Dynamic process adaptation: A context-aware approach

Vanessa Tavares Nunes, Cláudia Maria Lima Werner
COPPE/PESC – Systems Engineering and Computer Science Program – UFRJ -PO Box 68511, 21945-970
Rio de Janeiro, Brazil
(vanunes, werner)@cos.ufrj.br

Flávia Maria Santoro
NP2Tec and Postgraduate Information Systems Program
UNIRIO
Rio de Janeiro, Brazil
flavia.santoro@uniriotec.br

Abstract— Dynamism of day-to-day activities in organizations is inextricably linked and there is a variety of information, insight and reasoning being processed between people and systems, in carrying out a business process. We argue that flexibility in processes could be managed in real time, using context information collected in the work environment. This paper proposes a context management architecture approach that aims to improve and automate dynamic process adaptation. We explain how process adaptation may occur in real time and discuss a scenario for this proposal.

Keywords - Context management, business process design, dynamic adaptation

I. INTRODUCTION

“A major challenge faced by organizations in today’s environment is to transform ideas and concepts into products and services at an ever-increasing pace” [1]. The institutionalization of a process-oriented approach is already a reality and there is a consensus that processes in general (at different levels of abstraction) are essential for an organization’s performance [2].

According to Weske [3], business processes are a set of activities performed in a coordinated/collaborative manner in an (multi) organizational and technical environment towards a business goal. “Business processes are the key instrument to organizing these activities and to improving the understanding of their interrelationships” and the impact “among them” [3].

One of the challenges discussed both by academia and industry is related to the ability of an organization to respond to changes in an efficient and effective way. The concept of process flexibility is related to the need to understand situations that happen while people, systems and resources interact and demand adjustments.

Customizing a general process to make it applicable to a particular situation defines the concept of process adaptation. It requires experience and involves knowledge about various aspects of the business such as environment, people, used technologies and the organization itself, and also aspects outside of it.

These aspects influence each process instance in execution, supporting the idea that it is not possible to predict everything that could happen during its execution, since new situations can arise changing the scene. So, the design of a complete process model, predicting all possibilities that might occur, is giving place to a more flexible or "organic" design based on redesign, reuse and adaptation [1]. However, the identification of possible adaptations (i.e., adaptation on flow sequence, roles, information, technologies and resources) in a process instance, to address the situations occurring at a certain moment, is not a trivial task.

The concept of context [4] appears as a pointer to distinguish among the available information during the execution of a process, those that are relevant in order to provide inputs for the analysis of the people who is collaboratively participating and the process instance adequacy to the current situation, addressing organizations goals and strategy. Through context analysis, it is possible to assess the moment when the necessary adjustment needs to occur to ensure the progress of the process instance.

Rosemann et al. [5] define context-sensitive processes (context-aware processes), in which the process design should take into account certain contextual information that can influence the need for adjustments. The result of a process instance is indeed influenced by the context around it.

Nevertheless, in practice, there is a lot of information going on during process instances’ performance, and the analysis of the relationships among them becomes a challenge. Thus, the research problem investigated is that there are limitations regarding the support for decision-making in dynamic process adaptation.

In order to search for means to expand the perception of context in process instance performance, the focus of this research lies in a broaden process dynamic adaptation approach by performing a systematic management of the related context. The goal of this paper is to present an architecture that supports process instance adaptations, by dealing with adaptation needs when they occur and providing the necessary adjustments.

This paper is organized as follows: Section 2 presents an analysis of dynamic process adaptation approaches viewed through four aspects; Section 3 discusses the Context Management life cycle within dynamic process adaptation as a focus; Section 4 describes the proposed Context Management architecture whose use is exemplified in Section 5; Section 6 presents the conclusions, work in progress and future work.
II. DYNAMIC PROCESS ADAPTATION

The introduction of dynamics in process adaptation defines the customization of a process to make it applicable to a particular situation when it occurs, i.e., on the fly. In order to identify existing approaches in the literature to support dynamic process adaptation, we consider four specific interrelated aspects, presented as follows.

A. Notations

Approaches dealing with notations and techniques for carrying out dynamic process adaptation should support the standardization of modeling the common and variable parts of processes so one can identify what varies (and what are the adaptation rules among parts of the process) and enable the reuse of parts of the process.

