Bridging the Gap between the Business Processes and the Service Oriented Architecture

Karim Dahman

Bridging the Gap between the Business Processes and the Service Oriented Architecture

Karim Dahman (dahman@loria.fr)
University of Lorraine, UHP - INRIA Lorraine - 54506 Vandoeuvre-lès-Nancy, France

Bridging the gap between the (ever) evolving Business and its supporting IT capabilities is of high relevance in today's enterprise information system roadmap [1]. Without an agile development method and stable IT architectural principles there is no chance to foreshadow a successful route to valuable computerized software solutions. The Service Oriented Architecture (**SOA** [2]) has emerged as a new paradigm for modeling, building and managing *software architectures* to support the Business Processes (**BP** [3]) in loosely-coupled, distributed and continuously changing environments. In many cases, the experience shows that with a Model Driven Engineering (**MDE** [4]), where the *BP models* drive the software applications, less effort is gone into their alignment during the initial development, and IT developers can rapidly proceed with the instantiation of the *technology-neutral SOA implementations* [5]. However, in order to enable BP evolutions without loosing the control on the *BP/SOA alignment* many challenges still need to be addressed. In particular, how to ensure the automated propagation of the BP model evolutions to the SOA model adaptations while keeping them synchronized remains an open question. Therefore, the *BP-to-SOA incremental transformation* is a major use case for the SOA engineering.

Several frameworks have emerged to help organizations to charter successful route to the SOA implementation. The Model Driven Architecture provides a mean for using UML-based model transformations to direct the course of the system development stages [4]. However, as UML is neither fully service-oriented, nor business process-centered, it needs to address *the gap between the emerging process notations and the lower-level standards*. In [6], the authors advocate the integration of the business values perspective to the SOA engineering. They define a conceptual framework for the design space, but unfortunately fail to provide a practical development scenario. In [7], we have presented a MDE scenario that starts off with a *(source) BP model* specification, and that leads to a *canonical (target) SOA model*. The scenario is enough mature to accomplish the automatic model conceptual mappings, however, it does not provide a support for an incremental transformation. First, it requires manual target model adaptations to reflect the source model evolutions. Of course, adapting the generated SOA model is potentially error-prone, leads necessarily to their misalignment with the BP layer, and implies design decision losses. Second, the effort to compute the modifications on the target is proportional to the size of the source model as we have to enforce again a model transformation [8]. What sounds so straightforward and easy in theory with the model evolutions in the MDE, turns out to a very challenging endeavor in practice. The existing *general model synchronization frameworks* [8] cannot work well here. First, they require users to explicitly write synchronization code to deal with each update kind and on

each of the assorted models. Second, the mapping complexity of each combinatorial change is inherently compounded with the decisions regarding the potential information loss or gain related to different levels of the model's expressiveness.

Compared to these frameworks, our approach that we have introduced in [9] extracts informations automatically from the previous transformations and does not require users to write a synchronization code. Given a structural update on the *(source) BP model* and a transformation that has previously generated a corresponding *(target) SOA model*, we developed a *change forward transformation* approach to consistently propagate the BP updates into the related SOA. We consider that *a structural model update corresponds to the execution of a compound sequence of primitive change operations* such as model element additions, removals, and relocations on a graph structures that represent the BP and the SOA models. One way for obtaining a sequence of model update operations is by recording editing operations performed by the BP modelers. For example, the addition of a process fragment can be seen as a composite production of several primitive graph productions on the BP model. The *conditional graph rewritings* [10] allows us to detect the structural update conflicts between pairs of primitive operations. Our transformation trace-aware model synchronizer is implemented in a BP model editor. First, it locates the corresponding constructs in the previous transformation execution context that are affected by the source update. Then, with the identified correspondence and when it is possible over a consistency relation established between the BP and the SCA models, it translates the change operations that consistently update the target with the values from the source. In order, to make the model synchronization effectively incremental, we use the so called *update translation operator* [8]. Thus, the synchronization computations result in a reasonable decoupling from the source model size [10].

The claimed benefits of our MDE include enhanced development principles of the SOA engineering, a better reuse of legacy applications through service wrappings in software components and an easier adaptation to changes in the business environments through model update propagations. As a major contribution, our design-time approach makes the BP model transformation more tractable to reconfigure the SOA implementation without disrupting its structural consistency. Thus, the system architects can rapidly assess the necessity to propagate the change from the BP models to the SOA implementations. Rather than executing the entire transformation afresh and if the SOA update seems to be appropriate, then they can synchronize the *SOA platform-specific implementations* with the IT developers. The experimental results are encouraging, but still needs to be validated in a real-scale case study and tool. The integration of our update translation operator in a previously developed *ATLAS Transformation Language* chain for the *Eclipse SOA Tools Platform* is in development [11]. Furthermore, the BP/SOA round-tripping is essential to rapidly realign BP with the evolving SOA. Likewise, we have to consider update to both the business and IT architectures via pattern-based transformation techniques. Finally, making the MDE more tractable to reconfigure the BP and the SOA implementations without disrupting the non-functional capabilities remain as a future work.

References

1. H.-M. Chen, R. Kazman, and O. Perry, "From software architecture analysis to service engineering: An empirical study of methodology development for enterprise soa implementation," *IEEE Trans. Serv. Comput.*, vol. 3, pp. 145–160, April 2010.
2. M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: State of the art and research challenges," *Computer*, vol. 40, 2007.
3. M. Dumas and T. Kohlborn, "Service-enabled process management," in *Handbook on Business Process Management 1*, ser. International Handbooks on Information Systems. Springer Berlin Heidelberg, 2010, pp. 441–460.
4. D. Lopes, S. Hammoudi, J. Bzivin, and F. Jouault, "Generating transformation definition from mapping specification: Application to web service platform," in *Advanced Information Systems Engineering*, O. Pastor and J. Falco e Cunha, Eds., vol. 3520. Springer Berlin / Heidelberg, 2005, pp. 183–192.
5. S. Brahe, "Bpm on top of soa: experiences from the financial industry," in *The 5th international conference on Business process management*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 96–111.
6. C. Huemer, P. Liegl, R. Schuster, H. Werthner, and M. Zapletal, in *The 2nd IEEE Conference on Digital Ecosystems and Technologies*.
7. K. Dahman, F. Charoy, and C. Godart, "Generation of Component Based Architecture from Business Processes: Model Driven Engineering for SOA," in *The 8th IEEE European Conference on Web Services*, Ayia Napa, Greece, 2010.
8. M. Antkiewicz and K. Czarnecki, "Design space of heterogeneous synchronization," in *Generative and Transformational Techniques in Software Engineering II*, R. Lämmel and J. Visser, Eds. Berlin, Heidelberg: Springer, 2008, pp. 3 – 46.
9. K. Dahman, F. Charoy, and C. Godart, "Towards consistency management in the business-driven development of the soa," in *The 15th IEEE International Enterprise Distributed Object Computing Conf.*, Helsinki, Finland, 2011.
10. H. Giese and R. Wagner, "From model transformation to incremental bidirectional model synchronization," *Software and Systems Modeling*, vol. 8, pp. 21–43, 2009.
11. K. Dahman, F. Charoy, and C. Godart, "From business process to component architecture: Engineering business to it alignment," in *The workshop on Service-oriented Enterprise Architecture for Enterprise Engineering*.