, actions and interactions are also part of action contexts. Moreover, they reflect their past (action/interaction history), present (on-going actions/interactions) and future (actions/interactions to-do). Personal action contexts have also global variables that reflect emergent properties such as its general state, i.e. they may be active, suspended (due to lack of a resource) or interrupted by another action context. They also have a priority attributed by their owner and they may be triggered by some specific events i.e. they may have activation rules. Figure 6 summarizes action context state variables.

5.3 Inter-personal Interaction Contexts

Inter-personal relations and interactions are important influencing factors of multitasking behavior. Task interleaving not only depends on the kind of messages, but also on the nature of the interaction context shared by message sender(s) and receiver(s). Interaction contexts may refer to inter-personal, group, organizational or even societal levels. This work addresses interaction contexts at an inter-personal level. **Inter-personal relations** emerge from successive interactions between any two individuals and define the interaction rules between them [31]. Drawing from speech act theory [13], commitments are created, changed or cancelled through communicative acts or interactions. At the inter-personal level, commitments refer to answering to questions, accepting or rejecting proposals, performing requested actions, delivering information items, etc. Interactions and commitments are related to specific tasks or roles and consequently, to particular action contexts of the individuals involved. Whereas any two individuals share a single inter-personal relationship, they may share several interaction contexts.

We define (inter-personal) **interaction contexts** as (1) the interaction history (and their state) and (2) the *relevant commitments* (and their state) produced by interactions between specific *personal action context*s of two different individuals. Table 1 depicts an example of inter-personal interaction context state variables. These state variables include interactions between two or more personal action contexts, the commitment produced by the interaction and its state (pending, done, cancelled or re-scheduled). The state variables also include the original scheduled data and actual data of accomplishment.

5.4 Relation with other modeling primitives

Figure 7 shows the relationship between the proposed primitives and most common modeling primitives. Some of these primitives are based on the human multitasking models of section 2.1. Action contexts draw on action structures or schemas and personal scheduling rules draw on the contention scheduling component. Inter-personal relation and interaction context primitives draw on sociological approaches of context. Fig. 7 depicts first, the relation of action contexts with task, actor or role, or resource primitives. Second, it illustrates that interacting individuals share a set of interaction rules defined by their inter-personal relation. Third, the same two individuals may interact from different action contexts, creating several interaction contexts. These interaction contexts are supported and regulated by the inter-personal rules shared by the two individuals. Finally, figure 7 suggests a system architecture that integrated with proper clustering and classification techniques, enables more flexible association of actions and interactions with tasks, actors or resources through action and interaction contexts. This architecture is proposed and described in [33].

5.5 Acquiring the Context Models

Since action and interaction contexts are created and continuously modified by actions and interactions, acquiring these models require a combination of manual and automatic mechanisms to capture, analyze, group and classify actions and interactions. Whereas computer-mediated actions and interactions may be automatically captured, future actions (actions to-do) and actions in the physical world require manual mechanisms. Due to their personalized nature, action contexts are ultimately defined by their owners. Thus, although its acquisition can be aided with the use of automatic clustering and classification techniques, some degree of user intervention is always required.

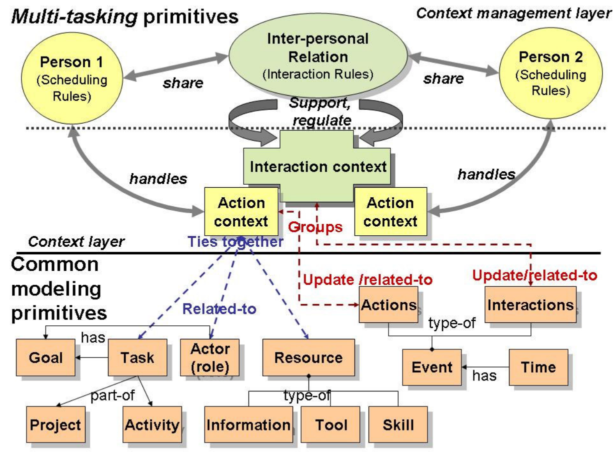

Figure 7. Multitasking vs. common modeling primitives

6. CASE STUDY

Some benefits of the proposed model are illustrated in this section through a case study in a real organizational environment.

