Information Systems 72 (2017) 77–94

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Designing business capability-aware configurable process models

Wassim Derguech [a,*], Sami Bhiri [b], Edward Curry [a]

[a] Insight Centre for Data Analytics, National University of Ireland, Galway (NUIG), Ireland
[b] OASIS, National Engineering School of Tunis, University Tunis El Manar, Tunisia

ARTICLE INFO

Article history:
Received 2 June 2017
Revised 2 October 2017
Accepted 3 October 2017
Available online 12 October 2017

Keywords:
Process aware information systems
Business process modeling
Configurable business process modeling
Merging business process models
Business capability

ABSTRACT

Process Aware Information Systems manage processes within organisations on the basis of business process models. These models can be created either from scratch or by reusing exiting reference process models.

Particular types of reference models are configurable process models that are created by merging multiple models into a single one that can be customized to the needs of the business experts. Using those models presents two main challenges: their creation and their configuration.

In this paper, we focus on the first challenge and propose a novel algorithm for merging process models into a configurable process model. The difference in our work is the pre-annotated process models with their business capabilities that report on what actions each process element achieves. Our algorithm generates configurable models that are also annotated with their capabilities that can be used to face the second challenge of these models: the configuration phase.

We tested our algorithm using real-world process models to evaluate the required creation time and resulting compression rate after merging the input models. The results show that the models can be created in few milliseconds and achieving a compression rate of 50%. We further carried out interviews with domain experts to assess the usefulness and the level of maturity of this work. The results show the importance of the automation of process merging using a tool support that we proposed. However, further adaptation efforts are required to integrate this work in the working environments of the interviewed experts.

1. Introduction

Reference process models describe proven practices for a specific industry. They are often aligned with emerging industry-specific and cross-industry standards [1,2]. One of the scenarios of use of reference process modelling is the reference process model customization [3]. It begins with a reference process model that provides configuration facilities. This model can be configured to specific needs of an enterprise e.g., by refining business rules or enabling/disabling some activities. Such reference models are called configurable business process models [4]. It is a reference model that can be tailored by end-users in order to meet their requirements and satisfy their business needs [4]. The management of such models, brings various challenges for their creation and configuration.

The basis of a configurable business process model is the integration of multiple behaviours of business processes into a single model. These behaviours are captured in *various* business process models that are called *business process variants* [4]. Configurable process models are constructed either via mining techniques [5,6] or the manual or automated merging/aggregation of several variants of a process model [4,7–10]. Manual creation of configurable process models is tedious, time-consuming and error-prone task. It requires the identification of common process parts, merging them and explicitly representing differences between models in terms of configuration options. The literature provides several approaches to overcome this challenge [5,9,11], the main issue with such approaches is that the resulting configurable models capture their configuration options in terms of model restrictions that are difficult to manipulate by end-users during the configuration phase.

The configuration of these reference models consists of enabling/disabling several branches of the model through manipulating configuration options [12]. This phase is difficult and requires advanced modelling skills for identifying and selecting the configuration options. Furthermore, the users cannot determine the impact (i.e., what functionality are they enabling or disabling from the configurable model) of each configuration decision they take unless they manually trace each branch of the configurable node and determine the functionality resulting from each of them. This

* Corresponding author.
E-mail addresses: wassim.derguech@insight-centre.org (W. Derguech), sami.bhiri@gmail.com (S. Bhiri), edward.curry@insight-centre.org (E. Curry).

https://doi.org/10.1016/j.is.2017.10.001
0306-4379/

can be resolved by creating an explicit link between the model configurations and the domain requirements and lifting the configuration phase from manipulating model restrictions to domain requirements. La Rosa [12] proposed to model domain requirements as a set of questions with answers explicitly linked to configuration options. In this case, the configuration phase consists on answering these domain related questions. Even though this solution helps in guiding the configuration, it requires a lot of manual work for creating these questions and linking them to the model restrictions.

The contribution of this paper is an algorithm that allows merging a pair of business capability-annotated process variants given as input and delivers a business capability-annotated configurable process model. Several methods have been proposed to merge business process variants [7–10], however, their main weakness resides in the fact that they do not consider tasks capabilities for matching business process tasks. They rely exclusively on the task labels for this operation. In contrast to existing proposals, this paper uses capabilities for matching similar tasks in different models. The resulting configurable model is also annotated with capabilities which facilitates the configuration and individualization steps [4,12,13].

