Business Process Variability: A Tool for Declarative Template Design[*]

Pavel Bulanov, Heerko Groefsema, and Marco Aiello

Distributed Systems Group, Johann Bernoulli Institute, University of Groningen,
Nijenborgh 9, 9747 AG Groningen, The Netherlands
{p.bulanov,h.groefsema,m.aiello}@rug.nl

Abstract. To lower both implementation time and cost, many Business Process Management tools use process templates to implement highly recurring processes. However, in order for such templates to be used, a process has to adhere substantially to the template. Therefore, current practice for processes which deviate more than marginally is to either manually implement them at high costs, or for the business to inflexibly comply to the template. In this paper, we describe a tool which demonstrates a variability based solution to process template definition.

Template utilization is a well known practice in the BPM domain. As an attempt to lower development time and costs, the practice is not without issues. One such an issue is that processes must adhere significantly to the template in order to achieve the goal of lowering implementation costs. We argue that the introduction of variability to BPM will provide solutions to this and other well known BPM issues [2]. When introduced to the BPM domain, variability indicates that parts of a business process remain either open to change, or not fully defined, in order to support different versions of the same process depending on the intended use or execution context [3]. In this paper, we briefly illustrate a modelling tool demonstrating a variability based solution to process template definition. The tool itself is based upon the Eclipse GMF framework and supports modelling of business processes using the BPMN notation. By extending the BPMN notation with the graphical elements of Process Variability: Declarative'n'Imperative (PVDI), the process modeller is provided with a large degree of template reusability and design flexibility. See [4] for a detailed description.

As an example, consider that the Netherlands consists of a total of 418 highly different municipalities which all have to implement the same services and laws created by the national government. These services and laws however may be implemented at the municipality's own discretion with regard to local requirements. One law which is highly subject to local needs is the WMO (Wet maatschappelijke ondersteuning, Social Support Act, 2006), a law which allows citizens with physical problems to be part of the community by providing wheelchairs, help at home, and home improvements. Figure 1 illustrates the tool used to implement and validate a variant of the WMO process found at a local municipality. Using domain specific information, in this case processes found at several municipalities,

[*] The research is supported by the NWO SaS-LeG project, http://www.sas-leg.net, contract No. 638.001.207.

G. Pallis et al. (Eds.): ICSOC 2011, LNCS 7221, pp. 241–242, 2012.

a process designer can implement a template. This template may range from being either a set of activities to a fully implemented business process. The process designer then introduces a number of PVDI constraints [4] to the template by including extra flows and groups in the business process model. In this case included were a frozen group (I), which allows no changes to its contents, a parallel link (H) and a number of constraint flows (A through G) which enforce elements or groups to follow each other either eventually (A,B,C,E,F,G) or immediately (D), and in a path (A,B,E,F,G) or all paths (C,D). From the finished template variants can then be created and validated. When creating variants a designer may adapt anything in the template except the PVDI elements. When ready, the variant can be validated regarding its compliance with the constraints defined within the template, and any element detecting problems within the variant will display an error (D,G). All the while, the complications of the underlying logic used for validation is hidden from the designer through the easy to understand graphical PVDI elements. The details of each of the elements used and their underlying validation logic can be found in [4]. Furthermore, a more detailed description of this entire life-cycle is available as a video at [1].

In this paper, we illustrated a tool which implements a new process template design technique offering high process design reusability and flexibility, while hiding underlying complexity via simple well understood design elements.

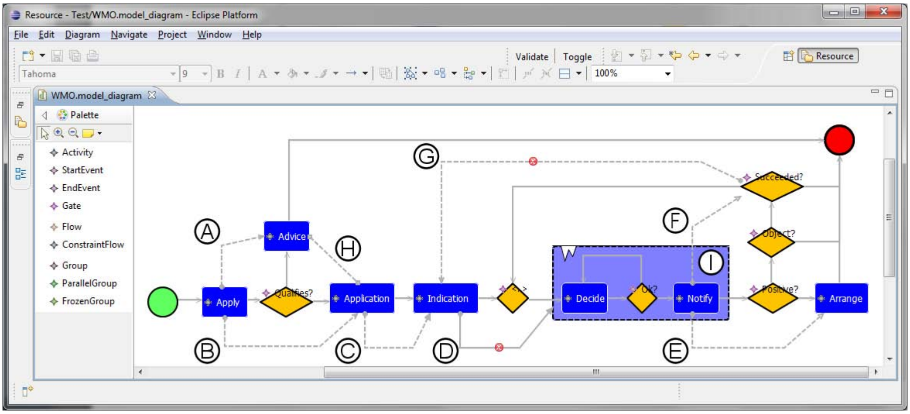

Fig. 1. Screenshot: Validating a variant

References

1. Demo video (2011), `http://www.youtube.com/watch?v=coWkSSRpHD8`
2. van der Aalst, W., Jablonski, S.: Dealing with workflow change: Identification of issues and solutions. International Journal of Computer Systems, Science, and Engineering 15(5), 267–276 (2000)
3. Aiello, M., Bulanov, P., Groefsema, H.: Requirements and tools for variability management. In: IEEE Workshop on Requirement Engineering for Services (REFS) at IEEE COMPSAC (2010)
4. Groefsema, H., Bulanov, P., Aiello, M.: Declarative Enhancement Framework for Business Processes. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 495–504. Springer, Heidelberg (2011)