Metaprocesses as Software Assets Reuse

Javier Darío Fernández-Ledesma
University Cooperative of Colombia, Department of Systems Engineering, calle 40 A N.41-05, Medellín, Colombia
javier.fernandez@ucc.edu.co

Keywords: Metaprocess, Processes, Reuse.

Abstract: Software reuse in the early stages is a key issue in rapid development of applications. Recently, several methodologies have been proposed for the reuse of components, but mainly in code generation as artifacts. However, these methodologies partially consider the reuse of abstract models built for domain analysis and business modeling. This article introduces a metaprocess-oriented methodology based on the model reuse as software assets, and starting from the domain specification and analysis phases. The approach includes the definition of a conceptual level to adequately represent the domain and a reuse process to specify the met-aprocess as software assets. The methodology has been applied successfully in the field of e-health; in particular a monitoring system for patients with cardiovascular risk has been modelled and analysed.

1 INTRODUCTION

Software reuse starting from the early stages of software development facilitates rapid development of applications. This contributes to increased productivity and quality in software development [8].

Metaprocess metamodeling and its specification as software assets for reuse in the early stages is a field in which much effort is needed to standardize software processes in the context of software industrialization.

The purpose of this article is to present a metamodeling-based approach intended to provide a conceptual perspective for a domain, and its specification as software assets to improve metaprocess reuse in early stages of software development. To illustrate the applicability of our proposal, this article also presents a real example currently running on the e-health domain: a monitoring system for patients with cardiovascular risk factors.

The remainder of the article is organized as follows. In section 2 we discuss related work. Section 3 provides a description to the proposed approach. Finally, we present the conclusions in Section 4.

2 RELATED WORK

There are several proposals regarding definition of metaprocesses in the context of software development. A metaprocess can be defined as a complete process meta-model that serves as a benchmark to be instantiated, including different cases or situations for the same domain. This metamodel contributes to the generalization of processes through its metamodeling, specification of methods, decomposition of tasks, and rules of consistency. Rolland and Prakash (2000) conceive metaprocesses as artifacts with general features for instantiation, customization and gradual refinement of processes and models. Therefore, Metaprocesses are generic specifications of activities, tasks, roles, and behaviors supporting the execution of processes with the main objective of obtaining an abstraction of the domain.

There are several proposals regarding metaprocesses in the context of software development, such as metaprocesses as metamodeling-based models (Conradi and Nguyen, 1994) or methodologies for process-oriented software development (Finkelstein et al., 1994). However, none of them provide descriptions of the reuse of the metaprocess specification as software assets for instantiation and customization in the early stages.

Greenfield and Short (2004) define "[...] metaprocess-oriented methodology can con-tribute to raise productivity and quality in the software construction process ". In particular, a methodology that takes into account the abstraction of the domain specification and process modelling, as well as its specification as software assets for reuse in the early stages can ensure shorter development times before

FernÃąndez-Ledesma J.
Metaprocesses as Software Assets Reuse.
DOI: 10.5220/0006224102210224
In *Proceedings of the Sixth International Symposium on Business Modeling and Software Design (BMSD 2016)*, pages 221-224
ISBN: 978-989-758-190-8

the release of applications. This can also reduce costs in areas with strong demand for the development of rapid applications in domains such as e-health.

Nowadays, there are several approaches that deal with the use of metaprocesses, but, in general, reusability is not taken into consideration in the early stages of software development. Our objective is to consider metaprocesses, as part of a generic environment, that integrally support model-oriented software development, taking explicitly into consideration the specific domain.

Ramsin and Paige (2008) incorporate the use of metaprocesses as an oriented methodology towards model-driven software development. Their work describes what has been accomplished in terms of component reuse, but does not describe reuse of models from the early stages of software development. Ouyang et al. (2009) present a methodological proposal which is oriented to business processes models with BPMN and it translate to BPEL. Ouyang et al. (2009) Proposes a translation technique; but it don´t consider the models reuse. Additional works have proposed a systematic approach to modelling that accompanies the use of metaprocesses in all phases of software development, starting from the early stages, in order to understand and analyse the domain, to design the software solution and build its implementation. Kühne (2009) incorporates the concept of metaprocesses into the evolution of software processes.

Asikainen and Männistö (2009) consider the need to semantically formalize software development processes using the metamodel processes. Levendovszky et al., (2009) incorporate the use of process metamodel patterns as a first step in formalizing specifications. However, these approaches do not take into account the issue of model reuse and its formalization in the early stages of software development; they use independent platform models and their implementation from the domain, but none of them include the problem of instantiation and customization through the reuse of metaprocesses.

