CONTEXT-BASED PROCESS LINE

Vanessa Nunes, Claudia Werner
COPPE/UFRJ – Systems Engineering and Computer Science Program
Rio de Janeiro, RJ - P.O. Box 68511 - Brazil

Flavia Maria Santoro
NP2Tec and Postgraduate Information Systems Program – UNIRIO
Av. Pasteur 458, Urca – Rio de Janeiro, RJ – Brazil

Keywords: Business Process Line, Context, Process Dynamic Adaptation.

Abstract: Complexity and dynamism of day-to-day activities in organizations are inextricably linked, one impacting the other, increasing the challenges for constant adaptation of the way to organize work to address emerging demands. In this scenario, there are a variety of information, insight and reasoning being processed between people and systems, during process execution. We argue that process variations could be decided in real time, using context information collected. This paper presents a proposal for a business process line cycle, with a set of activities encapsulated in the form of components as central artefact. We explain how composition and adaptation of work may occur in real time and discuss a scenario for this proposal.

1 INTRODUCTION

Organizations search for continuous development in order to adapt to the highly competitive market. Complexity and dynamism of day-to-day activities are inextricably linked, increasing the challenges for constant adaptation of the way to organize work to address emerging demands. When it comes to information systems, we observe that market is demanding systems to be quickly developed and able to evolve and adapt to new situations in everyday life (Grisogono, 2006).

A process-aware information system (PAIS) can be defined as "a software system that manages and executes operational processes involving people, applications, and information sources on the basis of process models" (Dumas et al., 2005). According to La Rosa (2009), PAIS are becoming pervasive in the business environment and system focus has shifted from the data-driven approaches to a more holistic notion of a business process. Several authors note that the systematic reuse and adaptation can also be applied to business processes (Montero et al., 2007). Within this scenario, La Rosa (2009) states that the decision associated with a variation point in a process is placed at design time; thus not based on data values available when the process is performed. Moreover, there is a variety of information and reasoning handled among people and systems, during process execution that characterizes specific situations such as work requirements changes. We called these events "context". The concept of "context" emerges as an approach to support the discussion of business issues and thereby optimize the adaptation of work processes in real time, using context information collected.

We propose to implement this view within a process line approach and this paper aims to describe a business process line cycle which has a set of activities encapsulated in the form of components. We argue thereafter how adaptation of work processes and the derivation of PAIS may occur in real time.

2 PAIS WITHIN A PROCESS LINE APPROACH

Process Line is a set of process components, organized to represent common and variants parts (set of activities) within a specific domain, that can be reused and combined, according to rules to compose and adapt processes dynamically

Nunes V., Werner C. and Santoro F. (2010).
CONTEXT-BASED PROCESS LINE.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 277-282

(Rombach, 2006; Washizaki, 2006). A process component is an encapsulation of information and behaviours under a certain granularity level (Gary and Lindquist, 1999), (Fusaro et al., 1998). Process Line Engineering involves the use of models, procedures, architecture and technology that build, operate and manage a process line, such as software product line (SPL) (Clements and Northrop, 2001).

Process Family Engineering (PFE) is an approach (Bayer et al., 1995) that aims to enable the production of processes, where each product is a set of processes enabled at a certain time. It produces a system that evolves at runtime, where the features are processes, and contains all the family features.

Montero et al. (2007) present the Business Family Engineering (BFE) to explore the feasibility of adapting SPL techniques to Business-Driven Development (BDD), oriented to reuse processes, across different businesses. BFE can be defined as: a set of software systems driven by business processes where each product of the family has a set of common processes and a set of variable processes. They concluded that PFE is useful for managing single businesses, however not for a set of businesses (BFE). Rombach (2006) presents SPPL (software product and process line) engineering where artifacts and processes (organized in similarities and differences) can be chosen based on a set of product and process requirements and constraints. Washizaki (2006) proposes a process-tailoring technique which intends to solve problems with component-based approaches by building a Process- Line Architecture (PLA) in order to derive project-specific processes, combining and reusing core processes and variants for a particular domain.

We propose to use context information as a way of supporting recognition and composition of processes, improving a process line use and management. The concept of context used is based on Dey et al. (2001) which represents any information that can be used to characterize the situation of an entity. Its importance is due to the ability to provide greater meaning to work, facts, artifacts, and decisions taken in business process. The more the perception of context is improved, the greater is the support for activities implementation in a work process, as well as the specification and development of systems that support the work.

