On the Support of Workflow Activity Patterns in Process Modeling Tools: Purpose and Requirements [*]

Lucinéia Heloisa Thom
Institute of Database and
Information Systems
Ulm University
James-Frank-Ring, 89069
Ulm, Germany
lucineia.thom@uni-
ulm.de

Manfred Reichert
Institute of Database and
Information Systems
Ulm University
James-Frank-Ring, 89069
Ulm, Germany
manfred.reichert@uni-
ulm.de

Cirano Iochpe
Institute of Informatics
Federal University of Rio
Grande do Sul
Av. Bento Goncalves
Porto Alegre, Brazil
ciochpe@inf.ufrgs.br

ABSTRACT

Patterns increase reuse of existent knowledge (e.g., design solutions, source code) within organizations and help to achieve consistency between applications. Patterns for process design have received considerable attention by both business analysts and researchers. Several patterns categories have been proposed including patterns for control and data flow, resources, process change and exception handling. Workflow activity patterns, which are within business processes (e.g., approval, task execution request), however, have not been explored so far. Related to this problem we have proposed a set of workflow activity patterns in the ProWAP project. Each pattern represents a recurrent business function frequently found in business processes. The completeness and existence of these patterns is evaluated through an extensive analysis of real process models. We further investigated the frequency of co-occurring activity patterns. In this paper we discuss how to implement activity patterns in Business Process Management based technologies. In particular we describe the main purposes of our BPM activity patterns -based Tool as well as basic requirements related to its development. In addition we provide a discussion on the representation of activity patterns in BPMN 1.2 in comparation to this representation in UML 2.0.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Implementation—*Purpose, requirements*

General Terms

Workflow activity patterns

Keywords

workflow activity patterns, business process modeling, implementation requirements, BPMN

1. INTRODUCTION

Process models can be assembled out of a set of recurrent business functions (e.g., task execution request, approval, notification), of which each has a specific semantics. In practice, corresponding business functions are often implemented from scratch for each business process model. Consequently, process design becomes inefficient, complex and time-consuming when dealing with large process model collections [19].

In order to deal with this problem we relate such business functions to a set of well-defined workflow activity patterns: *Request for Activity Execution with/without Answer*, *Approval*, *Notification*, *Decision-making*, and *Information Request* [12, 15]. This pattern set is closer to the vocabulary and abstraction level at which business processes are usually described by domain experts. This fosters pattern reuse when modeling business processes and therefore contributes to more standardized and better comparable business process models [15]. Generally, multiple activity patterns can be composed in a process model using workflow patterns like Sequence, AND-Split, AND-Join or XOR-Split. An empirical study, in which we analyzed 239 real-world process models confirms the existence of the seven activity patterns [15, 14]. We learned that many of the analyzed process models can be designed based on the investigated patterns; i.e., the set of identified activity patterns is necessary as well as sufficient to design the 239 process models, at least at a certain level of granularity.

We also investigated the frequency of co-occurring activity patterns in real world process models [5]. We implemented a process model mining tool to be used for identifying the activity patterns co-occurrences. Our miner allows analyzing process models instead of event logs as proposed in literature. This can be considered as a very important functionality to automatically identify activity patterns co-occurrences (e.g., the pattern pair APPROVAL → NOTIFICATION) in real-world process models.

The identified activity patterns are independent of a concrete process modeling language; i.e., they can be integrated into any process modeling tool. To achieve a precise semantics we have formalized activity patterns using π-calculus [13]. A process model specified in π-calculus can express the dynamic behavior of the process, thus making it possible to verify formal properties of the model like soundness (e.g., absence of deadlocks and livelocks) and model equivalence [6].