6.1 Organizational setting

The case study involved a 3 week observation of a software development team of 4 programmers and the project leader. The team develops web applications for a commercial bank. Team members perform systems analysis, design, programming, test and maintenance activities. They also provide user support for the applications developed by them. During the observation period, the team performed tasks on the following applications; (1) *Suppliers,* (2) *Claims,* (3) *Clients' Correspondence (called Mail application),* (4) *Evictions* and (5) *Marketing Campaigns.* The team leader performed both system developing and *project management* tasks.

6.2 Method and techniques employed

The work started with a briefing session with the team leader where the research goals and potential methodologies for data collection were discussed. The team leader also made a short description of the team members and their roles (table 3). In this briefing it was decided to collect data through an observation technique based on ethnography [18], where the observer is also a participant of the observed setting. The observation was performed by team members, and coordinated by the team leader. Due to its exploratory nature, data capture and analysis, as well as model acquisition activities were performed manually.

Table 1. An (inter-personal) Interaction Context

Person1-Person 2	Date	Commited-by	Committed-to	type	Description	State	Original date	Actual date
1. Mariana **-**	06-12	~~Mariana (pac-10)~~	~~Catarina (pac-5)~~	~~**PROPOSE**~~	~~Solution to automatic table update problem~~	~~done~~	~~06-12~~	~~06-12~~
2. Catarina	06-12	~~Catarina (pac-5)~~	~~Mariana (pac-10)~~	~~**ANSWER PROPOSE**~~	~~Accept or reject solution~~	~~done~~	~~06-12~~	~~06-12~~
	15-12	Catarina (pac-5)	Mariana (pac-10)	**REQUEST**	Perform tests	pending	15-12	

Table 2. Sample of structured actions and interactions

Nº	Day	Actor or Sender	Receiver	Action or Interaction	Description	Tools	Information	Human competencies
8	6-01	Catarina		**SOLVE**	automatic table update problem	Sql Server, message management application	Sql Server and message management application documentation	programming & debugging skills
9	6-01	Catarina	Mariana	**PROPOSE**	solution to automatic table update problem			

Computer and non-computer mediated actions and interactions of the team members were registered in a chronological order. Each action or interaction was described with a separate sentence. Three weeks of actions and interactions were registered, encompassing 534 sentences. Registered sentences were first parsed using grammatical rules to separate the subject and predicate (verb and its complements). Synonym verbs were replaced by a single verb to avoid inconsistencies. Each action and interaction description was complemented with the set of application, information and human (competencies) resources involved. Parsed interactions were further structured using speech theory (for more details see [8]). Table 2 shows a sample of collected actions and interactions, once parsed and structured. We acquired action context models in a bottom-up fashion. First, actions and interactions were grouped according first, to their description and second, to the resources used. Second, a list of used resources was elaborated, separating them in three types (information, tools and human), according to the personal action context resource composition, illustrated in fig. 7.

Table 3. Subject actors and personal action contexts

Team member	*Role(s)*	*pac*
Mariana	Group leader, Programmer	13
Gonçalo	Programmer	2
Catarina	Programmer	4
Alexandre	Programmer	4
Carla	Programmer	2

6.3 Identifying Personal Action Contexts

As a result, 25 personal action contexts were identified and described. Table 3 depicts the number of personal action contexts associated to each team member during the three week period. While in some cases a personal action context encompassed several tasks, in others one task encompassed several personal action contexts. Thus, the relation between personal action contexts and tasks is ultimately defined by each individual. One example of the first case is the personal action context of Alexandre; the cards and mail data collection action context (fig. 5). This context is related to a single task; the Mail Application Programming task. In fact, this task is related to several action contexts. The second case is illustrated in Mariana's *management report elaboration action context* (pac 19a in table 4). This action context is related to the *project plan, annual budget* and *project status report elaboration* tasks.

6.4 Capturing Action Context Switches

Grouping actions and interactions in action contexts allows for context switch detection. Mariana, as the group leader, handled a greater number of contexts than the remaining team members. For this reason we selected her to analyze context switches and discovering scheduling rules.

