Adaptive Business Process Modeling in the Internet of Services (ABIS)

Monika Weidmann, Falko Koetter, Maximilien Kintz
Fraunhofer Institute for Industrial Engineering IAO
and University of Stuttgart IAT
Stuttgart, Germany
Email: firstname.lastname@iao.fraunhofer.de

Daniel Schleicher, Ralph Mietzner
Institute of Architecture of Application Systems
University of Stuttgart
Stuttgart, Germany
Email: lastname@iaas.uni-stuttgart.de

***Abstract*—In the Internet of Services many companies work together in interorganizational business processes. To enable ad-hoc business interaction it is necessary to align business processes of the business partners, especially in communication processes. These business processes can be partly standardized, but need to be slightly adapted for several similar use cases by the involved companies. This fosters adaptability and reuse for the business partners. We present an approach for adaptive business process modeling in the Internet of Services (ABIS), which allows creation of adaptable process templates. These templates are then used to create variants of processes allowing companies to work together in an interorganizational setting.**

***Keywords*-business process management, adaptive business processes, internet of services, process modeling.**

I. INTRODUCTION

The main idea of the Internet of Services (IoS) is to use the Internet as a medium for offering and selling services [1]. An infrastructure is needed to bring together service consumers and providers to trade services and enable the new business models, where organizations work together to deliver a service to consumers in a previously unknown manner [1] [2]. Business processes, which have been defined and owned by one company in the past, are now used to support a cross-company process flow [2]. In our work with insurance companies we experienced a need for standardized business processes, especially considering interorganizational communication processes, which provide a service, for example a repair service, for insurance customers. Although most companies wish for standardized reference processes to become available, there still persists a need for individualization. Additionally, the requirement to improve products, processes, and customer satisfaction, as well as changing market conditions, regulations, and laws cause a rising need for adaptation of business processes [3]. As in some business processes various partners are included [4], changes to the process affect the interorganizational communication directly. Companies are challenged to comply with different processes to communicate with their respective partners. Thus a need arises for multiple companies to adapt processes together, resulting in sound process models.

The main contribution of this paper is the introduction of concepts, which allow the creation of adaptable interorganizational business processes based on a real use case and real business requirements. The goal is to enable business users with knowledge in process design to create process variants through direct interaction with the process model. In our approach called *Adaptive Business process modeling in the Internet of Services (ABIS)* we define new modeling elements to allow the creation of *process templates* in which *process fragments* may be inserted to create *process variants*. Process templates allow to model standardized and adaptive parts of a process including cross-company dependencies. Process fragments can be modeled independently by different participating companies. We call the process of creating a *variant* from such a template and fragments a *binding*. In contrast to the definition of the process templates and process fragments, binding the process template involves all participating parties. We chose BPMN 2.0 as notation because of its various abstraction levels and its increasing business support [4].

The remainder of this paper is structured as follows: In Section II, we analyze related work dealing with the adaptation of business processes and show the shortcomings we address in this paper. Section III describes a motivational example, which made apparent the need for adaptive business processes within this context. We use this as a continuous example throughout the paper. Section IV gives a detailed description of the introduced diagrams and modeling elements and shows how they can be applied to the use case. The future work is described in Section V before a conclusion is given in Section VI.

II. RELATED WORK

In this section we present related work with the focus on variable business processes. We compare selected approaches in a table according to different criteria before we introduce the variability model we use in our approach.

Previous work has been done on variability in software, for example in [5]. Recently, these concepts have been extended to provide variability in service-oriented systems [6], which combine services in order to provide higher level functionality (see also [7]). To compose services into a service chain, *executable process models* can be used [2]. Variability in process models can be added at *design time*

 ISBN: 978-1-61208-124-3

Table I
COMPARISON OF VARIABILITY MODELING APPROACHES

Criterion	Provop	PESOA	ProCon	MultPers
Integrated variability visualization	-	✓	-	-
Responsibility modeling	-	-	(✓)	✓
Dependency modeling	✓	✓	✓	✓
Integrated dependency visualization	-	✓	-	-
WYSIWYG variant creation	(✓)	(✓)	-	-

and at *runtime*. We do not consider runtime variability as for example described in [8] or [9]. In our use case and project experience it is important to define a process model in advance as a guideline for business partners to be followed during the automatic or manual execution of the process. However, the runtime aspect of adapting business processes should not be disregarded in future work.