Though it is known that pattern utilization helps to promote good design practices and to improve the quality and performance of process modeling, contemporary Business Process Management as well as Modeling tools like Intalio [4], ARIS Toolset [10] and WBI Modeler [8] do not support process designers in defining, querying and reusing activity patterns as building blocks for business process modeling [3]. In this paper we present preliminary results concerning the development of a business process modeling tool (BPM activity patterns -based tool for short), which fosters the modeling of business processes based on the reuse of activity patterns. In particular, we discuss main purpose concerning the development of the tool as well as core implementation requirements. We provide an evaluation of the capabilities of the BPMN and their strengths and weaknesses when being utilizing for mapping workflow activity patterns. BPMN is becoming a standardized graphical notation for representing business process models. The current version, i.e., BPMN 1.2 also solve ambiguities and inconsistencies presented in BPMN 1.2. Mapping the activity patterns to BPMN will facilitate to detect their occurrence in process models. Moreover, as BPMN is becoming a well-known notation to business analysts it will become easier to perform design experiments in order to empirically validate whether or not the activity patterns contribute to reduce real efforts when designing process models.

The remainder of this paper is organized as follows: In Section 2 we discuss related work. Section 3 briefly reviews the workflow activity patterns. Section 4 describes the main purpose regarding the development of our BPM activity patterns -based tool as well as underlying requirements. In this section we also provide a short introduction to the BPMN notation and report on the representation of the activity patterns in BPMN. Section 5 concludes with a summary and an outlook on future research.

2. RELATED WORK

Recently, a variety of workflow patterns was suggested for capturing different aspects in Process-Aware Information Systems including control and data flow, resources, process change, and exception handling patterns [18, 20, 19]. In particular, the service interaction patterns proposed in [1] are of special interest for our approach. These patterns allow for web services interactions, pertaining to choreography and orchestration, to be benchmarked against abstracted forms of representative scenarios. As example consider the Send and the Send/Receive patterns. We can use the service interaction patterns to implement the activity patterns. However, so far there has been no mapping of activity patterns onto process (meta) models and process modeling tools respectively.

Concerning workflow patterns, tool support is provided by YAWL [17], which uses extended workflow nets as building blocks for workflow specifications. Workflow nets involved in a workflow specification can be connected to each other by composite tasks. However, this tool does not support patterns related to activities of a business process (e.g., task execution request, approval).

The PICTURE approach proposes a set of 37 domain specific process building blocks [2]. More precisely, these building blocks are used by end users in public administrations to capture the process landscape and are also specific to this domain. Finally, ProCycle presents an approach implementing process change patterns in ADEPT2 [21].

Related to the evaluation of BPMN in respect to workflow patterns representation, an interesting approach is proposed in [23]. The approach reports on particular limitations of BPMN with respect to the representation of workflow patterns (e.g., control and data flow, resources) (in comparation with UML 2.0). According to the authors, BPMN is slightly stronger when it comes to capture the Interleaved parallel routing and the Syncronizing merge patterns and slightly weaker in its support for the Discriminator pattern.

3. WORKFLOW ACTIVITY PATTERNS

A workflow activity pattern (activity pattern, for short) refers to the description of a recurrent business function as it can be frequently found in business processes. Table 1 shows the semantics of the activity patterns used in this paper. The complete set of activity patterns is described in [12, 15]. Figure 1 gives an overview of the seven activity patterns we identified.

We have characterized each activity pattern by giving a description, an example illustrating the context in which the pattern can be applied, a problem motivating its use, specific issues, design choices (patterns variants) which allow for pattern parameterization keeping the number of distinct patterns manageable, related patterns, and implementation details. The design choices were defined based on the process models we analyzed. For example, we define three variants of the approval pattern, namely single approval (i.e., approval is required from exactly one organizational role), iterative approval (i.e., sequential approval is required from a list of reviewers) and concurrent approval (i.e., approval is required from a list of reviewers simultaneously). Information about the variants we defined for the other seven patterns can be found in [15]. Note that Table 1 only shows selected pattern variants, but does not contain all possible parameterizations. Reason is that most of these properties are out of the scope of this paper. Finally, we informally summarized pattern semantics based on UML activity diagrams [15].

