Discovering Personal Action Contexts with SQL Server Integration and Analysis Services

Marielba Zacarias[1,2,3], Rubem Gomes[2], José Coimbra[2],
H. Sofia Pinto[3,4], José Tribolet[2,3]

1 Center for Organizational Engineering, INESC, Lisboa, Portugal.
2 Universidade do Algarve, ADEEC-FCT, Faro, Portugal.
3 Department of Information Systems and Computer Science, IST/UTL,Lisbon, Portugal.
4 ALGOS, INESC-ID, Lisboa, Portugal
mzacaria@ualg.pt, sofia@algos.inesc-id.pt, jose.tribolet@inov.pt

Abstract. Evidence of multitasking at work can be observed through several case studies. Moreover, task interleaving is constantly increasing due to the improvement of information and communication technologies. However, despite the availability of several personal information and task management tools, an appropriate support for multitasking is still lacking. Some reported limitations of current tools are: (1) overhead posed on users, (2) lack of automatic means for associating user actions with task or project contexts and enabling automatic switching between these contexts and (3) exclusion of interpersonal elements. Another important limitation is the design of current tools, which organizes actions and interactions according to single pre-defined schemas (e.g. tasks, projects or resources). We argue that this inflexibility is caused by the lack of an underlying theoretical model of multitasking. To overcome these limitations, this paper proposes a systems architecture that combines models of human multitasking and personal contexts. With the proposed architecture we aim at enabling semi-automatic discovery, display, switching, analysis and management of personal action contexts. This architecture is currently being implemented in a prototype with Visual Studio .net 2005 and SQL Server. In this paper, we report results on the automatic discovery of personal action contexts with SQL Server Integration and Analysis Services. These results are compared with results obtained through manual means.

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 information and communication 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. Related 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 [14], 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]. Three main reported limitations of these tools are: (1) configuration and organization overhead posed on users, (2) lack of more intelligent and flexible means for associating user actions with tasks or project contexts and enabling an automatic switching between these contexts [27] and (3) exclusion of inter-personal elements [15]. The value of automatic user observation and the discovery of context identification heuristics in overcoming the first two limitations is acknowledged in [27]. From our point of view, the discovery of scheduling heuristics including personal and inter-personal factors will indeed enhance 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 this limitation is due to the lack of an underlying theoretical model of human multitasking.

To overcome these limitations, we propose a generic systems architecture based on models of human multitasking and personal action contexts. The need to model action contexts in addressing the problem of human multitasking at work was argued in [7]. 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*. An appropriate support to multitasking entails acknowledging (1) the several action contexts handled by a single individual and (2) how he or she handles these action contexts i.e. discovering personal scheduling heuristics.

The architecture proposed in this paper aims at enabling semi-automatic means for the discovery, display, switching, analysis and management of personal action contexts. The main expected contributions are: (1) to provide a better support to multitasking through context-based interfaces and (2) to enable the reuse of context information for a bottom-up discovery of tasks, as well as personal and inter-personal work-related features. This paper describes first, the proposed architecture. Second, it reports and compares results on manual and automatic discovery of action contexts, using SQL Server 2005 Integration and Analysis Services. The remaining of this paper is structured as follows; section 2 summarizes the theoretical framework of this work. Section 3 describes the proposed architecture. Section 4 describes the method and discusses the results of action context discovery. Section 5 summarizes related work on multitasking. Finally, section 6 gives our conclusions and future directions.

2. Theoretical framework

This section starts by summarizing the theoretical ground of the proposed architecture, which integrates a model of human multitasking model the notion of context. Then, it describes the clustering techniques used in implementing the architecture

2.1. The Attention-to-Action (ATA) Model of Human Multitasking

Figure 1 illustrates the ATA model [10]. This model has three subcomponents: *action schemas*, *contention scheduling*, and 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 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 occasions, 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**.

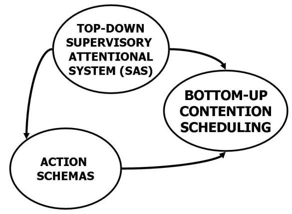

Fig. 1. The ATA model

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 to organize complex actions and perform novel tasks by selectively activating or inhibiting particular action schemas, superseding bottom-up influences of contention scheduling and accommodating a person's overall capacities and goals.

2.2. Context Modeling Approaches

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

Engineering approaches

Context in the Operating Systems field refer to the context of processes [12]. Contexts are regarded as a *state* and are implemented with tables maintained by the operating system that have an entry for each process. This entry contains information about the process' state (*running, blocked* or *ready*), 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* states so it can be restarted later as if it had never been stopped.