In the field of event-driven process chains (EPCs) much work has already been done towards configuration and adaptation [10] [11]. However, due to the complexity of the underlying approaches, and the missing direct interaction of the user with the process model, these approaches do not address the goals of ABIS.

We consider four most relevant concepts related to ABIS. The (1) *Provop* approach allows modeling of variability using so-called *options* on a basic process model, which alter the model by deletion, insertion, or modification operations [12] and has been extended with concepts to guarantee soundness [13]. (2) PESOA uses UML-like constructs for modeling *process families* [14]. A different approach called (3) *process configurator* (ProCon) allows explicit modeling of logic in a tree-based approach enhancing a process with variability [15]. A recent effort called (4) *Multi-Perspectives Variants* (MultPers) defines a data structure to describe a family of process variants [16].

We compare the four approaches using the five criteria in Table I. As we plan to provide a multi-user approach, the modeling of responsibilities is needed. Next, dependencies between decisions are to be modeled and visualized within the process models for reasons of usability. Finally, we consider it important to provide a What You See Is What You Get (WYSIWYG) approach for the creation of variants, as in our experience many business users are already aware of process models [4]. However, a high usability is very important for business process management tools in general [4].

Different ways for modeling variable process models do not refer to one configurable process model, but the specification of single process fragments, which then can be glued together in order to reuse concepts and create different process models [17] and [18]. Although we will use concepts of separate process building blocks, the need of the business users is a standardized process template to start with.

In our ABIS method we use the approach of Mietzner et al. in [19] and [7] for generic variability modeling in XML files. Here, the variability is added to XML files without altering the original file. This enables a separation of the process model (BPMN 2.0 XML file) from the variability (XML file) for storage and interchange of process models without extension of the BPMN 2.0 metamodel. The approach allows the definition of *variability points* and *alternatives* [19]. The alternatives can be explicitly defined, specified by the user, or be left empty. *Dependencies* allow to enforce a binding order of variability points, whereas *enabling conditions* can limit the choices for alternatives for variability points depending on previous choices. Based on this approach we can enable a multi-user derivation process for ABIS in future work, separate the variability from the process model, allow the modeling of complex dependencies, and provide tool support for the creation of variants supporting the user with automatic choices if only one alternative is left.

III. Real World Use Case

In this section we present a simplified real world use case we came across on our work in the openXchange project (www.openxchange-project.de), dealing with creating a service network of small and medium sized enterprises to handle property damage claims. In active claims management, insurance companies often involve external partners for various tasks like creating a survey report or removing the damage. In the course of process standardization we came across individualization needs, as the companies want to:

- Work together with partners through IT-supported processes
- Use standardized predefined processes supporting their business needs
- Have individualization options for certain aspects of the business process
- Have a sound process model to communicate with their partners

We modeled a part of the active claims management process in Figure 1, where one simplified interorganizational business process is presented. The lanes in the process model were omitted to save space. At the top you see the customer's process, which in our example is an insurance company in need of a building repair service. Below it you can see the external partner or contractor. We reduced the detailed commissioning process to the following tasks: preparing, sending, and receiving requests and confirmations, and the handling of reports.