Cechticky et al., (2006) proposes to reuse code components for real time applications. This proposal is based on the reuse of code, but does not cover models and metaprocesos. Park et al (2007), use code components as software assets to facilitate reuse of these components, which is done independently from models. De Freitas (2009) incorporates flexibility through the reuse of application code, without addressing the issues of model and metaprocess reuse for generating applications. Finally, Rodriguez et al (2011) propose reusing design knowledge, in order to reduce costs and difficulties in software development, as a contribution to the methods developed in collaborative systems such as groupware design patterns, pattern languages and frameworks. These other proposals do not include the theoretical and conceptual articulation of metaprocess usage in software development through the fostering of reuse, instantiation and customization using platform independent architectures, and the use of models and metamodels as a contribution to software industrialization.

3 A METHODOGICAL PROPOSAL FOR THE REUSE OF METAPROCESSES

This section introduces a methodology that consists of one level design for the description of metaprocesos (conceptual level) and a reuse process (see Figure 1). The representation and construction of metaprocesos begin from a generic metamodel, which is specified using different models to fully represent the domain, with customization applied at different development stages, from specification of requirements to design and software applications. We seek for a sufficiently expressive and complete representation of metaprocesos in order to cover the conceptual domain with elements which facilitate the reuse in the software development.

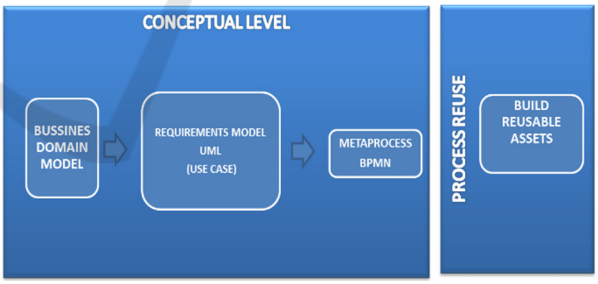

Figure 1: Metaprocesses specification levels as software assets expressed in models.

The conceptual level is a generic abstraction of the conceptual domain, which is represented through the business domain, requirements, and process models. The business domain model specifies the business cycle: mission, policies, business and process elements. The requirements model is based on use case diagrams. The metaprocess is based on the use of the BPMN standard. A metaprocess will be specified, as described below, on the basis of the business requirements captured through use cases and

process elements of the conceptual domain and their relationships.

Consequently, the conceptual level is specified using three steps as depicted in Figure 1:

1. The first step encompasses the representation of the domain through the Business Domain Model (BDM). BDM specifies business activities, business tasks and business roles. BDM contains diagrams such as the business diagram and process flow diagrams. The process flow diagram specifies the business activities and roles. In this way, a business activity is conducted using one or more activities process.
2. The second step is intended to build the Requirement Model (RM), including use case diagrams.
3. Finally, the third step is to build the Process Definition Model (PDM), which includes business process functionality both from a domain perspective and from the perspective of system support implementation for this software component.

This is followed by the metaprocess construction, based on information provided by the Domain Analysis, using the BPMN Notation, in which the activities (task in BPMN) and roles (swimlanes in BPMN) of the Metaprocess are clearly identified, as well as the use cases and applications or systems that support the execution of the metaprocess. Each activity and each role of metaprocess has been specified with a set of applications or systems or part thereof that support it execution for reuse in other cases as components reuse.

The reuse process enables the representation of metaprocesos as reusable soft-ware asset. Hence it represents the metaprocess architecture with its constituent elements specified as software artifacts. The specification of reusable software assets metaprocess as according to the standard is done by identifying each metaprocess as a reusable software asset with its component artifacts and attributes that describe them. It is defined by use, solution and classification, profile, related profiles and related assets as reusable software component elements of asset.

The reuse process is based on the OMG-AS and OMG's RAS standard, and uses a repository of reusable software asset to enable storage and search of assets in packed files, they contain these assets and an XML manifest (XML Schema) and, they are specified as .xsd as rasset.xml residing in the root directory of the asset accompanied by the respective. XSD or XML-Schema and another set of artifacts, files or subdirectories that help specify. There components are compressed into a single file with a .Ras to facilitate management of its reusable software assets.

Finally, this propose uses the mechanisms for building well-formed models and Reusable Asset Specification (RAS).

The mechanisms for RAS facilitate the search and retrieval elements of the metaprocess as software assets (models, components, artifacts) into repositories. The OMG- RAS standard proposes the organization of the files .Ras and rasset.xml file structure, assets can be searched, retrieved and sail them through services, which can be implemented as Web Services or other approaches, which states for each service the nature of the response and the response.

In this case the implementation of mechanisms for RAS it's possible through our own repository *Actives*.

4 DISCUSSION AND CONCLUSIONS

This methodological approach contributes to specifying domains by means of conceptual levels. These levels facilitate the creation of design models independently from the platforms. In this manner, it is possible to obtain an under-standing of the domain with the purpose of correcting problems inherited through deficient requirements gathering or a lack of comprehension of the same. As a result, we obtain specific elements at a conceptual level that can be reused in the development of future applications.

Our approach follows the Model-Driven Development (MDD) paradigm, through transformation of models between the early stages of the software development process.