The Artificial Intelligence (AI) field has performed 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" [19]. The intuition is that context can then be seen as a container where its content depends on some set of situational parameters (also called dimensions). Dimensions such as time, location, culture, topic, granularity and modality among others, have been proposed as defining elements of context space [20]. A proposal for a workflow context space in [21] includes the following parameters: function, behavior, causality, organization, information, operation and history.

Context-aware applications have also modeled contexts as a function of localization, user identity, activity and time parameters [22]**.** However, more recently this field and the CSCW field are acknowledging the need of richer context models providing ohter information than time and location [13, 24]. The need to model user contexts and interaction contexts for improving user support is acknowledge in [23].

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 this influence could be modeled in computer simulations. This work introduces an operational definition of context; *context is the set of all entities that influence human (or system's) behavior on a particular occasion*. The proposed 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 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.

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 [31]. Activity Theory (AT) [28] and Actor-Network

Theory (ANT) [18] 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.

2.3. Clustering

Clustering techniques are applied to discover groups based on attribute similarity, within data sets [30]. The choice of clustering approaches is usually dictated by the available tools. There are three main clustering methods; (1) nearest neighbor rule known as k-means, (2) incremental and (3) statistical clustering. K-means is an iterative approach that assigns instance data to their closest cluster according to a distance function. K-means associates instances deterministically in disjoint clusters. Since the distance function is typically based on Euclidian metric, it applies mostly for numeric data. Incremental clustering performs hierarchical groupings of instances according to a cluster to a category utility function, which is a cluster "quality" measure. Statistical clustering assigns instances to cluster probabilistically. From a probabilistic perspective, the goal is to find the most likely cluster set of a given data.

Probabilistic clustering entails identifying the probability density functions of the data source. Each distribution governs the attribute values of a different cluster. An efficient representation of the probability density function is the *mixture model*, which asserts that the data is a combination of k individual component densities, corresponding to the k clusters [32]. When the class of an instance is known, the cluster distribution gives the probability of an instance having a certain attribute value set. Table 1 illustrates a probabilistic cluster membership. Since data records may belong to all k clusters but with different probability, the mixture model allows overlapping clusters.

Table 1. Probability cluster membership example

instance	Cluster 1	Cluster 2	Cluster 3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
…	…	…	…
h	0.3	0.5	0.2

The Expectation-Maximization algorithm (EM) addresses the problem of identifying a set of k populations in the data, and providing a model (density distribution) of each of the populations from given data records (observations). In other words, the EM algorithm approximates the data distribution by fitting k component density functions f_h, with $h=1,2,..,k$ to a database D with m records and d attributes. Let $x \in D$, the mixture model probability evaluated at x is:

$$p(x) = \sum_{h=1}^{k} Wh.fh(x \mid \phi h),$$

where Wh = fraction of D records belonging to cluster h and sum to one:

$$\sum_{h=1}^{k} Wh = 1 \geq 0.$$

The functions $fh(x|\varnothing_h)$, h=1,2,..k are the cluster component functions modeling the records of each cluster and $\varnothing_h$ are the specific parameters used to compute *fh.* The probability of membership of the data record x in cluster h is:

$$\frac{Wh.fh(x|\phi h)}{\sum_i wi.fi(x|\phi i)}\ .$$

By making the assumption that the database attributes are independent, the component density functions can be decomposed as a product of density functions over each attribute:

$$fh(x\mid\phi h)=\prod_{i=1}^{d} fh,d(xj\mid\phi h),\ j=1,..d$$

This algorithm iterates over an initial cluster model improving successively its fitness to data. This means that it begins with an initial estimation of ∅ and updates it, in each iteration. Updated ∅ is evaluated with an underlying clustering criterion, which provides an objective function to measure how well the probabilict model fits the data. This objective function is log-likelihood of the data.

$$L(\phi)=\sum_{x\in D}\log(\sum_{h=1}^{k} Wh.fh(x\mid\phi h))$$

The algorithm terminates at a local optimum or saddle point of this clustering criterion (stopping criteria: $|L(\varnothing)j - L(\varnothing)j+1| \leq \varepsilon$). Due to its probabilistic nature, arbitrary shaped clustering can be effectively represented by the choice of suitable component density functions (e.g. poisson, spherical or non-spherical Gaussians for numeric attributes and binomial or multinomial distribution for categorical or discrete data).

Clustering with Sql Server 2005®.

In Sql Server, a mining model is defined by a data mining structure object, a data mining model object, and a data mining algorithm [29]. The **mining structure** is a data structure that defines the data domain from which mining models are built. A single mining structure can contain multiple mining models that share the same domain. The building blocks of the mining structure are the mining structure columns, which describe the data that the data source contains. These columns contain information such as data type, content type, and how the data is distributed. A mining structure can also contain nested tables. A nested table represents a one-to-many relationship between the entity of a case and its related attributes. For example, if the information that describes the customer resides in one table, and the customer's purchases reside in another table, you can use nested tables to combine the information into a single case. The customer identifier is the entity, and the purchases are the related attributes.