Table 4 shows Mariana's action contexts. During the observation, she handled 13 different action contexts. Since data was registered in a chronological order, once actions and interactions were associated to a particular action context, it was easy to analyze if a following action (or interaction) belonged to the same or to a different action context. In that case, a switch was registered.

Table 4. Mariana's action contexts

pac	Personal Action Context Name
10	Message App. Automatic table update problem supervision
11	Marketing Campaings App. Adjustments
12	Cards Data Collection for Mail App.
13	Claims App. Document association function program
14	Claims App. File upload Component Modification
15	Claims application integration tests
16	Claims Application User Support
17	Software publication request
18	Message Maintenance
19a	Management Report Elaboration
19b	Management-related interactions
19c	Technical interactions with Microsoft consultant
20	Suppliers App. programming (web components).

Figure 8 depicts the context switches of Mariana (group leader) on the first day of observation. On this day, Mariana handled seven different action contexts (colored circles) and performed 23 context switches, (numbered lines of figure 8). Switching causes were also registered for each context switch. Due to space limitations, the whole list of causes identified is not included.

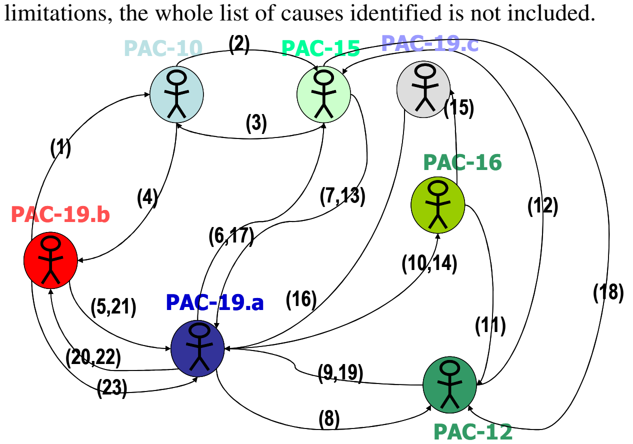

Figure 8. Mariana's context switches on first day

6.5 Discovering Scheduling Rules

Other action context state variables (fig. 6) are action context *activation (scheduling) rules* and *priority*. Registering switching causes enabled the discovery of these scheduling rules. Table 5 shows the scheduling rules discovered from Mariana's recurrent context switches and causes along the whole observation period. Action context priority was inferred from the number of times a particular action context interrupted active action contexts. These rules were rapidly and easily validated with the team members. This approach proved to be easier and faster than extracting rules exclusively from interviews, as tried in previous in-house experiences.

Underlying inter-personal rules: the validation of scheduling rules revealed underlying inter-personal factors. For example, the activation rule of context (pac 19.b - table 5) and its priority emerges from the fact that Mariana's chief expects rapid answers from Mariana. Also, the high priority given to user calls (pac 16) has to with Mariana's own policy of maintaining good relations with users. Thus, personal context scheduling rules embed inter-personal factors and can be made explicit analyzing these rules.

Table 5. Some Scheduling rules

pac	(Personal Action Context) Scheduling Rule	Priority
10	On user or Catarina's request	Normal
13	Auto-initiated, resume when free.	Normal
15	On team CG feedback reception	Normal
16	When Claims Application user calls	High
17	On team member's request	Normal
18	When Catarina is out	Normal
19.a	Before meetings, scheduled	High
19.b	On request of Dept. Head or team members	High
20	Auto-initiated, resume when free	Normal

6.6 Bottom-up task discovery

The observation of recurring action and interaction patterns also enabled the discovery of some tasks. These tasks were discovered during the process of grouping actions and interactions in their corresponding personal action contexts. Its correctness was validated by the team leader. Fig. 9 shows an example of the tasks discovered; the software publication procedure. Task specifications are grouped by task actors, to reflect the set of recurring patterns found in the personal action contexts of these actors.

7. Related work

Traditional task modeling approaches such as GOMS and CCT [15] are limited to modeling user behavior within tasks. Dix work on Trigger Analysis [32] defines several types of triggers and complements task models by including the triggers for each task. This modeling approach enables to capture not only how things are done, but by whom when and where they are done. However, trigger analysis is proposed as an enrichment of task models and thus, it is a task-based approach. Consequently, it does not provide straightforward means to discover generic personal scheduling heuristics (e.g. dispatching short task first).