In this scenario, a few proposals extend the process model notation to represent variability. Rosemann and Aaslt [6] propose an extension of EPC (Event-driven Process Chain) modeling notation [7], called C-EPC (Configurable Event-driven Process Chain), through the representation of variation points called "configurable nodes" to which variants or alternatives can be associated. Besides, restrictions (called "configurations requirements" and guidelines) may be applied in order to guide or restrict possible adaptations.

Other approaches suggest techniques to express variability in processes that can be applied to several notations. The BDD +04 approach [8] proposes the use of projection models, where it starts with a reference model that covers all possibilities and from which it is possible to create a projection for a specific scenario by eliminating the paths that are not relevant.

The PESOA project (Process Family Engineering in Service-Oriented Applications) [9] establishes the concept of variant-rich process model that represents an extended process model with the use of annotations and stereotypes so as to introduce variability in the model. Stereotypes are used to represent, for example, in the variation points, what can vary (<<VarPoint>>, what is optional (<<Optional>>) or abstract (<<abstract>>), and in each of these points, what are the possible variants (<<Variant>>). Other approaches, such as [10] and [11], also adopted the use of annotations to represent variability, but with less semantics than PESOA.

Another way to represent flexibility in processes may be through business rules because: (i) they can increase dynamicity in processes by providing a better support for flexibility in process adaptation at different levels of granularity and for various process elements (not only the flow activities) [12]; (ii) it is independent of a specific process modeling notation and, given a semantically rich representation language, the translation to process modeling notations (mainly for viewing) represents a small effort; (iii) the representation of business rules through formal languages supports process reasoning and the verification of correctness [12].

B. Process variability management techniques

This aspect is related to methods and infrastructure that enable process (re)design through the combination of process elements. Mechanisms are needed to support the management of these elements in a flexible way based on reuse and adaptation [1], while maintaining adaptation constraints and rules as well as process elements repositories.

Techniques such as [13][14][15] to systematize the design and reuse of software can also be applied to business processes. This scenario underlies existing theories of configurable process models [16] motivated by the use of the design by reuse paradigm.

The BPF (Business Process Family) and FPE (Family Process Engineering) [14] provide techniques based on the concept of product lines [17] and its implementation in the process domain. Process lines [13][15] implement the theory of software product lines [17] where the product becomes a process. The focus on process line techniques is to transfer the existing approaches in the product line engineering to the process field because we believe that the more adherent to the organization’s and client’s needs is a process, better is the outcome product. The PESOA project [9] was one of the first to implement this theory aiming at identifying service-oriented applications by means of the process activities that they support.

C. Technological infrastructure to support dynamic process adaptation

In dynamic environments it must be possible to quickly implement new business processes, to enable deviations and adaptations on-demand by dynamically adding, deleting or moving process parts (i.e., activities, roles, information etc) and to support dynamic process evolution (i.e., to propagate process schema changes to already running process instances).

A technological infrastructure to support (semi) automated dynamic process adaptation has been called PAIS (Process-Aware Information Systems), which stands for "a software system that manages and executes operational processes involving people, applications, and/or sources on the basis of process models" [1].

According to [1], the shift from task-driven to process-aware information systems presents some advantages: (i) the use of explicit process models through systems provides a means for communication between business and IT people who are responsible for designing, implementing and maintaining the technical infrastructure supporting these processes; (ii) process orientation in information systems allows for change without recoding parts of the system; (iii) improves the routing efficiency, automatically disseminating necessary information and activities to people and applications; and (iv) the explicit representation of process models enables management support at redesign, evaluation and monitoring of the execution flow.

Among the existing PAIS approaches stand out [18]: YAWL [19], FLOWer [1], Declare and ADEPT, each one supporting flexibility in a different way.

D. Alternative identification and selection support

In order to characterize situations that occur during the execution of a process, one has to deal with the complex nature of processes where information such as assumptions, values, experiences, interactions, decisions and reasoning support the
evaluation of process’s adherence to the organization's needs. Rosemann et al. [5] also discuss the challenge of identifying, documenting and analyzing scenarios that impact the design and execution of processes.

BDD+04 [8] proposes to use business features to determine a specific scenario by choosing appropriate branches of a reference model. These features are associated with process elements (i.e., activities, events, organizational resources and objects) that may vary. In the PESOA project [9], the concept of feature is also used, following software product line engineering definitions. The proposal is to associate process variations to features in a feature model. So when selecting the features (based on the establishment of restrictions among them), one can determine which variants can be selected and combined together in a given scenario. Thus, it is possible to select features that the process should have, supporting the selection of process elements that can be combined.