A **data mining model** applies a mining model algorithm to the data that is represented by a mining structure. Like the mining structure, the mining model contains columns. A mining model is contained within the mining structure. The mining model contains two properties: Algorithm and Usage. The a*lgorithm* parameter defines the algorithm used to create the model. The usage parameter is a property that defines how a column is used by the model. You can define columns to be input columns, key columns, or predictable columns. A **clustering model** must contain a *key column* and *input columns*. You can also define input columns as being predictable. The Microsoft

Clustering algorithm offers two methods for calculating how well points fit within the clusters: Expectation Maximization (EM) and K-Means.

3. Proposed architecture

The proposed architecture is based on a model introduced in [7] and refined in [8]. Thus, this section starts describing (and further refines) the model elements addressed by this architecture. Then, the architecture is described.

3.1. Basic concepts

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). Modeling multitasking requires different primitives, capable of tying together other sets of commonly used primitives [11]. In [7] we have proposed two primitives; *action* and *interaction contexts to capture and model human multitasking*. However, these primitives need refinement. In this paper, we focus on action contexts and provide a description of its structure and state variables.

Personal Action Contexts

Each individual at work uses particular sets 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*. Figure 2 shows an example of an action context that belongs to an individual (Alexandre), which is a programmer from a case study. This action context shows the network of relevant resources for Alexandre, when collecting data required for an application to manage mail exchange with clients (Mail Application). As depicted in figure 2, this action context encompasses the following resources; *formal information items* (folders and database symbols): *mail records* and *template* and *mail application documentation cards database*; informal information items (i's in circle): name of responsible of cards application maintenance; *cards data owner name* and *cards data availability information;* application items (gray boxes): *MS outlook, Word* and *Excel* and *human resources* (names below people icons), which provide two kinds of competencies (ellipsis): skills *–data collection, analysis, mediation and MS office usage skills-* and knowledge about cards data and mail applications.

Individuals use, produce and change the state of resources (information, application or competencies) through *actions*. *Interactions* are actions performed by an individual to change the state of another individual(s). According to AT [28], interactions are always mediated by physical or psychological artifacts. Since actions and interactions create and continually update personal action contexts, they 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). 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 3 summarizes action context state variables.

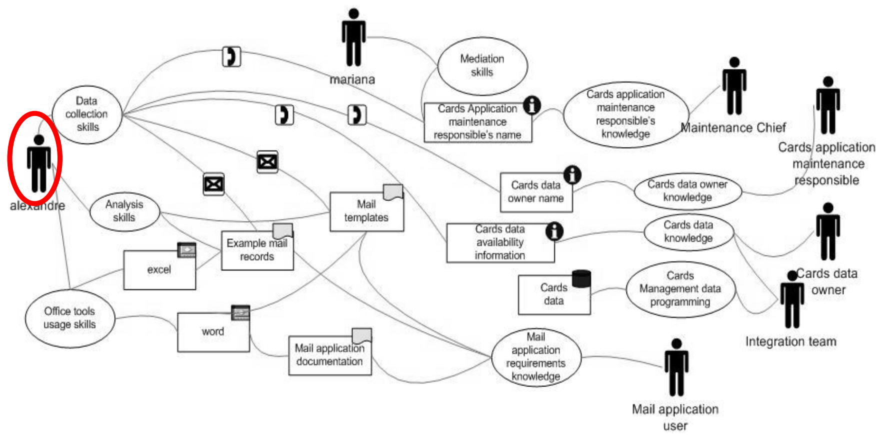

Fig. 2. An example personal action context (Alexandre): *data collection for mail application*

The action context definition given here draws elements from engineering, cognitive and sociological approaches (section 2.2). In terms of its structure, we follow sociological and cognitive approaches and regard it as networks of people and/or artifacts. In terms of its state, we the operating systems context notion and describe its state variables. These variables describe not only the state of individual network elements, but also the state of the network as a whole.

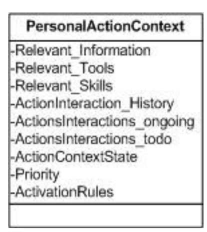

Fig. 3. Action Context State Variables

Personal Scheduling Heuristics

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 related to time, location and individual preferences or habits. Personal scheduling heuristics resolve potential conflicts when two or more action contexts with equal priority are activated at the same time. We expect to address this kind of rules in future work. We are presently focusing on action context properties. Modeling personal scheduling rules and action contexts as separate concepts, draws from the ATA model of human multitasking (section 2.1). Action contexts correspond to action structures and the contention scheduling component to personal scheduling rules that enable to decide between conflicting context activation rules.