The adaptable parts for creating other process variants are highlighted in grey color and described using the text annotations (1), (2), and (3):

- At (1) it shall be possible to choose if intermediate reports are used. If they are used, the process looks like

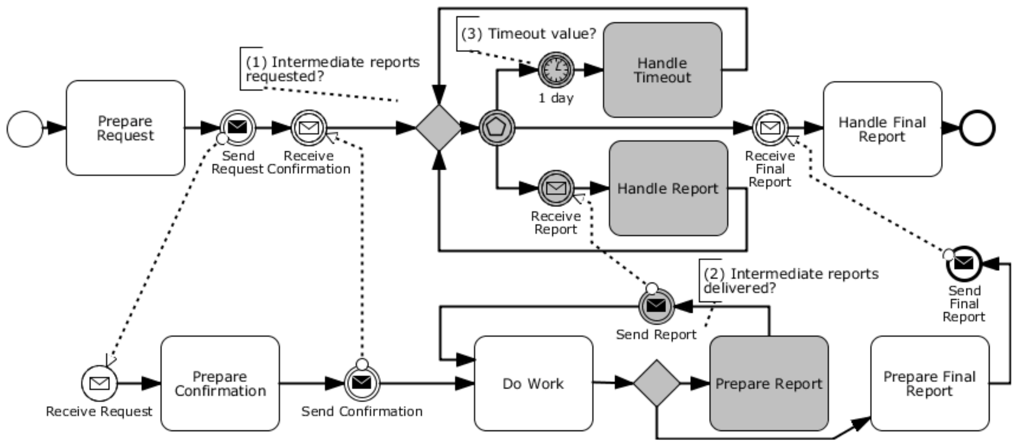

Figure 1. Real world use case: Active claims management

in Figure 1. Otherwise, the highlighted parts at (1) are omitted.

- At (2) it also shall be possible to configure the usage of intermediate reports. As it does not make sense to create reports which are not sent, (2) is considered *dependant* on the choice made at (1).
- At (3) the attribute *timeout* shall be configurable with the values '1 day' or '2 days'.

We will later show how these requirements are handled. Other possible use cases include the configuration of a B2B-online shop, the communication processes in case management (healthcare), or the configuration of air line catering processes.

IV. Modeling Variability in Business Processes with ABIS

In this section we show the scope of our variability modeling as a basis for the introduction of the ABIS concepts, which we then apply to the use case introduced above.

A. Variability scope

The list of BPMN 2.0 elements considered as variable so far is based on the BPMN 2.0 specification [20].

Event - There are multiple event types (error, escalation, etc.) defined in BPMN 2.0. They differ in the time dimension (interrupting vs. non-interrupting) as well as the flow dimension (starting a process, intermediate, ending a process). Changing these semantics via variability is error-prone from the business process designer view, as each of these events has to be used in a specific context. The resulting dependencies would add enormous complexity to a process model. We therefore do not consider the event types, time dimensions, and flow dimensions as variable to an event.

Activity - An activity can be a task or a sub-process (reusable and non-reusable). The choice of activities and sub-processes is one important element of variability in BPMN process models and is enabled by ABIS. Variability of activity type will not be permitted explicitly, but may be modeled by offering a choice between different activities.

Gateway - Gateways allow steering the process flow. Therefore, the condition-expression of the gateway and the options are important places to add variability. Variability of the semantics of the gateway itself (exclusive, parallel, etc.) is difficult to understand for human process designers. The reason is that especially business users without technical knowledge find it hard to model detailed execution semantics [21], which are implied by several BPMN artifacts. Therefore, gateway types are not variable in ABIS.

Sequence and Message Flow - Variable sequence and message flow is a requirement for variable activities. If an activity is added to the process, the message flow will have to be adapted to connect the activity to the process correctly. Variable sequence and message flows are enabled by ABIS.

Pools and Lanes - The creation of variable pools and lanes implies a fundamental change of the business goal. Hence, it is not considered here.

Others - Other constructs are for example groups, text annotations, etc. These will not be considered specifically in this context but might be added in the future.

Attributes are considered separately. All mentioned elements have several attributes, for example name and documentation. From the use case, typically three types of attributes can be considered variable: gateway conditions, thresholds on sequence flows, and implementation details. As some attributes are defined for all elements, others only for a subset, it is difficult to restrict the attributes which can be considered variable. We generally consider all attributes to be variable.