But according to La Rosa [20], the lack of support for selecting and configuring parts of the process is still an open issue. He proposes the identification of variants in a process by using specific questionnaires for a particular domain. Through the answers provided by the process manager, the Quaestio system [20] presents the most suitable variants that address the combination of answers.

Case-Based Reasoning (CBR) is a strategy well suited for managing new situations based on the solutions of similar previous ones, and has been proposed in the literature [21][22] for process change reuse and workflow adaptation support.

Resilience engineering, so as to manage exception handling or even unexpected situations, proposed by [23], aims to support process adaptation based on specific information diagnosis collected from the process users.

By analyzing the presented approaches, we noticed a variety of information, perspectives and reasoning steered among people and between people and systems, in which a series of circumstances are translated in work adaptation needs. We propose to use the concept of context to address the management of this information.

III. CONTEXT MANAGEMENT AS A SUPPORT FOR DYNAMIC PROCESS ADAPTATION

By analyzing the main approaches for dynamic process adaptation, we concluded that:

- Dynamic process adaptation involves not only the control flow, as most approaches focus on, but also process elements, such as business rules, roles, artifacts (inputs and outputs), systems and other resources [19]. The flexibility to adapt these elements remains an open topic;
- In some situations, it is not possible to know in advance how the process should behave in every situation, so there must be flexibility enough for process model evolution, on the fly;
- In process adaptation, the circumstances that are occurring must be considered [24], but also the understanding of how these conditions affect the process, and there is still no efficient technology to support this [13];
- There is no clear direction on how to identify the business aspects that influence process design and implementation. Snowdon et al. [25] and Aalst and Jablonsky [26] have highlighted that there may be a large number and variety of information that requires flexibility, involving questions such as: What may cause the need to adapt a process?; What needs to be adapted?; What are the possible adaptations?; and When is it possible or viable to accomplish them?;
- The analysis of situations that happen on a recurring basis may serve as input for process improvement, demanding changes in the process structure; the studied approaches are mostly not prepared for it. New patterns and rules may arise from process instance adaptations and an automated reasoning becomes necessary;
- There is the need for a systematic support for capture, storage and reuse of situations that arise while the execution of process instances is being performed, in order to identify adaptation needs addressing current organizations needs, and to create a body of knowledge about the organizations processes.

Also, when it comes to the context definition [4], it is possible to observe its consistency with reflections as outlined by [27][5][28], denoting the general understanding that context is always related to a focus (i.e., another context, a task, a person etc), and it is only considered relevant when it influences people’s, systems and environment behavior, and/or it is influenced by them. Context is also a complex concept and has unlimited dimensions (some may not even be described), consisting of a combination of relevant information, rules and propositions identified and captured in the environment.

Therefore, Fig. 1, based on [29][30][31], presents a model for the cycle of capture, storage, reasoning, and retrieval of context. This model was extended, and evolved since [32], to address issues related to the evolution of context and to focus its use, mainly, on the process dynamic adaptation. The model is discussed under five major aspects that encompass the entire context management life cycle integrated into an infrastructure to support dynamic process adaptation.

A. Identification and representation of contextual elements

Being context essentially a dynamic element that has high volatility, how to identify which context information is indeed relevant in different situations? Moreover, when is a context actually happening due to uncertainties to combine contextual elements? It is necessary to reach a common understanding related to the semantics of contextual elements, the relationship, impacts and dependency among them, and its implications to the organization.

In order to identify and use context, it is necessary to specify which contextual elements (CEs) will be addressed within the organization. The identification method may be amplified by
the analysis of intended “Goals/Strategies” and is under research. The CEs need also to be represented in an understandable and acceptable way to everyone involved. In Fig. 1, the contextual element is explicitly represented through the "Context Model". Concerning the representation of contextual elements, a representation language should be simple, flexible, extensible, expressive enough to be able to represent the real world, and generic enough to be able to adapt to changes in the domain.

**B. Capture of contextual elements**

The capture of contextual elements is directly related to the way they are presented in the environment and how the activity is performed. Manual, semi or fully automatic, “CE Capturing Mechanisms” may capture contextual elements when they arise or after the performance of a task.

In the case of manual mechanisms, the capture is purely human where the person who participates in the activity registers relevant information (requested by an application system). More information is available in the environment or is part of the result of an action, which requires mechanisms that automatically identify and capture them [31].