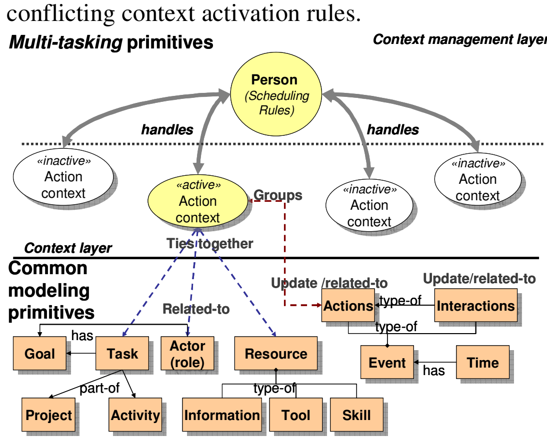

Fig. 4. Multitasking primitives vs common modeling primitives

3.2. Relation with other modeling primitives

Figure 4 shows the relationship between action contexts and other common modeling primitives. Each *person* handles *action contexts* using a specific set of *scheduling heuristics*. Action and interactions are events that change a person's state. Action contexts group together related *actions* and *interactions.* Action contexts also group together sets of resources (e.g. information, tools, skills) used by individuals performing some task or role, at particular intervals. Tasks and roles are typically related to specific goals and are part of broader activities or projects. Thus, action contexts tie together task, actor or role, resource and event primitives.

3.3. Acquiring Action Context Models

Since action 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. Computer-mediated actions and interactions may be captured through automatic means. Manual mechanisms are required to capture future actions i.e.actions to-do and actions in the physical world. Due to their personalized nature, action contexts are ultimately defined by their owners. Thus, although its acquisition can be aided through automatic grouping and classification mechanisms, some degree of user intervention is required.

3.4. A context-based system architecture for multitasking support

An appropriate support to multitasking requires systems that acknowledge the several action contexts handled by each user and how he handles these contexts. This entails: providing means of (1) capturing user actions and interactions and (2) associating them (along with their related resources) with their respective contexts. Moreover, discovering context activation rules and overall personal scheduling heuristics is essential in enhancing these association mechanisms. The model illustrated in figure 4 suggests a context-based systems architecture to address these issues. This architecture, depicted in figure 5, is composed of six components; capture (manual and automatic) of user actions and interactions, automatic context discovery, context explorer, context analysis and context integration. The benefits of this architecture will be tested through a prototype, written in visual studio .net and sql server 2005. Whereas the first 4 components are currently under development, the last two (context analysis and integration) will be developed at a later phase.

1. **Automatic capture.** A system based on this architecture must provide an automatic capture of computer-based user actions and interactions. Our prototype accomplishes this through a spyware. Due to the volume of information captured by this type of software, it is necessary to select the relevant actions to be stored by the system. The spyware component detects the resources related to each action (applications and documents). Interactions can be automatically captured through messaging and e-mail applications. Our prototype includes a messaging component where interaction types are made explicit by the user for the detection of some kind of inter-personal features. This is not addressed in the present paper.
2. **Manual capture.** A component for manual capture of actions and interactions is also being developed to include user actions in the physical world, actions to-do and face-to-face interactions, when necessary for analysis purposes.
3. **Context Discovery.** In order to enable an automatic classification of actions, interactions and related resources according to their respective contexts, automatic means to discover contexts must be provided. We accomplish this through Microsoft EM Clustering algorithm®.
4. **Context Display.** Our prototype includes a “context explorer” that enables a context-based display of resources i.e., resources are organized in personal action contexts discovered from captured actions and interactions. This interface allows user feedback by letting him modify action contexts as desired. The user can move resources, actions and interactions from one action context to another. It is also possible to reorganize action contexts i.e., create or delete contexts and join two action contexts together.
5. **Context Analysis.** This component is not intended for user support. Rather, it aims at supporting the organization through the reuse of registered context-related data. On one side, it will allow to obtain valuable information about task execution and task performers. On the other side, it will enable to discover other personal and inter-personal features for task management purposes.

6. **Context Integration.** This component is also intended for organization support and aims at using discovered personal, inter-personal and task-related features through the analysis component to acquire the finer-grained elements of organizational models (e.g. task, actor or resource models) and integrate them in selected modeling tools.

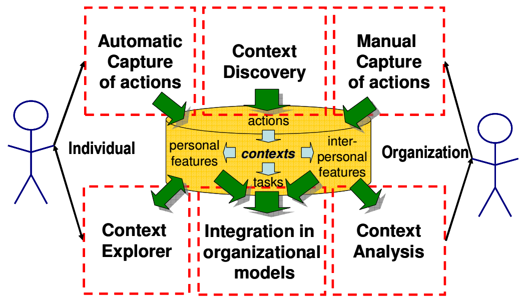

Fig. 5. The proposed architecture

4. Discovering Personal Action Contexts

Developing the context discovery and analysis components must be preceeded by research and experimentation to choose the appropriate algorithms to be embedded in our prototype. This section summarizes and discusses the results of these efforts.