4. IMPLEMENTING ACTIVITY PATTERNS IN BPM TOOLS

This section reviews basic concepts of the BPM activity patterns -based tool. It enables the design of process models based on the reuse of activity patterns. In principle, the basic concepts behind this tool can be added as extensions

Table 1: Selected variants of activity patterns representing business functions

WAP - Name	Description
WAP 1: *Approval*	An object (e.g., a business document) has to be approved by one or more organizational roles.
WAP 2: *Question-Response*	A question which emerges during process enactment has to be answered. This pattern allows to formulate the question, to identify the organizational role(s) who shall answer it, to send the question to the respective role(s), and to wait for the response(s) (*single-question-response*).
WAP 3: *Unidirectional Performative*	A sender requests the execution of a particular task from another process participant. The sender continues process execution immediately after having sent the request for performing the activity.
WAP 4: *Bi-directional Performative*	A sender requests the execution of a particular task from another process actor. The sender waits until this actor notifies him that the requested task has been performed.
WAP 5: *Notification*	The status or result of an activity execution is communicated to one or more process participants.
WAP 6: *Information Request*	An actor requests certain information from a process participant. He continues process execution after having received the requested information.
WAP 7: *Decision*	This pattern can be used to represent a decision activity in the flow with different connectors to subsequent execution branches. Exactly those branches will be selected for execution whose transition conditions evaluate to true.

to existing BPM tools as well; e.g., Intalio [4], Aris [10], and ADEPT2 Process Composer [9].

4.1 Purpose of the Implementation

Core functionalities of our BPM activity patterns -based tool are as follows:

- *Assisting users in designing process models*: First, the process designer chooses a workflow activity pattern and a corresponding design choice from a palette of patterns to configure a concrete pattern variant. Following this, the tool recommends the most suitable activity patterns to be used together with the activity pattern directly applied before. In addition, it informs the user about how frequently each pair of activity patterns (i.e., the previously applied activity pattern plus the recommended one) was used in earlier modeling (cf. Figure 1). This inference engine is developed based on the analysis results presented in [5]. Optionally, the designer can simply choose the next pattern to be included in the model without considering the tool recommendations. We also consider to provide mechanisms, which allow to apply the activity patterns in combination with elementary design elements from BPMN as well as existing patterns (e.g., workflow patterns, change patterns).

- *Facilitate the development of experiments to empirically validate whether or not the activity patterns contribute to reduce efforts for designing process models*: We plan to use our tool for conducting a series of experiments in which we compare process modeling with and without activity pattern support. The goal of these experiments is to verify the acceptance of the activity patterns by designers, to check whether the tool contributes to more correct models, and to figure out whether it reduces design efforts. To do so we have two options: a) the first one refers to the implementation of the activity patterns as extension of some existing BPM tool. This option could facilitate the development of our extension. Reason is that we can focus on the implementation of the patterns and in the metrics we will use to analyze the results of the experiments; b) the second option concerns the development of an independent editor based on the activity
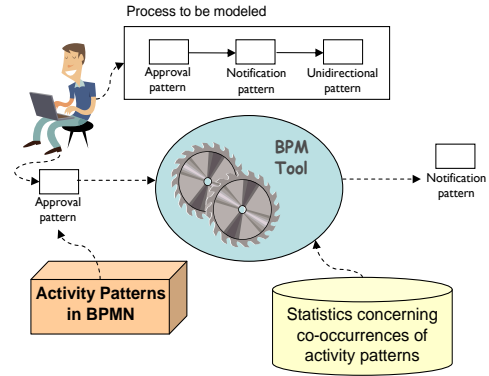

Figure 1: Architecture of a BPM tool based on activity patterns reuse

patterns reuse from scratch. This option may demand more human effort, hours of development and costs. Accordingly, we consider to follow the first option and implement the patterns in some existent tool.

4.2 Implementation Requirements

Using time to define requirements before starting the project avoids wasting time on inappropriate tools and limits completely inappropriate decisions [16]. This section highlights core requirements regarding the implementation of an extension for workflow activity patterns in existing BPM tool. In particular, we introduce four requirements: representation of the workflow activity patterns in BPMN; extensible graphical interface; WS-BPEL (Web Services Business Process Execution Language) generation; and inference engine support.

