

How to better align BPM & SOA – Ideas on improving the transition between process design and deployment

Sebastian Adam¹, Joerg Doerr¹

¹ Fraunhofer IESE
Fraunhofer Platz 1, 67663 Kaiserslautern, Germany
{sebastian.adam, joerg.doerr}@iese.fraunhofer.de

Abstract. Business Process Management (BPM) and Service Oriented Architectures (SOA) are considered as a powerful combination for supporting enterprise's success. While BPM addresses the (goal-driven) design, deployment, and continuous evaluation of business processes, SOA promises a supporting IT paradigm to make process applications and thus enterprises more agile and flexible. Nevertheless, there is still a gap between business and IT, as the services provided by an SOA cannot - until now - support the business processes immediately. In this paper, we focus on the transition between the design and the deployment of business processes and present our ideas on how this transition could be streamlined by a better alignment of BPM and SOA. Based on systems engineering and product line experience, we identify solution principles and discuss related questions.

Keywords: Business Process Management, Service-oriented Architecture, Business IT Alignment, Product Line Engineering

1 Why SOA does not really improve BPM yet

Business processes have become increasingly important in many enterprises. They determine the procedure for developing and distributing value for the customers and are key drivers for the three success criteria cost, quality, and time.

Enterprises remaining successful on the market have to deal with changing requirements such as new market demands, competition, or regulations and compliance. For this purpose, Business Process Management (BPM) [1] recommends a set of activities for managing and improving business processes continuously. BPM covers all phases of the business process lifecycle in order to provide a holistic process management support. Thus, the phases of design, deployment, and operation / evaluation are supported. In particular, BPM has to assure the efficiency and flexibility of business processes in order to meet the success criteria mentioned above. Without these properties, an enterprise will not be able to stay competitive on the market.

Flexibility and efficiency are strongly dependent on the information technology (IT) used. Although IT is not a guarantor for any competitive advantage, having no IT

will surely cause business failure [2]. Therefore, IT and especially software applications are often considered an enabler for market success.

With regard to BPM, IT should support all phases of the business process lifecycle in a manner that is as holistic as possible. But while design, simulation, and evaluation of a process are performed by business people (using IT, of course), implementation and execution are pure IT-related tasks, as business people usually do not have the required technical knowledge. The resulting separation between the business view and the IT view on business processes significantly complicates – in practice – an “ideal” BPM. The so-called business-IT gap (see line in Figure 1) thus leads to costly and time-consuming human intervention when business process models are to be brought into production or vice versa.

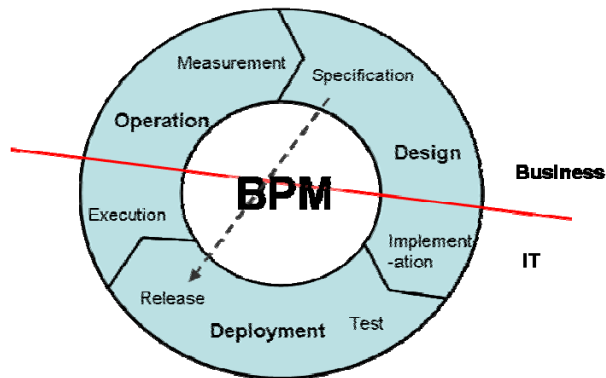


Figure 1. Business Process Lifecycle and the Gap between Business and IT

Because the transition from the design to the deployment phase is the most crucial one with regard to high flexibility, the vision of “business process automation” aims at lowering the effort for developing and releasing IT systems from business process models by automating this transition (see dotted arrow in Figure 1). In this context, the paradigm of Service Oriented Architecture (SOA) [3] is considered as a powerful partner for BPM, providing mechanisms for making process applications and thus enterprises more agile and flexible.

There are several definitions of an SOA. Some references consider SOA as an IT paradigm; others define an SOA as a holistic approach already integrating BPM aspects. For most enterprises, however, SOA is just a technical issue [4], even if its benefits are primarily concerned with business. The basic idea of SOA is the restructuring of IT systems or IT landscapes into loosely coupled, independent services. These services should allow the reuse of existing IT functionality in order to shorten the time between design and implementation when business requirements change.