Organizational setting and method employed

Data was collected from a case study conducted in a real organization. The case study involved a software development team of 4 programmers *(Gonçalo, Carla, Catarina, Alexandre*) and the project leader *(Mariana)*. The team develops web applications for a commercial bank. Team members perform systems analysis, design, programming, test and maintenance activities. During the observation, 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. Both research goals and data collection method were discussed in a briefing session. In this session it was decided to collect data through an observation technique based on ethnography [17]. Observation was thus performed by team members, and coordinated by the team leader. A set of actions and interactions was collected through a 3 week observation period. We discovered personal action contexts from these actions and interactions. This was accomplished both by manual and automatic means. The results of the manual discovery were used in validating automatic results.

4.1. Manual Process

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 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). 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 actions and interactions, once parsed and structured.

Table 2. Examples of structured Action and Interactions

#	*Day*	*Actor Send.*	*Rec.*	*Action. Interacc.*	*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			
10	6-01	Mariana	Catarina	**ACCEPT**	solution to automatic table update problem			

Identifying Personal Action Contexts: 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 figure 4. As a result, 20 personal action contexts were identified and described. While in some cases a personal action context encompassed several tasks, in others one task encompassed several personal action contexts. Thus, the relation between tasks and personal action and tasks must be 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. 2). This context is related to the Mail Application Programming task, which is related to several action contexts. The second case is illustrated in Mariana's *management report elaboration action context* (PAC 19a in table 3). This action context is related to the *project plan, annual budget* and *project status report elaboration* tasks.

Table 3. 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
20	Suppliers App. programming (web components).

Capturing Action Context Switches: Grouping actions and interactions in action contexts allows for context switch detection. Table 3 shows Mariana's action contexts. During the observation, she handled 12 different action contexts. Since data was registered in a chronological order, once actions and interactions were associated to a particular action context, it was analyzed if the next action (or interaction) belonged to the same or to a different action context. In that case, a switch was registered. Switching causes were also registered.

Table 4. Example action context properties (Mariana)

PAC	(Personal Action Context) Scheduling Rule	Priority
10	On user or Catarina's request	Normal
13	Auto-initiated, resume when free.	Normal
16	When Claims Application user calls	High
19.a	Before meetings, scheduled	High

Discovering Action Context emergent Properties: Registering context switches and their causes enabled to discover two emergent properties of action contexts; activation rules and priority (fig. 3). Table 4 shows some of Mariana's action context activation rules and their priority. Action context priority was inferred from the number of times a particular action context interrupted active action contexts.

4.2. Automatic Process

Automation of personal action context discovery has two phases; data preparation and clustering. The first phase was accomplished through the implementation of a Sql Server Integration Services® package. In the second phase, the clustering algorithm provided by Sql Server Analysis Services® was applied over different data sets.

Data preparation phase (blocks 1-8)

1. *Original input file format.* The starting point was an Excel® worksheet containing the set of registered actions and interactions, once structured (see table 2). Each worksheet row includes the following fields.

- **Number.** a sequence number which uniquely identifies the action or interaction and which reflects its chronological order;.
- **Day:** the day of occurrence (the whole observation took place during December, so the month was not registered);
- **Actor-Sender:** It is the actor name in the case of actions and sender name in the case of interactions. When actions were performed by several actors, their names were separated by commas.
- **Receiver:** Name of the receiver for interactions. Several receiver names are separated by commas. In the case of actions, this field is empty;
- **Action_Interaction:** interaction and interaction type described through a verb.
- **Description:** free-text description of the action or interaction
- **Tools:** applicational or technological items used in performing actions or interactions.

- **Documents_Information:** formal or informal information items used in performing actions or interactions
- **Human:** human resources used in performing actions or interactions. These resources are described in terms of competencies (skills or knowledge).

To facilitate the manual data entry process, rather than introducing specific contents, some worksheet cells were shaded with colors. Two different colors were used; gray and pink. These colors were used for the resource fields (tools, documents_information, human) and have the following meanings. Gray cells are used to avoid introducing the same resources of resumed actions. For example the mail application programming task is interrupted and resumed several times. This entails registering the related action each time it is resumed. However, the related resources of this action are registered only when the action is initiated. When resumed, the resources field was shaded in gray. Pink cells meant data not collected. This occurred when registering actions of external actors (other people interacting with the team)

2. Data integration process

The data integration process encompasses 8 blocks (blocks 1-8) that implement database creation and data import, cleansing and filling tasks. The first 3 blocks implement database creation tasks. **Block 1** creates the database to hold the worksheet contents. Since this database was only temporary, a simple scheme was used; composed of a table to store all actions and interactions. The table columns are identical to the worksheet. The database also includes some auxiliary tables used to fill the gray cells with their corresponding resources. **Block 2** creates the database tables. **Block 3** enables to delete the database tables when necessary (e.g. to modify the table columns).