**C. Identification, representation and storage of context information**

Context identification occurs from the combination of values of contextual elements collected during the execution of process activities. The reasoning about captured contextual elements values is not obvious in situations where there are uncertainties. A mechanism that identifies context must be flexible enough to reason in these situations.

After its identification, the context must be associated to the action performed (or actions, grouped into a higher level of abstraction), to characterize their occurrence, and allow the identification of adaptation needs. The "Context-Action Association" stores this pair in a "Context repository".

**D. Monitoring and process adaptation management**

The “Monitoring and management Broker” represents the central component that is actually responsible for making the context management life cycle happens. It contains inference mechanisms that are capable of:

- Identifying changes in the context of the process instance, suggesting relevant adaptations based on existing contextual rules.
- Identifying evolution in the "Context Model" which requires analysis and evolution of the "Context Rules Repository" based on new concepts, relationships and constraints.
- Identifying changes in the organization "Goals / Strategies” that requires analysis and evolution of the "Context Rules Repository” so as to maintain it adherent to them.
- Evolving the "Context Rules Repository” based on reasoning over the executed processes instances.

**E. Presentation and implementation of process adaptation**

For each identified adaptation need, “Retrieval Mechanisms” can notify those responsible for the process instance, suggesting relevant adaptations for the situation at hand. However, automatic “Process Adaptation Mechanisms” may decide to adapt the process and implement it in the process instance via “Process Implementation Mechanisms”.

To computationally support this context management life cycle, we present a dynamic process adaptation architecture in the next section.

![Figure 1. Context management life cycle model for dynamic process adaptation](image-url)
IV. CONTEXT MANAGEMENT ARCHITECTURE: A PROPOSAL

To put this model into practice, an infrastructure that encompasses the following requirements is necessary:

- Provide a centralized contextual element model that can be shared by systems, equipments, agents and services, which are available in the work environment. More than that, it must provide a model that actually represents relevant information to the organization;
- Acquire data (contextual element) directly from the work environment coordinating the various and diverse mechanisms for capture and retrieval of contextual elements in heterogeneous environments;
- Perform reasoning in order to identify complex contexts (high-level context information) by combining data acquired in the environment. The challenge is to propose ways to represent real-world scenarios in terms of computationally interpretable rules [33]. Complex contexts usually cannot be directly captured from the environment. An example of a complex context is to identify a meeting event. The number of people, the location and the noise level can be interpreted and combined to identify this information;
- Detect and resolve inconsistencies between the collected contextual elements and/or the context interpreted, since in real scenarios ambiguities and uncertainties are common characteristics [33];
- Maintain the integrity of the reasoning rules so as to able to properly identify adaptation needs and take implementation decisions;
- Maintain context definitions and context rules constantly updated. The context is constantly changing and it is necessary to continually check for information validity based on process instances already executed in the same or similar context (similarity is also a challenge in the area);
- Detect a process adaptation need when a (or more then one) context change occurs and demands.
- Decide what adaptation and when it should be performed. Furthermore, it should be able to perform the adaptation automatically when requested. Decision making also represents a challenge due to uncertainty and partial observations of the environment [34].

The infrastructure presented in Fig. 2 was defined to address those requirements. It has been extended from [35] to focus the knowledge management main goal as a source of relevant information used to infer adaptation needs. A-CoBrA [35], based on [36], is a layered agent-based architecture designed for the construction of context-aware systems. It acts as a central context server for the process and the associated process management infrastructure that supports from context capture and identification to adaptation performance. This architecture is viewed and discussed through four main mechanisms called: Aggregator, Mediator, Maintainer and Actuator.

A. Aggregator

The aggregator is composed of autonomous intelligent agents responsible for the capture of contextual elements, a context identification mechanism that reasons over a combination of contextual elements in a given situation, and an Activity Context Association mechanism in order to put context on focus.

It is desirable that process performers are not overloaded with the task of registering contextual elements, but instead, keep the focus on their work. Therefore, the mechanisms for capturing contextual elements should minimize this work by reducing the overhead of manual registration. Time is also an important issue when it comes to context. Each contextual element must be associated to the time interval in which it occurred to make it possible to work with different information that happened in certain time intervals.

The notion that the context model is a dynamic artifact demands that the architecture is flexible enough to allow coupling of different mechanisms to capture contextual elements in different formats and medias. So the aggregator needs only to be concerned on how to use contextual elements and not on how to capture them.