Dynamic adaptation of processes is one of the goals for the use of process lines due to the fact that when organization´s processes are well known, it is possible to make changes at runtime in order to adapt to new situations. So, we define the concept of process line based on context as process components, organized to represent common and variants parts (set of activities) within a specific domain that can be reused and combined, according to rules and based on changes in the context of people, systems and environments, to compose and adapt processes dynamically. Next section describes the details of the proposal.

3 A CONTEXT-BASED PROCESS LINE

The approach for the context-based process line engineering, depicted in Figure 1, was structured similarly to the SPL phases (Clements and Northrop, 2001; Kang et al., 1990) and previous proposals by Barreto (2007), Rombach (2006), and Schnieders (2006). As Rombach (2006) stated, we would, by means of a domain engineering process, create a generic (set of) process(es) that capture the commonalities and variabilities across a domain.

Domain Engineering comprises Analysis, Design and Implementation of the domain. It is achieved through two possible scenarios: first when there is a need to establish a process line, and second when requirements, needs and goals change, promoting the evolution of the process line. In both scenarios, these information, as well as process models and/or reference models, serve as input.

The three possible inputs for domain analysis originated from two approaches (Washizaki, 2006): bottom-up and top-down. In the first, organization process models are used to create the process line. In the other, it starts from scratch, based on reference models,to form new processes. We adopted the so called meet-in-the-middle, combining both two (Jaufman and Münch, 2005).

3.1 Context-based Process Line Phases

Domain Analysis aims to generate, the semantic relationships between the concepts used in the business processes, and thus make possible the use of them in organization. From the understanding of the concepts it is possible to identify context information considered relevant to describe the actions that occur, while performing work activities. The element "context model" provides an explicit representation of contextual knowledge that is being captured. According to Nunes et al. (2009), determining the information necessary for the composition of the context is not a trivial activity because it depends on the situation. External factors such as artifacts produced, discussions, messages exchanged and actions taken are relatively simple to be identified and caught.

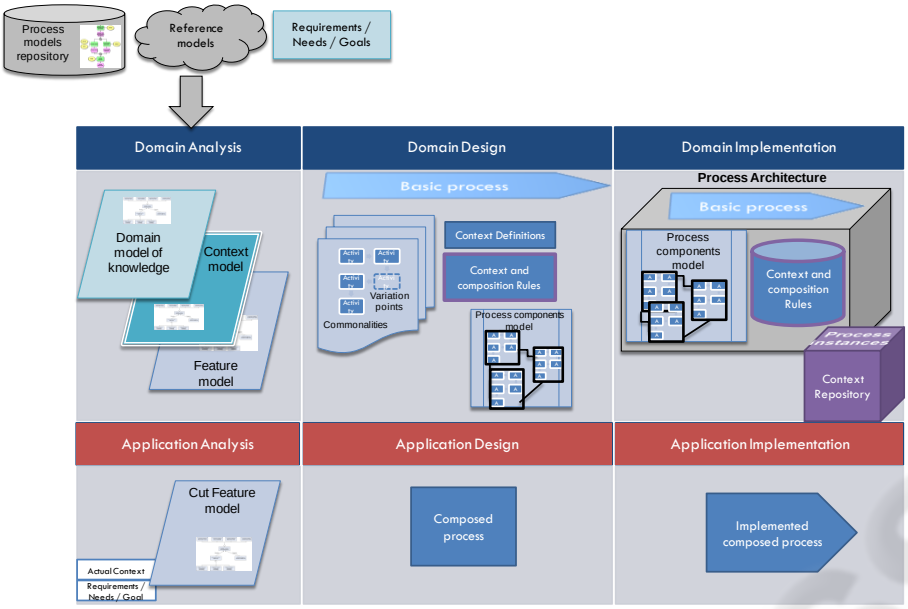

Figure 1: Context-based process line engineering approach.

However, individual aspects (interests, goals, experiences, etc.) are difficult to carry out. Also, to increase the awareness about the context and make it also computationally interpretable, it must be represented in a formal language. Finally, based on concepts and context understanding, and organization's requirements, needs and goals it is possible to define the features that represent domain characteristics of a process.

Domain Design is the most important phase of process line, since activities and processes are identified, i.e. what is core and what may vary (target for changes). The basic process (template) is defined, both from organization process models, and from reference models. For each process it is necessary to identify the commonalities and variation points and identify where a component begins and ends. The outcome of this phase is components containing activities invariants and variants, internal properties and interfaces with other components, and composition rules.