In general, atomic and molecular services can be distinguished [5]. Atomic services are interfaces to real system functions, while molecular services provide a feature from the business point of view by integrating atomic services or smaller molecules. For instance, creating a new database entry would be an atomic service, while the process for creating a new customer could be a molecular service.

The main challenges in developing such process-driven and service-oriented systems are the refinement and finally the mapping of business processes to the existing service infrastructure. The “naive” and often published assumption that new business process applications can be developed immediately by linking business process models with services is far from reality.

The reasons are obvious: On the one hand, atomic services are close to the technical implementation and thus not applicable for business analysts during business process design. To use them as business process automation units, business processes first have to be refined until a fine-grained technical level is reached on which atomic services can be integrated to provide the required high-level business functionalities (see left side in Figure 2). Due to a lack of semantic information in state-of-the-practice business process models, this stepwise refinement can only be done by humans and is thus costly and time-consuming work. But even the ongoing research endeavor on semantic services and semantic business processes does not seem to be a reliable contribution, as until now, it is completely unclear whether the additional (semantic) specification effort has a justifiable return on investment.

On the other hand, molecular services promise to be easier to apply during business process design. They are much closer to the business perspective and could be linked “as-is” with processes they support (see arrow in the middle of Figure 2). Unfortunately, this is only true in theory. In practice, however, the lack of systematic reuse often results in a mismatch between the business process models and the available services, i.e., the existing services do not immediately fit the requirements elicited during business process design. From a service point of view, this is caused by the fact that the service usage within changing contexts is often neglected, resulting in too specific, poorly reusable services. But also from the business process point of view, effective reuse is often hampered, as the actual capabilities of the service landscape are typically not taken into consideration, leading to the situation that existing services are poorly reused even if they would be appropriate candidates to support business processes.

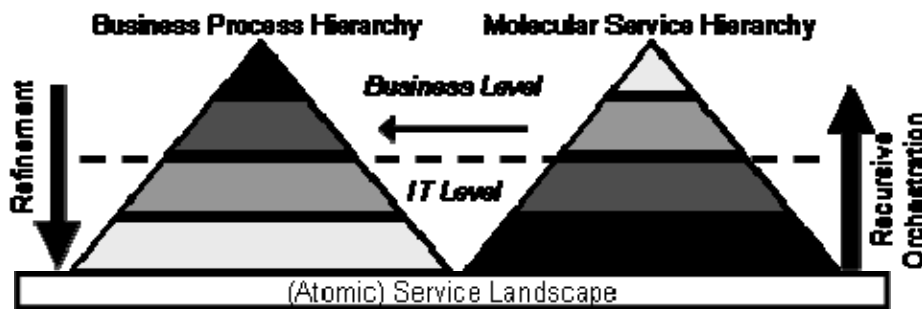


Figure 2. Business Process and Molecular Service Hierarchy

Molecular services are therefore often modified or newly developed when they are to be used in a new process context. This problem is quite similar to the reuse problems in component-based software development (see [6]).

Hence, without a systematic approach, the business-IT gap will not automatically be bridged by using SOA in BPM, i.e., SOA will not automatically provide real

benefits for business agility. Indeed, approaches such as [7] explain how “good” services could be identified based on business process requirements, existing assets, or both. However, identifying services this way does not automatically assure that they are also usable when they are to be applied in another process context. Consequently, the question of how services should be provided in order to be really usable in different business process contexts has – until now – not been answered satisfactorily yet.

To cope with this challenge, and ultimately improve the integration of BPM and SOA, we are convinced that BPM must be closely aligned with the actual SOA landscape of an enterprise in a systematic manner. In our opinion, this can only be achieved by an integrated engineering approach that provides services on the right level of abstraction and reusability, and thus, improves the transition between the process design and deployment phase significantly.

2 Integrating BPM & SOA with Software Engineering Methods

Experience has shown that ad-hoc procedures are often not able to assure repeatable success in developing systems that meet the needs of a customer in a reliable way. This observation also seems to be true when software development is just a small part of a higher-level concept such as BPM.

Thus, to closer align BPM and SOA, we see the necessity for a systematic engineering approach that integrates the activities of the design and deployment phase in BPM with typical steps of a software development methodology. A first step towards such a method could, for instance, be the definition of a BPM/SOA-V-Model that maps the products and phases from traditional software development to the products and phases of BPM. Such a model could help to integrate proven software engineering principles, e.g., abstraction, separation of concerns, traceability, or validation into the holistic BPM lifecycle. In this regard, our meta-model presented in [8] might be a valuable input to support traceability and validation issues.

