Management of multi-language business processes with "AProMoRe"

Marie-Christine Fauvet, M. La Rosa, A. Alshareef, M. Sadegh, R.-M. Dijkman, Marlon Dumas, G. Garcìa-Bañuelos, Jan Mendling, H.-A. Reijers, W.M.P. Aalst

Management of multi-language business processes with APROMORE*

M.C. Fauvet[1**], M. La Rosa[2], A. Alshareef[2], M. Sadegh[2], R.M. Dijkman[3], M. Dumas[4], L. Garcìa-Bañuelos[4], J. Mendling[5], H.A. Reijers[3], W.M.P. van der Aalst[3],

[1] LIG, University of Grenoble, France
[2] BPM Group, Queensland University of Technology, Australia
[3] Eindhoven University of Technology, The Netherlands
[4] University of Tartu, Estonia
[5] Humbolt-Universität zu Berlin, Germany

1 Introduction

APROMORE (Advanced PROocess MOdel REpository) is an open and extensible platform meant to face the challenge of how to deal with an increasing number of business process models within or accross organisations. To address the issues which are raised in in this context, it becomes essential to keep track of the various models as they may refer to each other, mutually overlap, supersede one another and evolve, while represented in different notations such as EPCs, BPMN, YAWL, etc. APROMORE provides features to store and disclose business process models of a variety of types and languages. Beyond that, APROMORE is built on top of existing tools and techniques, to provide state-of-the-art features for model presentations, analysis, integration, comparison, improvement an re-use.

The demonstration aims at providing an overview of APROMORE capabilities regarding import, export and edition of business process models captured in various notations. The main contribution is APROMORE support for dealing with models in different notations, and maintaining the relations that exist between models.

2 Multi-language business process representation

APROMORE has adopted an canonical format for process definition which deals with the multitude of process modeling notations [1]. The canonical process format (CPF) provides a common, unambiguous representation of business processes captured in different notations and/or at different abstraction levels, such that all process models can be treated alike. The idea behind this format is to represent only the structural characteristics of a process model that are common in the majority of modeling languages. Language-specific concepts are omitted because they cannot be meaningfully interpreted when dealing with process models originating from dierent notations, i.e. when

* brahms0.imag.fr:8080/Apromore-portal, login as icsoc, no password.

** Corresponding author: marie-christine.fauvet@imag.fr. This work has been done when she was visiting fellow at QUT - BPM research group, Brisbane, Australia.

cross-language operations are performed. Moreover, this canonical format is agnostic to graphical information such as shapes, line thickness and positions, which is contained in a concrete process definition. This information is stored separately in the form of annotations (APF), and only used when a canonical model needs to be presented to the user or converted to an original format (NPF).

3 Architecture

APROMORE has been implemented as an open-source SAAS (Software-As-A-Service)[6], and is based upon 7 services independant from each other (see Figure 1).

Apromore portal is a web application which, by the means of graphical interfaces, provides users operations to access, store, retrieve, manipulate, etc. process models. The portal offers limited functionalities to anonymous users while registered users have a choice of features depending on their roles. Apromore-portal delegates editing capabilities to existing editors (i.e. Oryx[7] for BPMN models). *Manager* service exposes operations needed by other applications and services to access process models. *Canoniser* is responsible for performing canonization and de-canonization of process models given in certain native format. *Toolbox* service encapsulates operations to be performed on process models. *Data access* service encapsulates data-centric operations responsible for writing/reading data upon requests made by other services while *Algo. Access* encapsulates accesses to algorithms.

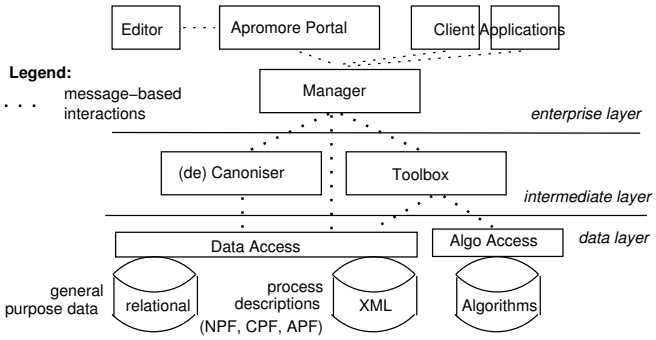

Fig. 1. APROMORE architecture

References

1. M. La Rosa, H.-A. Reijers, W.-M.-P. van der Aalst, D. R.-M., J. Mendling, M. Dumas, and L. Garcia-Bañuelos. Apromore: an advanced process model repository. Submitted.

[6] Sources are available at http://code.google.com/p/apromore/

[7] bpt.hpi.uni-potsdam.de/Oryx/WebHome