- *Representing workflow activity patterns in BPMN*: Several contemporary BPM tools have been developed based on BPMN. The notation presents a large number of elements making it possible to capture different design problems. Initially, we have represented our activity patterns with UML 2.0 (see [15]). This provides a clear idea of the semantics of each pattern. However, when implementing the activity patterns in existing BPM tools, it is fundamental to represent them in BPMN. This representation may facilitate the adoption of the activity patterns by BPMN design tools as well as the development of design experiments. Moreover, if we have the patterns in BPMN and the tool supports BPEL output from BPMN models, we already have BPEL output implemented for the pattern-designed processes. Finally, in Section 5 we discuss the mapping of the activity patterns to BPMN and briefly compare it with our earlier experience with UML 2.0.

- *Extensible Graphical Interface*: each pattern included in a process design needs to be configured according to specific parameters. The GUI of the BPM Tool needs to be extensible so that we can provide customized dialogs and other GUI elements for the entering of these pattern parameters.
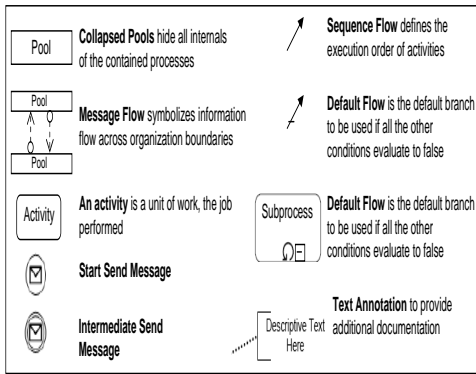
Figure 2: Summary of the BPMN Notation 1.2

Figure 3: WAP1 - Single Approval Pattern in BPMN

- *WS-BPEL generation*: in order to execute the designed process, the process model needs to be exported in WS-BPEL.

- *Inference Engine Support*: The BPM Tool needs an extension mechanism like a plug-in architecture to support the inclusion of the inference engine module. This inference engine has to be called after an activity pattern has been configured. The call identifies possible succeeding patterns as explained in [5].

5. ACTIVITY PATTERNS IN BPMN

This section reports on the representation of the activity patterns in BPMN 1.2. Due to space limitations we present 3 pattern samples and specific variants: WAP1 Approval (*Single approval*), WAP3 Unidirectional Performative (*Single request-response*), and Notification (*Single notification*). The complete set of activity patterns in BPMN can be found in [11]. We analyze the capabilities and weaknesses of BPMN 1.2 in respect to the mapping and compare it with UML 2.0 mapping we presented in [15]. To design the activity patterns with BPMN 1.2 we have used the Intalio tool which is an open-source tool supporting most BPMN elements.

5.1 Introduction to BPMN

The Business Process Modeling Notation (BPMN) was developed under the coordination of the Object Management Group [22]. The primary goal of BPMN is to provide a notation that is easily understandable by all business users, from the business analysts who create the initial drafts of the processes, to the technical developers responsible for implementing the process-aware information systems that will perform those processes, and finally, to the business people who will manage and monitor those processes [22]. Currently, several tools are realized on bases of BPMN (e.g., Intalio, Oryx Editor, IDS-Scheer Aris). Another, but not less important goal, is to ensure that XML languages designed for the execution of business processes, such as WS-BPEL can be visualized with a business-oriented notation (cf. [7]).

The elements from BPMN 1.2 used for the mapping of the activity patterns are summarized in Figure 2. For a description of BPMN the reader is referred to [7].
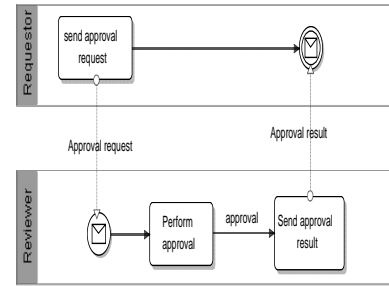
5.2 Mapping Activity Patterns to BPMN

In this section we introduce examples of workflow activity patterns modeled with BPMN. First we give a short description of the behavior of the activity pattern in BPMN 1.2. Following this, we discuss the representation of the activity pattern (i.e., related variant) in BPMN 1.2. We also indicate which service interaction pattern can be used to implement the activity pattern. Finally, we briefly compare the representation of the activity patterns in BPMN with the representation in Activity Diagrams from UML2.0.