Domain Implementation involves building process architecture and context repository from the artifacts generated in the design. The architecture provides the elements, patterns, and framework for any level of detail (Barreto, 2007). The architecture is implemented to be used in the composition and adaptation of processes, according to the composition and context rules established.

In Application Engineering phase, the process is effectively implemented and is ready to run and adapted at run time. First, in Application Analysis, from the current context, and the specific current requirements that must be met, the features are selected for the new process to be composed. The outcome is the feature model, containing only the selected features.

In Application Design, based on the selected features and the existing similar projects contexts, process components are selected to be combined within the architecture The composed process consists in a new component that can be also reused in compositions and adaptations of processes. Thus, it is incorporated into the process architecture.

Finally, in Application Implementation the process is implemented, for example as a process-aware information system, so as to be systematically managed by context mechanisms.

This proposal supports the analysis of user's context responsible for the process and the team that understands the needs of the projects process. But, to increase the context perception it is necessary to specify the contextual knowledge, represent it in a standard model and make it accessible to all involved (Nunes et al., 2009). A context management model integrated within the process line is described as follows.

3.2 Context Management in Process Line

Figure 2, based on Nunes et al. (2009), presents a model for the cycle of capturing, storing, reasoning, and retrieving of context.

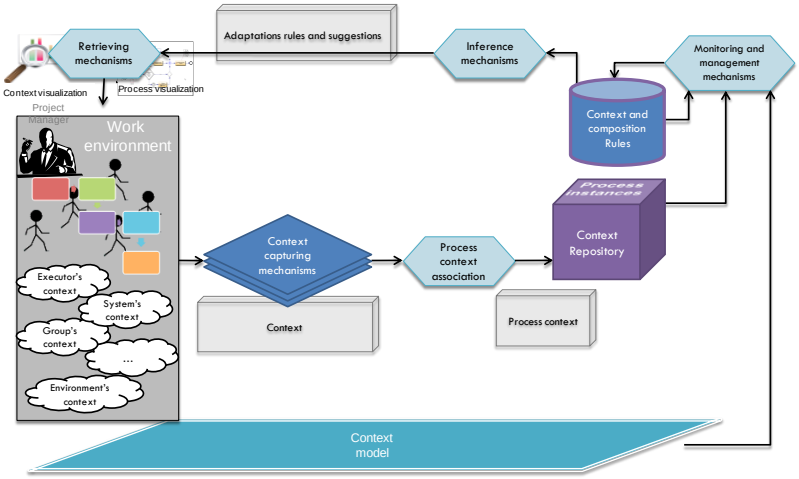

Figure 2: Context management life cycle.

This model was extended to address issues related to the evolution of context rules from the process line and incorporate context artifacts considered in the process line engineering cycle proposed. The model for context management life cycle is composed of four aspects:

- Representation of Context: A formal representation of context allows for building reasoning mechanisms to check context information consistency, they make possible comparisons with other contexts and inferences about new information from complex contexts involved. Currently, a context ontology is used to represent context (Nunes et al., 2009), nevertheless another meta-models are being studied.
- Capture and Storage of Context: The first step to compose and evolve the context repository is capturing the context that characterizes the actions being carried out in a work process in real time within a working environment. Capturing can be purely human (a professional logs its context). But context is available in the environment or in activity outcomes, which requires a mechanism that automatically identifies and captures them. To retrieve context information, they must signal a time unit, and mark the event (tasks, activities, process, etc.) when they were captured in accordance with what was defined in the establishment of the architecture of the process line.
- Retrieving and Presentation of Context and Suggestions for Adaptation: During work process, execution, inference mechanisms continuously monitor to identify changes in situations that require adaptation of the process. Based on context rules, these mechanisms suggest changes in the processes that may or may not be accepted by those responsible (ex. Project manager in Figure 2). Similarly, it can be performed directly by people themselves. Retrieving mechanisms are used to submit suggestions and to visualize previous contexts and current context rules.
- Monitoring and Management of Context and Context Rules: It is responsible for the context model evolution and the context itself. Context information may be necessary to characterize new aspects not planned before demanding a context model revision. Changes can enhance the generation of new context rules and/or the exclusion of others which, may have no relevance anymore. The direct interaction between individuals and interface for knowledge retrieval can create new context rules, allowing the perception mechanisms to make a selection increasingly optimized.