B. Variability modeling elements in ABIS

Considering the variability scope defined in the previous subsection, the addition of variability can be done within the constructs of the chosen modeling language or by extending the modeling language with new constructs. Using extensions,

 ISBN: 978-1-61208-124-3

the visibility of variability within the process is higher than using native language elements. For tool support, it is also easier to use explicit variability modeling than implicit semantics. Because of these reasons, we introduce new constructs with the following goals:

- Introduce *as few and as simple constructs* as possible to ensure high usability and allow easy comprehension
- Provide an *additive approach*, as deletion is more expensive from a user's point of view [22]
- Give a *graphical notation* which is not easily confusable with existing notations (considering BPMN 2.0)
- Enable the *scope*, as described in the previous section

To reach these goals, the following two *diagram types* are introduced:

Process template Within the process template, all BPMN elements are allowed. Additionally, elements for modeling variability are defined: *variable region*, *variable link* and *variable attribute*, which are then bound, resulting in a valid BPMN process variant.

Process fragments A process fragment in ABIS is a construct similar to a subprocess, which can be inserted into a process template at variable regions during *binding*. Process fragments in ABIS differ from subprocesses in two aspects. For one thing, they can define additional sequence and message flows to other elements of the process template. For another thing, they are inserted in the same scope as the variable region they replace. Process fragments may contain variable regions themselves. A process fragment is modeled with specialized start and end events called *fragment start and end links*. We do not allow deletion of process elements.

Process fragments could be either modeled within a process template or separately. We choose to model them separately, in order to fulfill the requirements of a distributed environment considering fragment repositories as has been researched in [17] and [18].

Additionally to these two diagram types, the following new *modeling elements* are introduced:

Variable region - A variable region is a new element for modeling similar to an activity. A variable region differs from an activity as follows: it has exactly one incoming and one outgoing sequence flow. A variable region is a placeholder for process fragments, which are inserted at this position in the current process template or process fragment.

Variable link - A variable link is a new element similar to the BPMN 2.0 throwing link event. In contrast to the throwing link event, a variable link is used to model the target of a message or sequence flow to show to which element this flow will be directly connected, that means without an additional catching link event. During the binding process, the incoming sequence or message flow of the variable link is connected to lead from its source to the specified *target* of the variable link. The link itself is then discarded.

Fragment start and end link - Fragment start and end links are used to denote where the incoming and outgoing message flow needs to be connected in the processes template. Exactly one fragment start and end link have to be included in each process fragment.

Table II
SCOPE - BPMN 2.0 ELEMENTS WITH VARIABILITY

BPMN Element	Can have variable attributes	Can be used in process fragments	ABIS variability modeling element
Event	✓	✓	
Activity	✓	✓	
Gateway	✓	✓	
Sequence and message flow	✓	✓	*variable link*
Pools and lanes	✓		
Others	✓	✓	

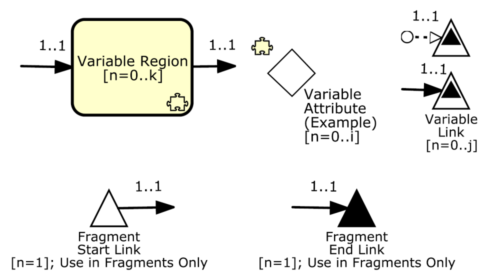

Figure 2. BPMN elements introduced

Variable attributes - As all attributes may be variable, a separate description of the variability is needed.

Table II shows an overview of how variability is added to the BPMN 2.0 elements considered in the scope. The variability can be either added through variable attributes, by usage of the element in process fragments (and variable regions), or by using the explicit ABIS modeling elements.

We introduce a graphical notation for the new elements in Figure 2. The cardinality restrictions are shown in UML notation. The elements in the top row may be used as often as needed in process templates and process fragments. The elements in the bottom row are only allowed in process fragments. The notation of the variable attribute of the gate is an example, as all BPMN elements may have variable attributes. The reason for using puzzle pieces and triangles is that these shapes do not have semantics in BPMN.