Blocks 4-8 implement data import, cleasing and filling tasks. **Block 4** deletes all data of tables previously created. This enables to run the package several times without worrying about data produced by previous executions, which could lead to erroneous results. **Block 5** imports the worksheet contents (registered actions and interactions into the corresponding database table. During this process, leading and trailing spaces are deleted and all text is changed to lowercase. Number and day fields, originally typed as real numbers, are changed to integer. **Block 6** fills in gray cells with their corresponding resources. Filling gray cells was accomplished through an inner join of rows with gray cells (empty resource fields) and rows with non-empty resources sharing the same actor name, action type and description. Since the inner join leaves out rows with insufficient or erroneous data fields, these rows must be inserted in the result table. This last step is implemented in blocks 7-8. **Block 7** collects actions left with empty resource fields by block 6. **Block 8** inserts these actions in the result table.

3. Preparing the input for clustering.

Clustering input is prepared in blocks 9-14. Since personal action contexts group actions and interactions of a single person, the clustering algorithm needs to be applied one person at a time. Actions and interactions are grouped according to the resources used. This means grouping actions and interactions with similar terms in the resource fields. Three tables were created for this end; (1) the table “inputPessoa contains all actions and interactions of a single person; (2) the table “pessoaDicRec” contains a dictionary of all terms found in the resource fields and (3) "pessoaVectermos" contains the resouce term vector used as input for the clustering algorithm.

Block 9 creates the "inputPessoa", "pessoaDicRec" and "pessoaVecTerms" tables. **Block 10** is an optional component that enables the deletion of tables created by block 9 when necessary. **Block 11** deletes the "inputPessoa", "pessoaDicRec" and "pessoaVecTermos" contents. This is necessary to avoid the accumulation of data of different executions. **Block 12** copies the data of a selected person to the "inputPessoa" table. **Block 13** creates the resource dictionary used by the selected person ("pessoaDicTerm"). Creating this dictionary entails merging information, application and human resources and extract the terms used in these fields. This term extraction process is needed since the worksheet may store several resources separated by commas. The terms included words appearing twice or more times. In order to improve the clustering results, the description field was included in the term extraction. **Block 14** uses the term dictionary to create the term vector as input for the clustering algorithm. First, resource and description terms are sorted. Action or interaction numbers from which the terms were extracted are included. This enables to create the vector term table ("pessoaVecTerm"). In this table, each record is composed by the action or interaction number and one resource term. Thus, actions related to several resource terms appear several times (figure 6).

Num	Term
1	e-mail
1	team member address
3	catarina address
3	e-mail
4	cg team telephone number
4	telephone
6	claims application
6	claims application knowledge testing proce...
6	test data
10	on-line record update
10	record logic removal

Fig. 6. Vector term table example ("pessoaVecTerm")

Clustering Phase

Applying the clustering algorithm encompassed first, defining a data source and a data source view (see section 2.3). The data source is a connection to the database where the vector term table is stored. The data source view relates the vector term table with the actions table "inputPEssoa" (figure 7).

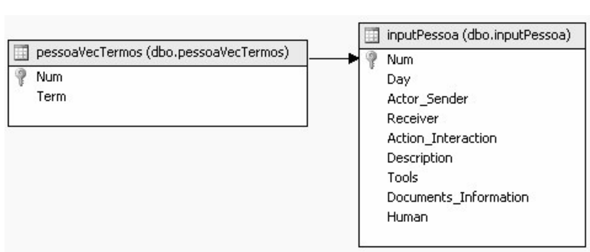

Fig. 7. Mining Structure

Second, the mining structure is created. This mining structure uses the "inputPessoa" as the case table and the vector term ("pessoaVecTermos") as nested table. Third, a mining model over the mining structure is created. This mining model includes the algorithm to be used. The mining model also requires defining the input and nested table keys ("num" and "term" in this case (Fig 8). EM algorithm with default parameters was used due to the categorical nature of input data. The only non-default value was the number of clusters. Default number of cluster is 1. We decided to use the same number of clusters identified by the manual process for each person (3 for Alexandre and 12 for Mariana), to facilitate the comparison of clustering results.

