Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 2354–2362

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

A Domain-specific Rule Generation Using Model-Driven Architecture in Controlled Variability Model

Neel Mani*, Markus Helfert [a], Claus Pahl[b]

[a] *ADAPT Centre for Digital Content Technology, Dublin City University, School of Computing, Dublin, Ireland*
[b] *Free University of Bozen-Bolzano, Faculty of Computer Science, Bolzano, Italy*

Abstract

The business environment changes rapidly and needs to adapt to the enterprise business systems must be considered for new types of requirements to accept changes in the business strategies and processes. This raises new challenges that the traditional development approaches cannot always provide a complete solution in an efficient way. However, most of the current proposals for automatic generation are not devised to cope with rapid integration of the changes in the business requirement of end user (stakeholder's and customer's) resource. Domain-specific Rules constitute a key element for domain specific enterprise application, allowing configuration of changes, and management of the domain constraint within a domain. In this paper, we propose an approach to the development of an automatic generation of the domain-specific rules by using variability feature model and ontology definition of domain model concepts coming from Software product line engineering and Model Driven Architecture. We provide a process approach to generate a domain-specific rule based on the end user requirement.

Peer-review under responsibility of KES International

Keywords: Rule Generation; Domain-specific rules; Business Process Model; Variability Model; Model Driven Architecture;

1. Introduction

Nowadays, Enterprises struggle with rapid changes due to the dynamic and competitive nature of the

* *E-mail address:* neel.mani2@mail.dcu.ie

1877-0509
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.206

environment. The businesses need to adapt and implement changes promptly. The changes occur almost at every front of the business process. For example, changes in demand of customers, changes in business strategies (internal and external stakeholder) or changes in various laws. Process model's languages[1] provide expressive and various verification techniques to ensure the reliable processes[2]. However, the languages restrict domain experts to make changes such as defining explicitly the process execution plan as pre-defined task control flow, data flow, and work/process allocation schema, etc. The changes reflect at the modeling stage or design phase, which makes the process model rigid[3]. These may pose challenges in customization, adaptability and maintenance of the system. These languages limit flexibility of enterprises[4,5] and their suitability (or sustainability) for the dynamic environment. As the nature of organisations is volatile (polices of business) and processes are often excessively rigid, Domain specific solution will make process model more focussed for a particular domain in statically and dynamically adaptable in terms of the process execution plan, reducing the dependency on programming, and to software developers.

Advent of model approach in Business Process Model (BPM) systems, domain or business experts usally work on high level of model designing, and maintaining the complex behaviours of their enterprise application. The software professional or programmers are using work on very low level of code who can modify code. Since, Domain experts are often working, designing, learning and thinking in critical way to solve the complex process of enterprise in form of decision-making rules, and policy, their intent is rather simple to understandable. The enterprises are looking for a new standard configurable domain solution for defining the rule, expressing and facilitating their integration for process model constraint. Domain expert and business expert(stakeholder) often work on very high level of abstraction (design model and use modeling language and designing tools) and think in conditional rules, mapping programmer and software expert intent into low level of programming code.

The several steps required for implementation of model transformation and configuration system. These are the steps of high-level design to low level execution. Enterprises, usually, have high level of legacy model and design as a domain model or process model. Automatic code generation is a well-known process of getting the low level of executable code from a given abstract model. Rule generation is the process by which higher level model is translated or transformed into the lower level program. It is a process of conversion of one form to another; it may be platform specific or platform independent or generic[6]. In Model-Driven Architecture(MDA)[7], techniques are expressed by models as the primary development artifact and use them as a basis for obtaining a configurable domain-specific rule for process model customisation in different ways[8], but it does not talk about variability of models (domain model and process model).

We are using digital content process domain for a case study. The web application's machine translation translates source language to target language. The system provides a web based platform where domain expert can edit the generated domain-specific rule in the natural language configurable editor. In this paper, we focus on two main elements: (1) How domain-specific rules(DSR)[9] can be generated from domain models automatically? (2) what are processes of approach to get a rule from domain model and what are the core component models with variability model. We also discuss the benefit of domain models, and configurable rule in process model customisation.

We discuss the related work in Section 2. In Section 3, we proposed approach and core component of this research. In Section 4, we describe the integration of process approach for DSR generation. Then, we saw how the rule generation approach could be implemented using domain model, process model and variability model the solution implementation in Section 5. We finish with an evaluate of the solution in Section 6 and some conclusion with future work in Section 7.

2. Related Work

The use of rule and ontology modeling formalism as the MDD source model, along with defining a metamodel to put it into the framework of MDA has been our proposal. Abdullah et al. (2007) [10], who proposes the possibility of depicting the profile elements to a Jess platform-specific representation, also embarks upon the idea of using a UML profile for the framing of knowledge.

Hecht, Piveta, Pimenta and Price[11] uses high-level programming as well as code generation approach in expediting the process of alteration between software design and its implementation in executable code. In order to