The context identification module must be able to: (i) identify context from contextual elements captured from heterogeneous sources; (ii) identify context from partial capture of contextual elements (treatment of uncertainties); (iii) associate time to context information event. The notions of time and interval represent the basic concepts that support the mechanism to correlate the different context elements [36]; and (iv) manage access to context according to the privacy policies (not discussed in this work) determined by people and the organization.

Basically, the context identifiers are composed of a set of procedures (CE capturing mechanisms) that acquire contextual elements of sensors, agents, documents, and other information sources in the environment where the process is running and reason (Context Identification) about the combination of these elements, identifying the contexts while events occur.

The Context Identification can be seen as a middleware between the capture mechanisms and the Mediator that doesn’t have to be concerned on how to collect the different contextual elements and how to identify context, but on how to reason it in relation to the process.

B. Mediator and Maintainer

The Mediator acts in identifying adaptation needs when a context change occurs. The Maintainer is responsible for managing context definitions and context rules attained to current organization’s reality. Their key features are: intelligent behavior and decision-making support skills.

The Context Inference module is responsible for identifying when a context change occurs during process execution and also identifying inconsistencies in the inferred context information.
In this sense, the notion of time is also an important variable [36], so as to determine whether, for example, context information with opposite meanings is happening at the same time interval and therefore would cause an inconsistency, or if it is happening at different time intervals, only indicating context change. The Process Adaptation Inference module is responsible for identifying when a context change occurs, the needed and possible adaptations and when they should be performed.

The Context Definition Maintainer is responsible for continuously maintaining context definitions updated to the current organization’s situation. It involves identifying context model evolution by adding / changing / deleting contextual elements and relationships among them. The implementation of context logic reasoning mechanism must be held separate from the running process through information systems and/or automation. Thus, the process implementation becomes less rigid and easier to be maintained and adapted. The Context Rules Maintainer is responsible for continuously maintaining (and storing in the Context Rules Repository) context rules updated to current organization’s position. These rules consist in identifying the combinations of process elements that best matches organization’s goals, considering the current context.

The Context Repository manages the storage of the context model as well as instances of the contextual elements associated with the actions performed while the process is running. It also stores the context model, which contains the description of the internal structure that states how this information is stored.

**C. Actuator**

The actuator is composed of mechanisms that present the context inferred results stating the need for adaptation and suggesting and/or making automatic adjustments to the process in the work environment.

The Process Adaptation module is responsible for processes automatic adaptation based on the decision results sent by the mediator (Process Adaptation Inference Module). It involves performing the necessary adjustment to change the process instance. The Context Retrieval Module aims to present change needs and adaptation possibilities to the process manager if it is his/her responsibility to make the decision and manually perform the necessary adaptations.

The Implementation Mechanisms are responsible for implementing the necessary changes in the environment components, in which the process is running, through automatically triggering actions like buying stuff, adapting systems, technologies, infrastructure, and indicating specific training etc. It may also treat process integrity in all aspects.

Next section presents an application scenario to discuss the architecture in action.

**V. APPLICATION SCENARIO DISCUSSION**

The following scenario (Fig. 3) illustrates how the architecture will behave within a process that represents the purchase of songs on iTunes\(^1\). It is known that Apple's strategy is to establish itself as a leading media supplier in the upcoming years around the world. Buying songs is performed today by thousands of people, which means that the same process described below is instantiated hundreds (if not thousands) of times a day.

An instance of the process begins when John accesses the iTunes Web site. The process instance and the Context Management System (CMS) performance are detailed below. This scenario shows how a particular situation, i.e., the fact that John was considered a low-risk client, can be analyzed and reviewed in order to achieve the organization’s goals, while

\(^1\) www.apple.com/itunes
focusing on the strategy, the adherence to internal policies and maintaining the security level established by the manager.