WAP1 Approval: Single Approval

In the *single approval* variant of the approval pattern a requestor sends an approval request to exactly one reviewer. This reviewer then performs the revision either resulting in approval or rejection [15]. We have illustrated the participants involved in this pattern using two *pools* called Requestor and Reviewer. First, an approval request is send to the reviewer through a *send activity* (activity type from BPMN). The receiver gets a message from the requestor asking for an approval *start message event*. Following this, he performs the approval activity and the result is sent to the requestor instance of a *send activity*. The requestor receives the result as *intermediate message event* (cf. Figure 3). As one can see, this pattern is not directly supported by only one BPMN element, but it is build with multiple BPMN elements. In UML 2.0 we have illustrated the single approval using two partitions and event signals exchanges between the two participants involved. As BPMN supports the concept of message exchange between process participants, it also eases the illustration of the message exchange concept required by the single approval pattern (see Table 2 for a comparative analysis). In both notations the single approval pattern can be implemented as a *send/receive pattern* [1].

WAP3 Unidirectional: Multi-request

In the *Multi-request* variant of the unidirectional performative pattern a requestor sends an activity execution request to multiple receivers simultaneously and continues process execution afterwards; i.e., without waiting for any response [15]. We have illustrated the participants involved in this pattern using two *pools* called Sender and Receiver. In order to illustrate the multiple instance activity we have used the *looping sub-process*. We set the parameter of the loop as *for each*. In Intalio the multiple instance activity is not
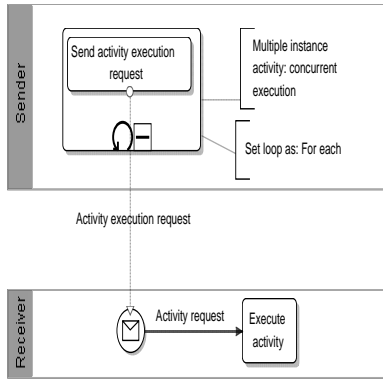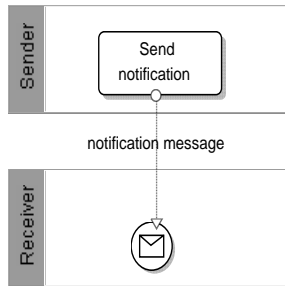
Figure 4: WAP3 - Multi-request Pattern in BPMN

Figure 5: WAP5 - Single Notification Pattern in BPMN

Table 2: Mapping of Workflow Activity Patterns in ULM 2.0 versus BPMN 1.2

WAP Name			UML 2.0	BPMN 1.2
WAP1: *Approval*	Single approval	*Requestor and Reviewer*	Two different partitions (A, B)	Two different Pools (A and B)
		Request for approval	Send Signal (from A to B)	Send activity (from A to B)
		Receiving of a request	Receive Signal (from A)	Start message event (from A to B)
		Performing of an approval	Elementary activity (executed by B)	Activity
		Send result of the approval	Send Signal (from B to A)	Send activity (from B to A) + Message Intermediate Event
WAP3: *Uni-directional*	Multi-request-response	*Sender and Receiver*	Two partitions (A, B)	Two different Pools (A and B)
		Request for activity execution	Send Signal to a *list* of performers (from A to B)	Subprocess composed by a send activity
		List of performers	Multi-instance activity	The subprocess is a multiple instance activity with look set as 'for each'
		Receiving of a request	Receive signal (for each member of the list)	Start message event (from A)
		Performing of activity	Elementary activity (executed by each member of the list)	Activity (executed by B)
WAP5: *Notification*	Single-notification	*Sender and Receiver*	Two partitions (A, B)	Two different Pools (A and B)
		Send notification	Send signal (from A to B)	Send activity (from A to B)
		Receive notification	Receive Signal (from A)	Start message event (from A)