A composed approach to model multitasking behavior (CMM) is proposed in [11]. CMM combines task, goal, event and time-based-models in an integrated model. CMM allows to define triggers for task groupings. Although more generic, these triggers are also directly related to tasks. Moreover, this work argues the need of modeling multitasking behavior through a different set of primitives, capable of tying together other sets of already defined primitives. Nevertheless, no new primitives are defined. None of these two modeling approaches address the bottom-up discovery of tasks from actions and interactions.

- **Team members:**
  - Upon completion of a software component, request Mariana its publication in quality environment
  - Upon publication in quality or production environments, test component
  - If tests in quality environment succeed, request Mariana its publication in production environment
  - If tests in production succeed, request user to perform tests
- **Team leader (Mariana):**
  - Upon component publication request from team member, request the component publication in quality to publication team
  - When informed by publication team, informs team member that the component is in quality or production
  - Inform systems developer about user test results
- **Publication team:**
  - When requested, perform software publication in quality or production
- **Users:**
  - When requested by systems developer, test software in production

Figure 9. Software Publication Procedure

8. CONCLUSIONS AND FUTURE WORK

This work proposes a context-based ontology to discover multitasking behavior. The proposed ontology defines a set of primitives for modeling multitasking and associates them with other modeling primitives used in user interface design. Some of its benefits are illustrated through a case study.

The identification and modeling of personal action contexts provides groupings of actions, interactions around related resources. This kind of grouping exhibits higher resemblance with models of cognitive structures than other grouping criteria such as tasks or projects. Moreover, regarding action contexts as entities allowed identifying context-related emergent properties such as its priority and activation rules. Action contexts may be related to several tasks. Conversely, a task may be related to several action contexts. This enabled the discovery of personal scheduling heuristics not related to specific tasks, as well as some underlying inter-personal rules. Through context action and interaction history it was also possible to find action and interaction patterns and to discover tasks in a bottom-up fashion. Although not addressed in this work, the defined interaction context model aims at facilitating the provision of proactive and timely reminders of the commitments associated to each individual, organized according to priority of the corresponding action context. It also aims at enabling the inference of other inter-personal factors influencing multitasking behavior.

The observation technique employed proved to be feasible, with participant collaboration. Nonetheless, analyzing the data through manual means has several limitations, particularly in finding and analyzing interaction contexts, due to the level of detail and volume of this data. We are presently aiming to capture actions and interactions and acquiring action and interaction context models through semi-automatic means. A prototype system for the observation and grouping of user actions and interactions is being developed. Clustering algorithms are being explored for this end. The prototype will include an interface for the display and manual capture and modification of action and interaction contexts. This prototype will also allow testing the suggested system architecture in supporting multitasking.

9. REFERENCES

[1] Parama C., Multitasking and the returns to experience, [online] gsbwww.uchicago.edu/labor/parama.pdf, April 2002

[2] Czerwinski M., Horvitz E., Wilhite S., A diary study of task switching and interruptions. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 175–182, 2004. ACM Press.

[3] Sohn M. et al., The role of prefrontal cortex and posterior parietal cortex in task switching. *In PNAS Proceedings of the National Academy of Sciences of the United States of America*, 97:13448–13453, 2000.

[4] Altmann E., Functional decay of memory for tasks. Psychological Research, 66:287297, 2002.

[5] Sohn M. and Anderson J., Task preparation and task repetition: Two component model of task switching. Journal of Experimental Psychology: General, 130(4):764–778, 2001.

[6] Red L., A multitasking fish story , [online] http://www.bizwatchonline.com/archive/august2001/index.asp, 2001.

[7] Zacarias M., Caetano. A., Pinto H.S., and Tribolet J., Modeling contexts for a business process oriented support. LNAI 3782, pages 431–442.Springer-Verlag, 2005.