The resulting BPMN model for process templates and variable regions will be stored as plain BPMN 2.0 with an additional XML file describing the variability points. Therefore, we do not define a BPMN 2.0 extension.

In the following section we will use the new constructs to show how the continuous example is adapted.

C. Application of ABIS to a real world use case

We have introduced an example for a commissioning process between a customer (insurance company) and a

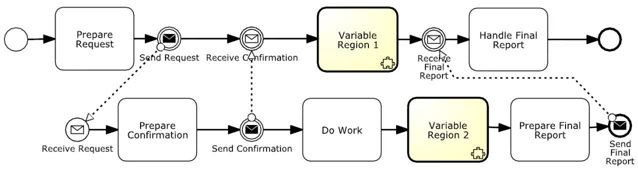

Figure 3. Process template for use case

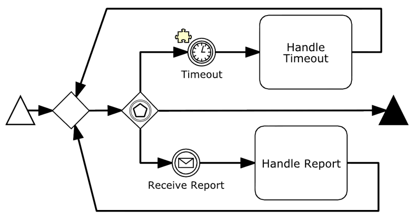

Figure 4. Process fragment A - option 1 for variable region 1

contractor (company offering repair services in buildings) in Section III in Figure 1. We take the following steps to model the variability:

- Identify the variable elements in the use case (already described in Section III)
- Create the process template for the use case (next step)
- Define the process fragments for the use case (see Figures 4 to 6)
- Model the structure of the use case (see Figure 7)

We will elaborate on a detailed methodology in future work.

Using the modeling elements and diagram types introduced before, we can replace (1) and (2) in Figure 1 by variable regions, as shown in Figure 3. Notice that the variable regions are marked with a puzzle piece. The variable timeout attribute has been placed in variable region 1. The variable regions have exactly one incoming and outgoing sequence flow.

The process fragment A in Figure 4 contains the first option for variable region 1. Here, the handling of the intermediate reports is modeled. Additionally, the timeout is variable and can be set to the different values a or b ('1 day' or '2 days'). Notice that all process fragments have exactly one fragment start and end link. Alternatively no intermediate reports are expected. Therefore, process fragment B - an empty fragment - is used (see Figure 5).

For the second variable region the first alternative is shown in Figure 6. It contains the preparation and the sending of intermediate reports. Notice the variable message flow to *Receive Report*. If no intermediate reports are requested, the empty fragment (see Figure 5) must also be used for variable region 2.

In order to bind the process template, the process fragments are inserted into the process template replacing the variable

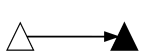

Figure 5. Process fragment B - option 2 for variable regions 1 and 2

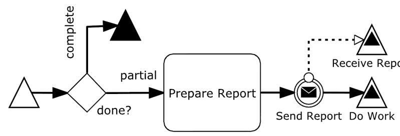

Figure 6. Process fragment C - option 1 for variable region 2

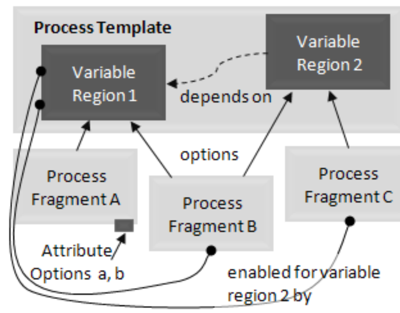

Figure 7. Structure of use case example

regions, resulting - when no variable elements are left - in a process variant. The sequence flows connecting to a variable region are connected to the respective start and end links. When the fragment has been inserted, all variable parts of the process fragment have to be bound. An example for this is the timer variable. The timeout value must be chosen after the insertion of process fragment A. For the example above, choosing process fragments A and C and setting the timeout to '1 day' would result in the initial process model (Figure 1).