For improving service reuse, [5] suggests applying product line engineering on the service-oriented paradigm. In general, product line engineering [9, 10] splits the development activities into two phases: family engineering and application engineering (see Figure 3). The purpose of family engineering is to systematically build up a reuse infrastructure from which products of the (product) family can be developed more efficiently. Developing these final products is then performed during application engineering by explicitly considering and integrating the assets of the reuse infrastructure.

The main concept of product line engineering with regard to reuse is the systematic (mostly proactive) definition of all product variants that should be developed based on the family’s reuse infrastructure. For that purpose, common and variable characteristics within the family are planned systematically. For instance, developing a set of different domain-specific project management software could be significantly improved by systematically defining in advance the common functions and the variable, domain-specific functions. For software developing enterprises, product lines have been proven to significantly increase development efficiency.

However, product line approaches cannot be transferred to the BPM/SOA context without adaptation. The most important difference we see in this context is the shift from software developing enterprises to software applying organizations. While software developing enterprises are mostly specialized on a certain set of software products and could therefore define a clear product family (domain scope), software applying enterprises use a multitude of completely different systems such as accounting, product planning, construction, office applications, etc. (called “system of systems”). Consequently, identifying a product family based on a common functionality between these systems is nearly impossible.

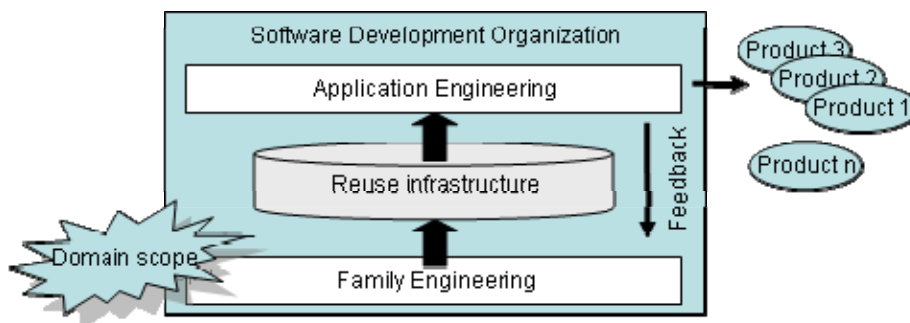


Figure 3. Overview of Product Line Engineering

At this point, the role of BPM and SOA comes into play: Since one goal of SOA in BPM is the integration of different systems in a process-oriented way, using processes as drivers for family engineering in this context seems to be more suitable. In fact, there are approaches such as [11] transferring the idea of product families into process families. The idea is to define all variations and commonalities within a set of business processes and to develop a related reuse infrastructure for developing the process-supporting application. In the context of runtime-adaptive processes that have to change dynamically to meet specific needs in a certain (already known) situation, we consider this to be a useful approach.

However, in the BPM context, we consider the modeling of all process variants and supporting services in advance as not applicable. First, enterprises usually have to react to unforeseen changes. So, modeling all process variants that might occur sometime in the future is just impossible. Second, improving the IT is never an end in itself for most software applying enterprises. Consequently, there must always be quick wins when investment in better IT infrastructures is to be justified.

Nevertheless, we are convinced that an integrated BPM/SOA engineering approach applying the reuse concepts of software product lines is a successful means for achieving closer and faster alignment between process design and deployment within the process lifecycle. Especially the early consideration of existing assets during the development of new (business process) applications, as proposed by product line application engineering, seems to be an appropriate means for assuring that business process requirements and services fit together better. Of course, the right level of granularity and abstraction for the services still has to be found in order to be usable during business process specification.

3 Solution Principles and Open Questions

Based on the observations mentioned above, we are currently developing an approach that aims at this integration. This approach will provide both a top-down part to develop services that are actually reusable in multiple contexts, and a bottom-up part that assures that the existing services can be easily (re)used during business process design.

However, while the principles are already clear, many questions are still open:

Principle 1: Provide services as business elements to increase reuse.