[8] Zacarias M., Marques A.R., Pinto H.S., and Tribolet J. Enhancing collaboration services with business context models. *CEUR Proceedings of the Workshops on Cooperative Systems and Context*, volume 133, July 2005

[9] Porter A., Study: Multitasking is counterproductive. [online] http://cnn.com/career, Dec. 2001.

[10] Rubinstein J., Meyer D., and Evans J. Executive control of cognitive processes in task switching. *Journal of Experimental Psychology: Human Perception and Performance*, 27(4):763–797, 2001.

[11] Wild P. J., Johnson P. and Johnson H., Towards A Composite Modelling Approach for Multitasking, *Proceedings of the Task Models & Diagrams for UI Design, TAMODIA'04*, pp. 17-24, ACM Press, 2004

[12] Searle J., Austin on locutionary and illocutionary acts. The Philosophical Review, 77:405–42, 1978.

[13] Tannenbaum A. S., Modern Operating Systems. Prentice Hall, 2nd ed., 2001.

[14] Riss U., Ricksayzen A., and Maus H., Challenges for business process and task management. Proceedings of the 5th International Conference on Knowledge Management I-KNOW '05, June 2005.

[15] Dix A. et al., *Human–Computer Interaction*. Prentice–Hall, 1998.

[16] Whittaker S., Bellotti V., Jacek G., E-Mail In Personal Information Management, *Acm Communications*, 49(1):68-73, 2004

[17] Kokinov B., A dynamic approach to context modeling. In Proceedings of the IJCAI Workshop on Modeling Context in Knowledge Representation and Reasoning, pp., London, UK, 1995. Springer-Verlag.

[18] Beynon-Davis P., *Information Systems*. Palgrave, Great Britain, 2002.

[19] Reassembling the Social : An Introduction to Actor-Network, Oxford University Press, 2005

[20] Benerecetti M., Bouquet P., Ghidini C., On the dimensions of context dependence: partiality, approximation, and perspective, *CONTEXT'03 Conference,* Springer-Verlag, LNAI 2116: 59-72, 2001

[21] Giddens A., The Constitution of Society, University of California Press, 1984

[22] Lenat D., *The Dimensions of Context-Space*, CycCorp, http://casbah.org/resources/cyccontextspace.shtml

[23] Maus H., Workflow Context as a Means for Intelligent Information Support, *CONTEXT'03 Conference,*, Springer-Verlag LNAI 2116: 261–274, 2001.

[24] Dey A., Abowd G., Towards a Better Understanding of Context and Context-Awareness,. *GVU Technical Report*, ftp://ftp.cc.gatech.edu/pu b/gvu/tr/1999/99-22.pdf

[25] Brézillon P., Using context for supporting users efficiently. In 36th Annual Hawaii International Conference on System Sciences (HICSS'03) , Track 5, IEEE Computer Society, pp. 127- 2003.

[26] Brézillon P., Role of context in social networks. FLAIRS Conference, pp. 20–25, 2005.

[27] Smith, G., Baudisch, P., Robertson, G. G., Czerwinski, M., Meyers, B., Robbins, D., and Andrews, D., GroupBar: The TaskBar evolved. *Proceedings of OZCHI 2003*.

[28] Henderson, A., Card and S. K., Rooms: The use of virtual workspaces to reduce space contention in a windowbased graphical user interface. *ACM Transactions on Graphics*, 5 (3), 1986.

[29] Kaptelinin V., UMEA: Translating Interaction Histories into Project Contexts,., *Proceedings of the CHI'2003 Conference*, 5(1):353-360, 2003

[30] Engeström Y., Miettinen R. and Punamäki R.L. Perspectives on Activity Theory, Cambride University Press, 2005

[31] Magalhães R., Organizational Knowledge and Technology, Edward Elgar Publishing , 2004

[32] Dix A., Ramduny-Ellis D., and Wilkinson J., Handbook of Task Analysis for Human–Computer Interaction, Trigger Analysis – understanding broken tasks, pp. 381–400. Lawrence Erlbaum Associates, 2004.

[33] Zacarias M., Gomes R., Coimbra J. Pinto H.S., Tribolet José, Discovering Personal Action Contexts with SQL Server Analysis and Integration Services, *accepted for the IVNET '06 Conference*, Brazil, 2006