represented by the three squares symbol as originally suggested by BPMN (cf. Figure 4). However, semantics is the same, since the activity execution request will be send for *each* participant of a list of participants. The execution will be in parallel i.e., the execution request will be send at the same time to participants of the process (concurrent execution). Considering the service interaction patterns the multi-request can be implemented as a One-to-many send pattern [1]. An alternative to represent a multiple instance activity in BPMN is to represent each receiver as single pool. This option, however, can result in large process models depending on the number of receiver pools [22]. In UML 2.0 we used the multi-instance activity (with parameters) to represent the concurrent request. From an illustration perspective the multi-instance activity from UML 2.0 looks clearer when compared to the idea of a looping sub-process in BPMN (see Table 2 for a comparative analyses). With Intalio we need to make annotation that the loop is *for each*. For sure this can be solved with a graphical extension of Intalio.

WAP5 Notification: Single-notification

In the *Single-notification* of the notification pattern a sender sends a message (notification) to a single receiver (cf. [15]). In BPMN this pattern can be illustrated with two *pools* representing a sender and a receiver respectively. The receiver gets a message from the sender (cf. Figure 5). In comparison with UML 2.0 the representation in BPMN is slightly clearer since the message in BPMN illustrates exactly the semantics of the Notification pattern. UML 2.0 is applicable to design process models but it does not provide a direct mapping of whatever business process element (e.g., notification). Table 2 summarizes the differences between the two mappings; i.e., BPMN 1.2 and UML 2.0.

6. CONCLUSIONS

This paper has reported on the purpose and requirements concerning the development of a BPM tool bases on workflow activity patterns. The main purpose of our tool are twofold: a) assisting users in designing process models; b) facilitate the development of experiments to empirically validate whether the activity patterns contribute to reduce real efforts for designing process models. In particular, we discussed the representation of activity patterns in BPMN 1.2 in comparation with the representation in UML 2.0.

Main advantages of our approach can be summarized as follows: (a) sufficiency and necessity of the activity patterns for process design has been evidenced in an earlier work at least with respect to process models similar to the ones we analyzed; (b) activity patterns are tool-independent which makes it easy to adopt them to any BPM suite; and (c) the recommendation mechanisms of our BPM activity patterns -based tool can be very useful not only to reduce the complexity in process design, but also to improve its correctness [5].

Regarding the representation of the activity patterns in BPMN basic advantages are: (a) BPMN is becoming a well-known standard notation for business process modeling. The representation of the patterns in BPMN may facilitate their adoption by some BPM tool based on BPMN. It must also become easier to perform experiments to compare process design with and without the support of activity patterns as

well as corresponding co-occurrences. (b) BPMN showed suitable for modeling most of the activity patterns and corresponding variants. When comparing with UML 2.0 some aspects look clearer like the message exchange between process participants. On the other hand, the representation of multiple instances looks more explicit in UML 2.0 when comparing with the representation provided by Intalio. We could observe that some structures (e.g., participants related to a multiple instance activity) can be represented in different ways in BPMN. Accordingly, we believe that not always our activity patterns will be directly identified (e.g., an approval can be designed in different ways). In the future we intend to investigate methods to deal with this.

Our next step is to choose the tool we will extend with the activity patterns and respective recommendation mechanism. Having the patterns implemented our first goal is to perform a series of experiments to verify whether activity patterns reuse help to reduce design efforts as well to increase quality of process design. We also consider to perform additional analyses of real processes models categorized by application domains. The goal with this analyses is to verify whether specific activity patterns are more predominant in particular domains (e.g., healthcare, automotive).

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] A. P. Barros, M. Dumas, and A. ter Hofstede. Service interaction patterns. In *Business Process Management*, pages 302–318, 2005.