![Figure 3. Work process model Scenario](image)

### TABLE I. WORK PROCESS SCENARIO EXECUTION

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access iTunes:</strong> John accesses iTunes to browse for songs</td>
<td></td>
</tr>
<tr>
<td>At this time, software agents automatically capture the IP of the machine that is accessing the store.</td>
<td></td>
</tr>
<tr>
<td><strong>Login the store:</strong> John logs in iTunes</td>
<td></td>
</tr>
<tr>
<td>Software agents automatically capture:</td>
<td></td>
</tr>
<tr>
<td>• John’s personal data (name, SSN, birth date and address);</td>
<td></td>
</tr>
<tr>
<td>• Bank details (credit card, company and credit situation), data from previous purchases at the Apple’s store (equipment, brand, year, configuration, payment method and history, credit card company acceptance etc);</td>
<td></td>
</tr>
<tr>
<td>• Registration status on the PayPal website (purchases, payment method and history).</td>
<td></td>
</tr>
<tr>
<td><strong>Select desired songs:</strong> John selects the desired songs,</td>
<td></td>
</tr>
<tr>
<td><strong>Request purchase:</strong> John then clicks to purchase the selected songs</td>
<td></td>
</tr>
<tr>
<td><strong>Check CC’s address:</strong> iTunes checks the credit card data and checks its address,</td>
<td></td>
</tr>
<tr>
<td>As PayPal is an Apple partner (working as another payment option), agents automatically capture historical data from previous purchases made by John.</td>
<td></td>
</tr>
<tr>
<td><strong>Context Identification</strong> - CMS infers that John is a regular customer, he has bought equipment at Apple’s store, no problems with payment or credit card authorization where registered, and he has a valid and international CC. Also, CMS infers, through its relationship with PayPal, that John is an old client (more than 2 years of account) who has done other purchases on other sites, all ending up regularly.</td>
<td></td>
</tr>
<tr>
<td><strong>Context-Action Association</strong> - CMS (Context-Action Association) associates the identified context to the process current running activity,</td>
<td></td>
</tr>
<tr>
<td><strong>Process Adaptation</strong> - CMS (Process Adaptation Inference) identifies an adaptation need, inferring that John is considered a safe customer, based on the organization’s goals, and therefore he can purchase the requested songs.</td>
<td></td>
</tr>
<tr>
<td><strong>Process Adaptation</strong> - CMS (Process Adaptation) changes the process, allowing John to purchase the songs.</td>
<td></td>
</tr>
<tr>
<td><strong>Process Adaptation</strong> - CMS (Implementation Mechanisms) communicates with iTunes and changes the system to allow John to perform the purchase.</td>
<td></td>
</tr>
<tr>
<td><strong>Transact CC authorization:</strong> iTunes transacts John’s CC authorization, which is approved.</td>
<td></td>
</tr>
<tr>
<td><strong>Authorize purchase:</strong> iTunes authorizes the purchase and allows John to download the songs.</td>
<td></td>
</tr>
</tbody>
</table>

Software agents were able to capture data from previous purchases made at Apple’s store and others via PayPal. The context identified based on the reason upon such data allowed the “Context Inference” to identify a situation where it was worth to consider an analysis based on the fact that Apple’s goals is to raise sales. The process was adapted by the “Process Adaptation” and implemented by the “iTunes instance implementation mechanism” through the introduction of a new output event after the “Check CC’s address” activity, specifying that “Customer John is allowed to make the purchase”.

Here, specifically, the contextual elements collected are of relatively easy access, and privacy policies (it was not the focus of this work but it will be addressed in the architecture implementation) and the exchange of information between organizations were not treated, but we recognize that there may be a greater complexity in dealing with cases where manual labor is a strong feature and where the exchange of information among people is informal. Considering subjective aspects, like personal feelings, this capture and identification is also a challenge, as well as the reasoning and adaptation needs.

### VI. CONCLUSIONS

This research focuses on a computational support for dynamic process adaptation addressing the management of context in a dynamic perspective, to serve as a basis to: maintain the suitability of a process instance through all its execution life to organizations and people’s needs/goals; discover new situations and demand new process behaviors not previously defined; understand how context affects process and identify process improvement opportunities.

The first results achieved were the establishment of the “Context management lifecycle model for dynamic process adaptation”, which allowed the focus on the architectural perspective and also the shaping of two workaround master thesis. One is dealing with the measurement of relevance among contexts and the other with the reasoning over low-level context information to identify high-level context information.

Currently, this work is in the architecture’s design phase. Also, reasoning techniques are being investigated in order to start the implementation phase by the Context Inference Module responsible for identifying when a change in context occurs and also inconsistencies in the inferred contexts. In order to identify context change Case-based reasoning,
workflow exception and workflow change approaches, among others, are been considered so as to understand what type of information is considered in order to reason over a process adaptation and how this information is manipulated.

ACKNOWLEDGMENT

Flávia Maria Santoro is partially supported by CNPq (Brazil) under the grant 305404/2008-3.

REFERENCES