In Figure 7 the *structure* of the example use case is shown. The two variable regions are marked in dark grey. The process fragments A, B, and C are the options for the variable regions as described above. Hereby, A and B can be *bound* to variable region 1, and B and C can be bound to variable region 2. Finally, the process fragment A also contains a variable attribute (the timeout), with the options a and b.

As variable region 2 is directly *dependant* on variable region 1, only one choice has to be made for variable region 1, which then directly affects variable region 2. The dependency between the variable regions 1 and 2 is indicated by a dotted line, showing that variable region 1 needs to be bound before variable region 2, as the variable link to *Receive Report* in variable region 2 (see Figure 6) would have no target otherwise. Additionally we use *enabling conditions* to limit the choices for variable region 2 according to the choice made in variable region 1. For formal definitions of the applied variability model see [7]. We will describe a detailed concept of how to apply the variability model to our modeling approach ABIS in future work.

V. Future Work

In future work we will develop a prototype which supports the creation of process templates, process fragments, and binding alternatives to create a process variant. Additionally we will describe the architecture of the resulting solution, the process of filling in the variability points, as well as introduce an algorithm for handling of dependencies and automatic choice of alternatives based on previous decisions. We will validate our approach using a set of real world process models. In the course of this work we might consider defining additional variable BPMN 2.0 elements if needed. Further on we will work on a methodology with corresponding role models to support the application of our concept. We will consider combining the adaptable business processes with concepts of modifying runtime business processes like for example in [23]. However, these efforts strongly depend on the future development of BPMN 2.0 runtime engines. Finally we will analyze the possibility to enhance our approach with compliance-specific features in order to support design of compliant business processes as for example in [24].

VI. Conclusion

In this paper we introduced a method for adaptive business process modeling in the Internet of Services called ABIS. The goal of our approach is to enable business users to create their own process variants in an interorganizational setting based on standardized variable process models. To accomplish this, we introduce two new diagram types with additional modeling elements in BPMN 2.0. The business users may *separately* model parts of the process using *process fragments*, the first new diagram type. The second diagram type is the *process template*. Process fragments are inserted into process templates in order to create an interorganizational process variant. While fragments may be modeled independently, creating the process variant involves all participating parties. The presented concept was applied to a real world use case.

Acknowledgment

The work published in this article was partially funded by the openXchange project of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ09011 and by the MASTER project (www.master-fp7.eu) under the EU 7th Research Framework Programme Information and Communication Technologies Objective (FP7-216917).

References

[1] J. Cardoso, K. Voigt, and M. Winkler, “Service engineering for the internet of services,” in *Enterprise Information Systems*, ser. Lecture Notes in Business Information Processing, W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, J. Filipe, and J. Cordeiro, Eds. Springer, 2009, vol. 19, pp. 15–27.

[2] A. P. Barros and M. Dumas, “The Rise of Web Service Ecosystems,” *IT Professional*, vol. 8, no. 5, pp. 31–37, Sep. 2006.

[3] C. Wolf and P. Hermon, “The State of Business Process Management 2010,” *BPTrends Reports (February 2010)*, 2010, last accessed 13.01.2011. [Online]. Available: www.bptrends.com/surveys_landing.cfm

[4] S. Patig, V. Casanova-brito, B. Vögeli, C. Bern, and B. Voegeli, “IT Requirements of Business Process Management in Practice An Empirical Study,” *8th International Conference on Business Process Management (BPM’2010)*, pp. 13–28, 2010.

[5] I. Jacobson, M. Griss, and P. Jonsson, *Software Reuse: Architecture, Process and Organization for Business Success*. Addison-Wesley Professional, 1997.

[6] S. H. Chang and S. D. Kim, “A variability modeling method for adaptable services in service-oriented computing,” *11th International Software Product Line Conference (SPLC 2007)*, no. 10557, pp. 261–268, Sep. 2007.