[2] J. Becker, D. Pfeiffer, and M. Rackers. Domain specific process modelling in public administrations - the picture-approach. *Lecture Notes in Computer Science*, Electronic Government:68–79, 2007.

[3] E. Gamma. *Design Patterns*. Addison-Wesley, 1995.

[4] Intalio. Creating process flows. Technical report, 2006.

[5] J. M. Lau, C. Iochpe, L. H. Thom, and M. Reichert. Discovery and analysis of activity pattern co-occurrences in business process models. In *Proc. of the 11th International Conference on Enterprise Information Systems (ICEIS)*, pages 83–88, Milan, Italy, 2009. ICEIS Press.

[6] C. Li, M. Reichert, and A. Wombacher. On measuring process model similarity based on high-level change operations. In *Proc. of the 27th International Conference on Conceptual Modeling (ER)*, pages 248–264, Barcelona, Spain, 2008. Springer.

[7] OMG. *Business Process Modeling Notation (BPMN) Version 1.2*, 2009.

[8] R. Peisl. Discover and model your business processes with websphere business modeler. Technical report, IBM, 2008.

[9] M. Reichert, S. Rinderle, U. Kreher, and P. Dadam. Adaptive Process Management with ADEPT2. In *Int. Conference on Data Engineering (ICDE)*, pages 1113–1114. IEEE Computer Society Press, 2005.

[10] I. Scheer. Aris platform: Product brochure. Technical report, 2007.

[11] L. H. Thom. Workflow Activity Patterns in BPMN. Technical report, Ulm University, 2009. (under development).

[12] L. H. Thom, C. Iochpe, V. Amaral, and D. Viero. *Towards Workflow Block Activity Patterns for Reuse in Workflow Design*, pages 249–260. WfMC Workflow Handbook 2006. Lighthouse Point : Future Strategies, FL, USA, 2006.

[13] L. H. Thom, C. Iochpe, M. Reichert, B. Weber, M. Droop, G. Nascimento, and C. M. Chiao. On the Support of Activity Patterns in ProWAP: Case Studies, Formal Semantics, Tool Support. *Revista Brasileira de Sistemas de Informacao (iSYS)*, 01:27–53, 2008.

[14] L. H. Thom, M. Reichert, C. M. Chiao, C. Iochpe, and G. N. Hess. Inventing less, reusing more and adding intelligence to business process modeling. In *Proc. of the 19th International Conference on Database and Expert Systems Applications (DEXA)*, pages 837–850, Turim, Italy, 2009. Springer.

[15] L. H. Thom, M. Reichert, and C. Iochpe. Activity patterns in process-aware information systems: basic concepts and empirical evidence. *Int. Journal of Business Process Integration and Management*, 4(2):93–110, 2009.

[16] F. Tourniaire. *Just Enough CRM*, chapter 5, pages 113–150. Prentice Hall PTR, 2003.

[17] W. van der Aalst and A. ter Hofstede. YAWL: yet another workflow language. *Information Systems*, 30(4):245–275, June 2005.

[18] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51, July 2003.

[19] B. Weber and M. Reichert. Refactoring process models in large process repositories. In *Proc. of the 20th international conference on Advanced Information Systems Engineering (CAiSE)*, pages 124–139, Berlin, Heidelberg, 2008. Springer-Verlag.

[20] B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support features - enhancing flexibility in process-aware information systems. *Data and Knowledge Engineering*, 66:438–466, 2008.

[21] B. Weber, M. Reichert, S. Rinderle-Ma, and W. Wild. Providing integrated life cycle support in process-aware information systems. *Int. J. Cooperative Information Systems*, 18(1):115–165, 2009.

[22] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.

[23] P. Wohed, W. van der Aalst, M. Dumas, A. ter Hofstede, and N. Russell. Pattern-based Analysis of BPMN - An extensive evaluation of the Control-flow, the Data and the Resource Perspectives (revised version). Technical report, BPM Center.org, 2006.