4 APPLICATION SCENARIO DISCUSSION

We discuss the proposal application within a scenario the project management domain. In order to establish a project management process line, people may identify processes already been performed and stored in a repository, reference models previously used or the ones that the organization intends to use, and requirements, needs and goals that must be attended. So, the basic process of performing a project planning is depicted in Figure 3.

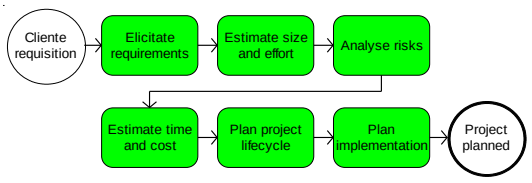

Figure 3: Work process model.

In Domain Analysis, the domain model can be elaborated, by using conceptual maps like those proposed by Abels et al. (2006) used in this work: Project, Task, Phase, Resource, Machine, Event, Risk, Objective and Result. In order to construct the context model we used Nunes et al. (2009) context ontology model including the following context information: problem size, problem complexity, client relationship, business relevance, objective, requirements stability, team experience, group working experience and team proximity.

In order to perform project management activities, methodologies for requirements elicitation activity are listed: Observation, Protocol Analysis, Scenarios, Interview, JAD, Prototyping, Quality Function Deployment, and Social Analysis.

In Domain Design, Figure 4 shows the work process with its commonalities and variabilities. Variation points (dotted line activities) can be customized by the avaliable components (variants) based on dependency rules shown in Table 1.

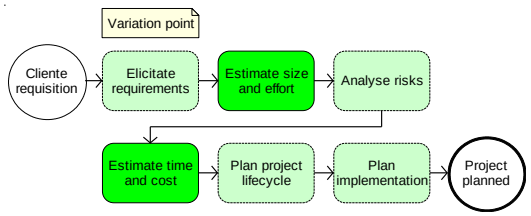

Figure 4: Core process with variation activities.

Table 1: Rules.

Feature model rules	Interview *requires* Social Analysis
	JAD *excludes* Interview
Composition rules	Observation *goes before* Interview
	Scenarios *goes before* Prototyping
	Social Analysis *goes before* Prototyping and Scenarios
Context Definition	Collaborative organization environment: group working experience = high, team experience >= medium
	Complex development: problem size = high, problem complexity – high, client relationship <= medium
Context rules	Collaborative organization environment *implies* JAD
	Complex development *implies* Prototyping *and* Interview

The process components model (Figure 5) represents internal properties (incomes, outcomes, data, etc), process behavior and state transitions, implementation scheme, and interfaces.

In the Domain Implementation, the process architecture extends the infrastructure presented by Gatti et al (2010), which establishes an activity context-aware architecture to support knowledge management in processes.

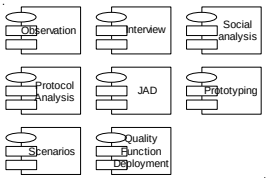

Figure 5: Process components model.

Thus, having the context-based process line created, for each new project, the organization can use the composition mechanism, creating new process through context interpretation.

In Application Design the mechanism helps the project manager to make up the requirements elicitation activity, composing the four techniques as shown in Figure 6. The mechanism suggested applying Social Analysis, Scenarios and Prototyping in this sequence. The project manager also decides to parallelize the interview activity with the Social Analysis. In the Application Implementation, the project manager decides to use a BPMS to automate the process and starts to run.

During its execution, he/she realizes that the team has gain synergy and people are working more collaboratively. The automated mechanism also recognizes this situation by the context information continually collected. The project manager decides to adapt the process substituting the Interview technique by JAD. The automated mechanism also realizes that the Social Analysis technique is not needed anymore, but he/she decides to maintain it because the client and also the team happened to like this approach. The adapted process goes as shows Figure 7 and the context monitoring continues.

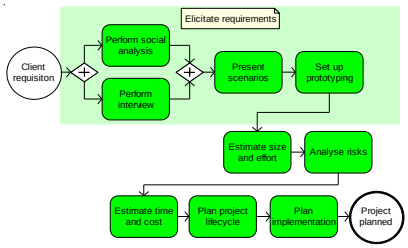

Figure 6: Composed process.