To minimize the gap between business and IT in the BPM lifecycle, we are convinced that (molecular) services must be provided on the level of business elements in order to be understandable and thus usable by business analysts. Furthermore, providing services on this level can also assure that existing assets are taken into consideration already during the early phase of business process design. Thus, the degree of actual service reuse could be increased significantly.

But: What is the “right” level of service abstraction and granularity? Which common elements exist in the business view as well as in the technical view? Which details can be encapsulated in order to allow orchestrating services in a business-like manner without neglecting technical information for execution?

Principle 2: Plan service variants systematically to increase reusability.

Service reusability can be significantly increased through systematic identification, planning, and development of commonalities and variations. Knowing in advance in which context or in which configurations a service will be used helps to develop service variants that are beneficial in different situations.

But: On which level of abstraction and how should variability be considered in order to optimize reuse? The process level seems too coarse-grained and the atomic, technical level too fine-grained. The right level must be somewhere in between, but where? And when the right level is found: How can the business analyst identify and (later) configure the variations in a way that is as business-like as possible?

Principle 3: Build up and improve a highly economical reuse infrastructure.

Reuse infrastructures based on services have to be planned systematically. Adding a multitude of services in an unsystematic manner hampers efficient reuse.

But: How can enterprises build up a product line-like service infrastructure without large ex ante investment? How can quick wins be achieved?

4 Conclusion

In this paper, we discussed the role of SOA in BPM and presented the practical challenges regarding an ideal interplay. We explained that an unsystematic SOA will not entail any business benefits, and what should be done for improvement.

As a software and systems engineering institute, we are currently working on the systematic and integrated engineering of BPM-supporting SOA systems that will

actually minimize the gap between business and IT. One of the main results will be the alignment of business process models with the services actually provided by the service landscape in order to streamline the transition between the design and deployment phase. For this purpose, we plan to adapt established product line approaches to assure a high degree of reuse already during an early phase.

As benefits we expect that significantly less effort will be needed for the refinement of business processes into business process applications. Thus, new business process applications based on already existing services can be developed with less effort compared to approaches that have identified and provided their services in another way. Of course, building such service infrastructures in a systematical way requires more effort than ad-hoc development. Nevertheless, we are convinced that the effort for building a service in a product line oriented way can be amortized after reusing a service twice or thrice. Moreover, as a side-effect, we expect that the service landscape can be kept smaller, leading to fewer costs for maintenance.

References

1. Smith, H., Fingar, P.: Business Process Management – the Third Wave. Meghan-Kiffer Press, 2003
2. Carr, N.: Does IT matter? Information Technology and the Corrosion of Competitive Advantage. Harvard Business School Press, 2004
3. Erl, T.: Service-oriented Architecture: Concepts, Technology, and Design. Prentice Hall, 2005
4. Amadee AG: Studie SOA Check 2007 – Trends im deutschen Markt. <http://www.soa-check.net> (last visit 10 February 2008)
5. Lee, J., Muthig, D., Naab, M.: Identifying and Specifying Reusable Services of Service Centric Systems through Product Line Technology. SOAPL Workshop, SPLC 2007
6. Atkinson, C., Muthig, D.: Enhancing Component Reusability through Product Line Technology. In: Proceedings of the 7th International Conference on Software Reuse: Methods, Techniques, and Tools, LNCS 2319, Springer 2002
7. Wahli, U. et. al: B.: Building SOA Solutions using the Rational SDP. IBM Redbook. <http://www.redbooks.ibm.com> (last visit 10 February 2008)
8. Adam, S., Doerr, J.: On the Notion of Determining System Adequacy by Analyzing the Traceability of Quality. In: Proceedings of Workshops and Doctoral Consortium, Vol. 1, Caise'07, Tapir Academic Press, 2007
9. Bayer, J. et al: PuLSE: A Methodology to Develop Software Product Lines. In: Proceedings of the Fifth Symposium on Software Reusability (SSR99), ACM Press, 1999
10. Clements, P., Northrop, L.: Software Product Lines: Patterns and Practice, Addison Wesley, 2001
11. Bayer, J. et al: Process Family Engineering: Modeling variant-rich processes. <http://publica.fraunhofer.de/starweb/servlet.starweb?path=pub.web&search=N-49329>, 2005