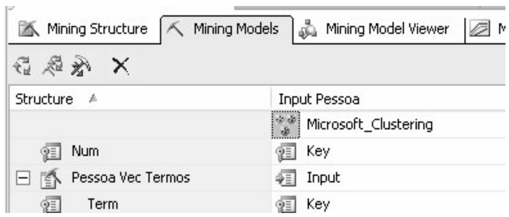

Fig. 8. Mining model

Clustering Results

The automatic clustering process was applied to two team members of the case study; (1) A regular team member (Alexandre) and (2) the team leader (Mariana). Fig 9 depicts the clustering results for Alexandre. Cluster Darkness reflects the number of cases associated to this cluster. The lines between clusters reflect some degree of cluster similarity.

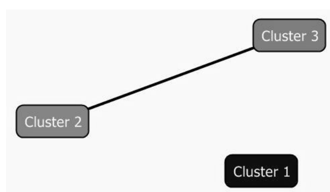

Fig. 9. Clustering Results for Alexandre

As illustrated in table 5, cluster 1 is mostly composed of programming (visual studio .net, sqlserver, msdn, etc.) and mail application resources. Cluster 1 has a strong resemblance with the action context pc1 (see underlined terms in table 6) found by the manual process. The terms related to this cluster had a 100% probability of belonging to it. This fact shows very well the differentiation of this cluster from the remaining ones. As show in figure 9, there is no line between this cluster and cluster 2 or 3.

Table 5. AutomaticTerm-Cluster association

Cluster	*Term*	*Prob.*
1	VS.net, .net programming skills, database, internet google, sqlserver documentation, mail application files, mail application database, sqlserver, mail application, mail application requirement, mail applications documentation, msdn	100%
2	e-mail, quality environment, mail application	18%
	cards data	17 %

2	Publication	15%
	updated evictions web service, telephone, web-service	14%
	production environment	11,%
	mariana address	10%
	Word, web service testing procedure, template publication procedure knowledge, web service test data, success, meeting, Client number	9%
	Evictions web service	6%
	Other terms	5%
3	evictions web service	56%
	evictions web service problem	26%
	problem	23%
	evictions web service problem knowledge	22%
	cards application maintenance	21%
	debugging skills	15%
	maintenance chief address, publication	14%
	production environment	12,%
	Other terms	8%

Clusters 2 and 3 have high resemblance to the corresponding action contexts (pac2, pac3) found through the manual process (see underlined terms in table 6). However, these two clusters share several terms. This produces an overlapping between these two clusters, not seen in the manual clusters. One reason for this was the lack of precision and consistency of input data, which is as a major problem in these techniques.

Table 6. Manual context-resource association

pac	Description	Resouces (Tools, Documents_Information, Human)
1	Mail application programming	VS .net, mail application requirements, .net programming skills, sqlserver, mail application files, mail application files and database, msdn, sqlserver documentation , net documentation, sqlserver usage skills, google.
2	Cards data collection	cards data, mail application files, e-mail, mail application requirements, mail application client, Word, management skills, , integration team, cards data owner , cards application maintenance responsible, cards data requirements knowledge, telephone, Excel, maintenance chief
3	Eviction Web Service Problem	eviction ws requirements, ws programming skills, eviction ws problem description, eviction ws files, eviction ws debugging skills, ws development in visual studio .net skills, visual studio .net documentation, publication procedure knowledge, e-mail

Figure 10 depicts the clustering results obtained for Mariana. Due to space limitations, the clustering results for Mariana will not be analyzed in full detail. We include it here to illustrate a case involving a greater number of contexts. The manual process produced 12 action contexts for Mariana (table 4). Though the algorithm was asked to group Mariana's data in 12 clusters, it only produced six. Analyzing these results, it was concluded that cluster 2 groups the clusters 13-16 identified in the manual process. These clusters are all subsets of the claims application and as such, they have commom terms. Grouping claims application actions and interactions in several clusters was due a Mariana's preference that could not be detected by the algorithm. The same situation was found in cluster 4, which groups the manual clusters 17 and 20.