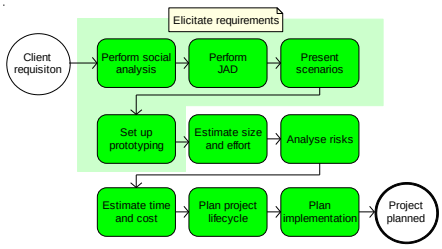

Figure 7: Adapted process.

5 CONCLUSIONS AND FUTURE WORK

This paper aimed to describe a proposal for process line engineering, based on product line engineering and introducing the manipulation of context to address the issue of evolution and adaptation of processes at run time.

The handling of context information is made through the extension of the model proposed by Nunes et al., (2009), which was adapted to support the concepts described in process line. The architecture described in Gatti et al. (2010) is being adapted to implement the proposed model. We are also conducting studies to identify the best representation of context information, such as features and ontologies.

ACKNOWLEDGEMENTS

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

REFERENCES

Abels, S. et al., 2006. PROMONT – A Project Management Ontology as a Reference for Virtual Project Organizations. In On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. pp. 813-823. http://dx.doi.org/10.1007/11915034_105 [Accessed January 20, 2010].

Barreto, A., 2007. Uma Abordagem para Definição de Processos de Software Baseada em Reutilização, Universidade Federal do Rio de Janeiro / COPPE.

Bayer, J., Buhl, W., Giese,C., Lehner, T., Ocampo, A., Puhlmann, F., Richter, E., Schnieders, A., Weiland, J., and Weske, M, 1995. Process family engineering. Modeling variant rich processes. Technical report.

Chen, G., Kotz, D., 2000. A Survey of Context-Aware Mobile Computing Research, Dartmouth College. In: http://portal.acm.org/citation.cfm?id=867843.

Clements, P., Northrop, L., 2001. Software Product Lines: Practices and Patterns 6th ed., In: http://www.informit.com/store/product.aspx?isbn=0201703327.

Dey, A., Abowd, G., Salber, D., 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction, 16(2-4), 97-166.

Dumas, M., Aalst, W.V.D., Hofstede, A.T., 2005. Process-aware information systems: Bridging People and Software through Process Technology, John Wiley and Sons, NY, USA.

Fusaro, P., Visaggio, G., Tortorella, M., 1998. REP - ChaRacterizing and Exploiting Process Components: Results of Experimentation. In Proceedings of the Working Conference on Reverse Engineering (WCRE'98). IEEE Computer Society, p. 20. In: http://portal.acm.org/citation.cfm?id=837006.

Gatti, L.A., Santoro, F.M., Nunes, V.T., 2010. An Agent-Based Architeture for Knowledge Management in Context-Aware Business Processes. Computer-Supported Cooperative Work in Design 2010.

Grisogono, A., 2006. Success and Failure in Adaptation. In International Conference on Complex Systems (ICCS2006).

Jaufman, O., Munch, J., 2005. Acquisition of a project-specific process. In 6th International Conference on Product Focused Software Process Improvement, PROFES 2005, June 13, 2005 - June 18, 2005. Oulu, Finland: Springer Verlag, pp. 328-342.

Kang, K. et al., 1990. Feature-Oriented Domain Analysis, CMU-SEI. In: http://www.sei.cmu.edu/domain-engineering/FODA.html.

La Rosa, M., 2009. Managing variability in process-aware information systems. In: http://eprints.qut.edu.au/20531

Montero, Pena, Ruiz-Cortés, 2007. Business Family Engineering: Does it make sense? II Congreso Español de Informática (Cedi 2007), (Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos (Tjisbd)), 34-40.

Nunes, V.T., Santoro, F.M. & Borges, M.R.S., 2009. A context-based model for Knowledge Management embodied in work processes. Inf. Sci., 179(15), 2538-2554.

Rombach, D., 2006. Integrated Software Process and Product Lines. In Unifying the Software Process Spectrum. pp. 83-90. In: http://dx.doi.org/10.1007/11608035_9.

Schnieders, A., 2006. Variability mechanism centric process family architectures. In Engineering of Computer Based Systems, 2006. ECBS 2006. 13th Annual IEEE International Symposium and Workshop on. p. 10 pp.

Washizaki, H., 2006. Building software process line architectures from bottom up. In 7th International Conference on Product-Focused Software Process Improvement, PROFES 2006, June 12, 2006 - June 14, 2006. Amsterdam, Netherlands: Springer Verlag, pp. 415-421.