[7] R. Mietzner, “A Method and Implementation to Define and Provision Variable Composite Applications , and its Usage in Cloud Computing,” Ph.D. dissertation, Universität Stuttgart, 2010.

[8] M. Koning, C.-a. Sun, M. Sinnema, and P. Avgeriou, “VxBPEL: Supporting variability for Web services in BPEL,” *Inf. Softw. Technol.*, vol. 51, no. 2, pp. 258–269, 2009.

[9] C. Dorn, T. Burkhart, D. Werth, and S. Dustdar, “Self-adjusting Recommendations for People-Driven Ad-Hoc Processes,” *8th International Conference on Business Process Management (BPM’2010)*, pp. 327–342, 2010.

[10] M. Rosemann and W. M. P. V. D. Aalst, “A configurable reference modelling language,” *Information Systems*, vol. 32, no. 1, pp. 1–23, 2007.

[11] H. Reijers, R. Mans, and R. Vandertoorn, “Improved model management with aggregated business process models,” *Data & Knowledge Engineering*, vol. 68, no. 2, pp. 221–243, 2009.

[12] “Modellierung und Darstellung von Prozessvarianten in Provop,” in *Modellierung 2008 Conference*, ser. LNI, T. Kühne, W. Reisig, and F. Steimann, Eds., vol. 127. GI, 2008, pp. 41–56.

[13] A. Hallerbach, T. Bauer, and M. Reichert, “Guaranteeing Soundness of Configurable Process Variants in Provop,” in *IEEE International Conference on E-Commerce Technology*, 2009, pp. 98–105.

[14] A. Schnieders and F. Puhlmann, “Variability Mechanisms in E-Business Process Families,” in *Proc. International Conference on Business Information Systems (BIS 2006)*, 2006, pp. 583–601.

[15] A. Werner and H. Müller, “Geschäftsprozesskonfigurator - Softwaregestützte Geschäftsprozessberatung,” *ERP Management*, vol. 5, no. 2, pp. 25–28, 2009.

[16] S. Meerkamm, “Configuration of Multi-Perspectives Variants,” in *1st International Workshop on Reuse in Business Process Management (rBPM10)*, Hoboken, NJ, USA, 2010.

[17] H. Eberle, T. Unger, and F. Leymann, “Process Fragments,” in *On the Move to Meaningful Internet Systems: OTM 2009*, ser. Lecture Notes in Computer Science, R. Meersman, T. Dillon, and P. Herrero, Eds. Springer Berlin / Heidelberg, 2009, vol. 5870, pp. 398–405.

[18] D. Schumm, D. Karastoyanova, F. Leymann, and S. Strauch, “Fragmento: Advanced Process Fragment Library,” in *Proceedings of the 19th International Conference on Information Systems Development (ISD’10)*. Prague, Czech Republic: Springer, 2010, pp. 1–12.

[19] R. Mietzner, A. Metzger, F. Leymann, and K. Pohl, “Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications,” in *PESOS ’09: Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 18–25.

[20] Object Management Group (OMG), “Business Process Model and Notation (BPMN) Version 2.0,” 2009, last accessed 13.01.2011. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/

[21] B. Silver, *BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0*. Cody-Cassidy Press, 2009.

[22] O. Holschke, “Impact of Granularity on Adjustment Behavior in Adaptive Reuse of Business Process Models,” in *Business Process Management*, 2010, pp. 112–127.

[23] K. Vidačković, N. Weiner, H. Kett, and T. Renner, “Towards business-oriented monitoring and adaptation of distributed service-based applications from a process owners viewpoint,” in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*, ser. Lecture Notes in Computer Science, A. Dan, F. Gittler, and F. Toumani, Eds., vol. 6275. Springer Berlin / Heidelberg, 2010, pp. 385–394.

[24] D. Schleicher, T. Anstett, F. Leymann, and R. Mietzner, “Maintaining compliance in customizable process models,” *On the Move to Meaningful Internet Systems: OTM 2009*, pp. 60–75, 2009.