In order to carry out a quantitative evaluation of the merging algorithm proposed in this paper, two main metrics are considered: *time* required for merging business process models and the *compression rate* gained after the merging operation. These two metrics have been used by La Rosa et al. [9] for evaluating their business process merging algorithm.

- *Time*: for organisations, time is important and should not be spent on manual creation of configurable models. La Rosa et al. [9] mentioned that it took a team of five analysts and 130 man-hour to merge *manually* 25% of an end-to-end process model. Therefore, an automation support for merging business process variants is needed to help saving time and money.
- *Compression rate*: the compression of a repository of business process variants into a single configurable model has multiple benefits: guaranteeing consistency between business process models, avoiding business process clones [14], etc.

This paper evaluates also the proposed algorithm with respect to a set of requirements that the have been used previously in the literature:

1. *[Behaviour Subsumption]* The merged model should allow for the behaviour of all the original models. Traditionally, the merging operation is manually made by business analysts which comes with the risk that some aspects of the original models are accidentally neglected [7]. With automation support for merging process variants, this risk can be minimized considerably.
2. *[Traceability]* Each element of the merged process model should be easily traced back to its original model [9,10]. A business analyst needs to understand what the process variants share, what are their differences, etc. This can be made possible if they can trace back to the variant from which an element originates.
3. *[Deriving Original Models]* Business analysts should be able to derive the input models from the merged process model [9,10].

The remainder of this paper is organized as follows: Section 2 further describes the concept of configurable business process models and introduces the formal definition of a capability-annotated configurable business process model. Section 3 introduces a running example that will be used in the rest of the paper. Section 4 presents the merging algorithm. Section 5 reports on the implementation and validation of the algorithm. Section 6 analyzes the related work and Section 7 concludes the paper and discusses future research directions.

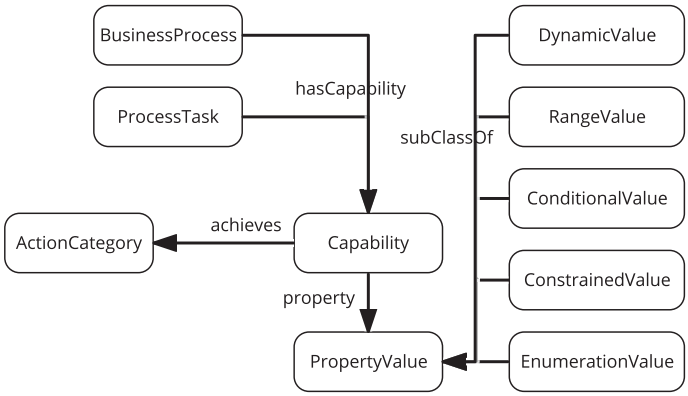

Fig. 1. Snippet of the business capability meta-model.

2. Basic concepts

2.1. Business capability

An important concept used in this paper is the *Business Capability*. It has been defined in the literature from various perspectives:

- From an organizational and resource perspective: Organisational Capability: the ability of organizations to efficiently use their resources (i.e, human capital, knowledge, available data, etc.) to generate value and achieve their objectives [15,16].
- From a control flow perspective: Planning Capability : the way organizations achieve their goals by capturing explicitly process tasks and their temporal and logical order [17].
- From a service perspective: IT Capability: the effect of a service in terms of data generated or change of the world [18] that are explicitly represented in terms of Inputs, Outputs, Preconditions and Effects (IOPE for short).
- From a functional perspective: Business Capability: the action performed by a service, computer program, etc. that creates a value for the customers [19].

In this paper, we consider the business capability from a functional perspective. We argue that this concept is highly required for describing what is being achieved by enterprise services, business processes and tasks. As depicted in Fig. 1, we propose to model a business capability as an action category enriched by (zero or many) functional or non-functional properties. These properties refine the given capability by giving more details about aspects of interest of the corresponding action.

More formally, in the proposed model, capabilities are defined as a Category and a set of property entries (see Definition 1). A property entry is a couple (property, value) where *property* is a domain-specific functional feature or a domain-independent non-functional property and *value* is the value or the possible values that a property can have. Both *property* and *value* refer to ontological terms.

Definition 1 (Business Capability)**.** A couple *Cap = (Category,Properties)* is a business capability, where:

- *Category*: This concept is similar to [19] that defines, in a natural language, what is the action being described. Different to [19], we consider the category as a concept from a domain related ontology that comes form a shared agreement on its semantics. A category is a specific property that is present in all business capability descriptions via the property *achieves* (see Fig. 1).
- *Properties*: Represents a set of pairs *(Property, Value)* that correspond to the set of features of the business capability.