The methodology proposed for the metaprocess at the conceptual specification level as software assets for reuse in the early stages of software development is in-tended to facilitate the development of domain process oriented applications, in this case for e-health. This methodology facilitates the software development process in one case, in which guided models contributed to the development of applications from the domain, independently from the development platforms. Now, models, metaprocesos, components and artifacts are being used to develop other systems, such as an interoperability platform for a pre-hospital system.

The monitoring system for patients with cardiovascular risk, and the interoperability platform

for the pre-hospital domain has been implemented at IPS University Hospital (Medellin, Colombia), and this system is being requested by other countries and other regional hospitals in Colombia.

The system has been evaluated by measuring its impact on the indicators. Briefly, the statistics and analysis of its implementation indicate that there is a significant improvement in the allocation of hospital resources and patient care times.

As future work, we plan to formalize the methodology through the use of logic languages using a formal definition Noguera et al. (2010), as well as considering the use of patterns and their mechanisms, for model reuse, as software as software assets not specified in the OMG standard.

ACKNOWLEDGEMENT

We thank DINAI of University Cooperative of Colombia.

REFERENCES

Acuña, S. & Ferré, X. Software Process Modeling. pp.1-6 In: Proceedings of the 5th. World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001). Orlando Florida, USA. , (2001)

Asikainen, T. & Männistö, T. Nivel: a metamodeling language with formal semantics, pp 521-549. Software & Systems Modeling. Volume 8, N. 4, (2009)

Baisley, D., Björkander, M., Bock, C., Cook, S., Desfray, P., Dykman, N., Ek, A., Frankel, D., Gery, E., Haugen, Ø., Iyengar, S., Kobryn, C., Møller-Pedersen, B., Odell, J., Övergaard, G., Palmkvist, K., Ramackers, G., Rumbaugh, J., Selic, B., Weigert, T., Williams, L.: OMG Unified Modeling Language (OMG UML), Superstructure v 2.2. OMG (February 2009)

Cechticky, V., Egli, M., Pasetti, A., Rohlik, O., Vardanega, T. A UML2 Profile for Reusable and Verifiable Software Components for Real-Time Applications, pp.312-325. ICSR 2006, LNCS 4039, (2006)

Conradi, R. & Nguyen, M. Classification of Metaprocesses and their Models. pp 167-175. Software Process, (1994)

De Freitas, J. Model business processes for flexibility and re-use : A component-oriented approach, pp 1-11. IBM Developer Works Journal, (2009)

Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., Nuseibeh, B. Software Process Modeling and Technology, Research Studies Press LTD. Londres. (1994)

Greenfield, J & Short, K. Software Factories: Assembling Aplications with Patterns, Model, Frameworks and Tools. John Wiley & Sons. (2004)

Kühne, T. Editorial to the theme issue on metamodeling, pp. 447-449. Software & Systems Modeling. Volume 8, N. 4, (2009)

Levendovszky, T., László, L., Mészáros, T. Supporting domain-specific model patterns with metamodeling, pp 501-520. Soft. & Systems Modeling. Vol 8, N 4, (2009)

Noguera, M., Hurtado, M., Rodríguez, M., Chung, L., Garrido, J. Ontology-driven analysis of UML-based collaborative processes using OWL-DL and CPN, pp 726-760. Science of Computer Programming. Volume 75, (2010)

OMG. Software & Systems Process Engineering Meta-Model Specification doc.ormsc/(2008-04-01)

OMG: Business Process Model and Notation (BPMN) v1.2. Object Managment Group (OMG) (2008)

OMG. Reusable Asset Specification. OMG Available Specification Version 2.2. (2005)

Ouyang, C., Dumas, M., Van der aalst, W., Ter hofstede, A., Mendling, J. From business process models to process-oriented software systems. CM Trans. Soft. Eng. Methodologies. 19, 1, Art. 2 (August 2009)

Park, S., Park, S., Sugumaran, V. Extending reusable asset specification to improve software reuse, pp 1473-1478. Proceedings of the 2007 ACM symposium on Applied computing - SAC '07, (2007)

Ramsin, R. & Paige, R. Process-Centered Review of Object Oriented Software Development Methodologies. pp 1-89. Computing, Volume 40, N. 1, (2008)

Rodríguez, J., Ochoa, S., Pino, J., Herskovic, V., Favela, J., Mejía, D., Morán, A. Towards a reference architecture for the design of mobile shared workspaces. pp 109-118. Future Generation Comp. Systems. Vol 27 (2011).

Rolland, C., Prakash N. On the Adequate Modeling of Business Process Families. Université Paris1 Panthéon Sorbonne. Francia, (2000)

Uribe, C., Isaza, C., Florez, J. Qualitative-Fuzzy Decision Support System for Monitoring Patients with Cardiovascular Risk, pp 1621-1625. Proc: Conf. on Fuzzy Systems and Knowledge Discovery. Vol. 3, (2011).