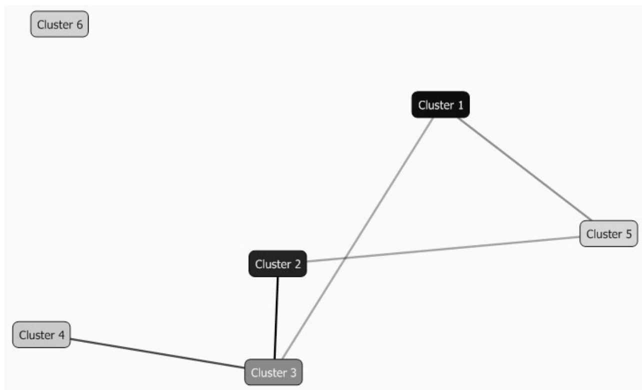

Fig. 10. Clustering Results for Mariana

5. Related Work

There are several research prototype tools intended to enhance user support regarding multitasking issues. GroupBar [25] 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 as units. Also, once the user lays out their work in a preferred configuration, GroupBar saves and restores these layouts. ROOMS is another project-oriented tool [26], 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 users [27]. The system UMEA aims at overcoming these limitations through a design which (a) organizes resources into project pools consisting of documents, folders, URLs and contacts, (b) monitors user activities and tracks resource usage in each project, (c) automatically organizes and updates resources to make them easily available to users when resuming each project. Communication-based environments tools (e-mail) organize resources around contacts, communication threads or messages. TaskMaster [33] is an e-mail and task management tool that organizes resources around a combination of tasks and communication threads (defined as "thrasks"). Thrasks are similar to personal action contexts. However, whereas thrasks relate threads to a single task, action contexts have a many-to-many relationship with tasks. None of these approaches reuse user information for organizational purposes.

6. Conclusion and Future Work

Several research works on tools supporting human multitasking acknowledge the need of discovering personal scheduling heuristics in order to overcome current limitations of personal task and information management tools. In this paper, we describe a generic systems architecture that aims at providing this kind of heuristics by enabling a semi-automatic capture, discovery, display, switching, analysis and management of personal action contexts. The proposed architecture also aims at reusing collected information for organizational support.

The present work addresses issues related to the discovery of personal action contexts. Context discovery was performed through manual and automatic means. Automatic discovery was implemented with a SSIS package and Microsoft EM Clustering Algorithm. Manual clusters were used in validating automatic clusters. The automatic process produced acceptable starting point clusters that can be refined according to user preferences. Moreover, cluster quality may be improved through automatic capture mechanisms which reduce data inconsistency and errors.

The benefits of this architecture will be illustrated through a Visual Studio.net® prototype, which is currently under development. Next steps include embedding the clustering algorithm in the context discovery component, to enable the grouping of actions and related resources in clusters. Resulting clusters will be used by the context explorer to display of actions and resources, according to the actions and resources estimated degree of membership to each context. Thus, actions and resources may appear in several contexts. This organization may be modified according to user actual preferences. Context-related information will be used in the context analysis component to discover, task, personal and inter-personal features. Once completed, the prototype will be tested on a real work environment. A complete evaluation of the benefits of this architecture entails inclusion it in a personal task or information management tool. This will be explored after testing our research prototype. At a later phase, the inclusion of these features on an enterprise modeling tool will be explored.

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, 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. Tannenbaum A. S., Modern Operating Systems. Prentice Hall, 2nd ed., 2001.
13. 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.
14. Dix A. et al., *Human–Computer Interaction*. Prentice–Hall, 1998.
15. Whittaker S., Bellotti V., Jacek G., E-Mail In Personal Information Management, *Acm Communications*, 49(1):68-73, 2004
16. 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.
17. Beynon-Davis P., *Information Systems*. Palgrave, Great Britain, 2002.
18. Reassembling the Social : An Introduction to Actor-Network, Oxford University Press, 2005
19. 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
20. Lenat D., *The Dimensions of Context-Space*, CycCorp, http://casbah.org/resources/cyccontextspace.shtml
21. Maus H., Workflow Context as a Means for Intelligent Information Support, *CONTEXT'03 Conference,*, Springer-Verlag LNAI 2116: 261–274, 2001.
22. 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
23. 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.
24. Brézillon P., Role of context in social networks. FLAIRS Conference, pp. 20–25, 2005.
25. Smith, G., Baudisch, P., Robertson, G. G., Czerwinski, M., Meyers, B., Robbins, D., and Andrews, D., GroupBar: The TaskBar evolved. *Proceedings of OZCHI 2003*.
26. 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.
27. Kaptelinin V., UMEA: Translating Interaction Histories into Project Contexts,., *Proceedings of the CHI'2003 Conference*, 5(1):353-360, 2003
28. Engeström Y., Miettinen R. and Punamäki R.L. *Perspectives on Activity Theory*, Cambride University Press, 2005
29. Sql Server 2005 Online Books [online] http://www.microsoft.com/technet/prodtechnol/sql/2005/downloads/books.mspx, 2005
30. Witten, I. H., and Frank E. Data Mining. Morgan Kaufmann Publishers, 2000.
31. Giddens A., The Constitution of Society, University of California Press, 1984
32. Bradley P.. Fayyad U. Reina C. Scaling EM (Expectation-Maximization) Clustering to Large Databases, *Technical Report MSR-TR-98-35*, Microsoft Research, 1999
33. Bellotti V., Ducheneaut N., Howard M., Smith I., and Grinter R., Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload, *Human-Computer Interaction*, 20: 89–138, 2005