Thomas S. Heinze, Thomas M. Prinz (Eds.)

Services and their Composition

7th Central European Workshop, ZEUS 2015
Jena, Germany, 19–20 February 2015
Proceedings

Volume Editors

Thomas S. Heinze
Friedrich Schiller University Jena, Institute of Computer Science
Ernst-Abbe-Platz 2, 07743 Jena, Germany
t.heinze@uni-jena.de

Thomas M. Prinz
Friedrich Schiller University Jena, Institute of Computer Science
Ernst-Abbe-Platz 2, 07743 Jena, Germany
thomas.prinz@uni-jena.de

Preface

As the ZEUS workshop series continues in providing a forum for young researchers to present and discuss their early ideas and work in progress centered around service technology and business process management, we were happy to host the seventh edition of the Central European Workshop on Services and their Composition (ZEUS) in Jena. For this years edition, we selected 11 submissions for the workshop program. Each submission went through a thorough peer-review process and was evaluated by at least three members of the program committee with respect to relevance and scientific quality. Accepted contributions covered topics in compilation of business processes, human-centered process design and execution, cloud-enabled applications, and process security. We would like to thank all authors and reviewers for their contributions and efforts. Finally, a special thank goes to Stefanie Rinderle-Ma (Workflow Systems and Technology Group, Vienna University) for completing the workshop program by an inspiring keynote on "Change and Compliance in Cross-Organizational Process Scenarios".

Jena, May 2015

Thomas S. Heinze
Thomas M. Prinz

Organization

ZEUS 2015 has been organized by the Chair for Software Engineering, Institute of Computer Science at Friedrich Schiller University Jena, Germany.

Steering Committee

Oliver Kopp	University of Stuttgart, Germany
Niels Lohmann	Carmeq GmbH, Germany
Karsten Wolf	University of Rostock, Germany

Local Organizers, Program Committee Co-chairs

Thomas S. Heinze	Friedrich Schiller University Jena, Germany
Thomas M. Prinz	Friedrich Schiller University Jena, Germany

Program Committee

Stefan Appel
Anne Baumgraß
Dirk Fahland
Christian Gierds
Georg Grossmann
Nico Herzberg
Meiko Jensen
Oliver Kopp
Agnes Koschmider
Matthias Kunze
Philipp Leitner
Jörg Lenhard
Henrik Leopold
Jan Mendling
Andreas Meyer
Stephan Reiff-Marganiec
Andreas Schönberger
Ruben Verborgh
Matthias Weidlich

Sponsors

IBM Deutschland GmbH, Ehningen, Germany
Adesso AG, Dortmund, Germany

Contents

Compilation of BPMN-based Integration Flows

Daniel Ritter

SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf
daniel.ritter@sap.com

Abstract Enterprise Application Integration plays an integral role for the communication between applications not only in service-oriented architectures. However, their modeling and configuration remain under-represented. In previous work, the integration control and data flow syntax and semantics have been expressed in the Business Process Model and Notation (BPMN) as a semantic model for message-based integration, while the concrete compilation to several runtime systems was left open. In this work we share our ideas for a general compilation approach along the *Message Redelivery on Exception* (MRoE) integration capability and basic message processing strategies, from which we derive compilation patterns. These patterns are used to translate BPMN models via an intermediate property graph model to a runtime system graph representation that allows the generation of executable runtime code.

Keywords: Business Process Model and Notation, Graph Model, Integration Flow, Message-based Integration, Runtime Systems

1 Introduction

Although *Enterprise Application Integration* (EAI) continues to receive widespread focus by organizations, e. g., for integrating existing business with cloud applications, the modeling of integration scenarios remains vendor-specific and covers mostly their control flow aspects [10]. Besides other requirements, suitable modeling approaches should offer (P1) well-defined, standard modeling capabilities for interoperability and ease of use, (P2) cover the integration semantics (e. g., message creation, routing), and (P3) executable runtime semantics. Most prominent, non-commercial examples are the *Enterprise Integration Pattern* (EIP) icon notation [7], the text-based *Apache Camel* [2] or the UML-based *Guaraná* DSLs, however, none of them supports all of the requirements (P1–3).

Our *Integration Flow* (IFlow) modeling approach [10], which is productively used in SAP's *Integration as a Service* product, maps the common EIPs and integration semantics (P2) to the *Business Process Model and Notation* (BPMN) [8] (P1), which is a "de-facto" standard for modeling business process semantics and their runtime behavior [10] and specifies their composition to integration and adapter processes [10,12] as well as their behavior in exceptional cases [13]. Despite some deviations (e. g., BPMN *Message Flow* as integration adapter, process instantiation / termination [10]), the IFlow execution semantics (P3) can

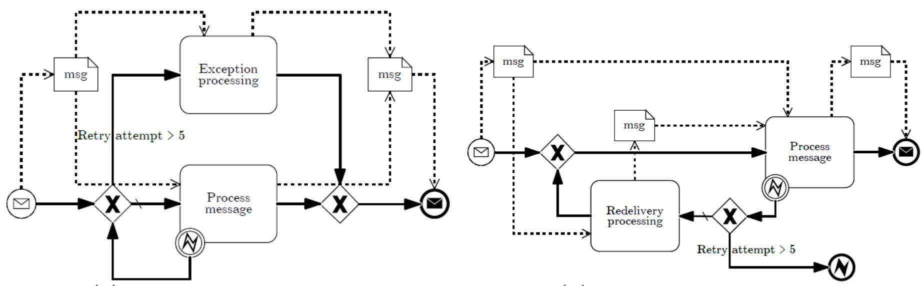
T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

be represented close to the BPMN specification, which makes IFlows partially executable on standard BPMN engines. Open remaining questions are the compilation to standard integration systems and a general compilation model for different kinds of integration runtime systems (e. g., databases [11]), which we address in this paper. Similar work can be found, e. g., in the related business process domain [9].

The contributions of this work are the collection of general compilation (model) requirements and the concrete list of capabilities for one integration processing aspect, i. e., message redelivery, in Sect. 2, the mapping of the most relevant BPMN concepts for IFlows to Apache Camel constructs representing a standard (open-source) integration system in Sect. 3, the definition of basic compilation patterns for common message processing strategies in Sect. 4, and a graph-based model and compilation approach explained by sample graph re-writings in Sect. 5.

2 General Requirements

The integration flows are a *Domain-specific Language* (DSL) in the sense of Fowler [6], from which we derived the following, general requirements. The XML representation of the IFlow–BPMN file shall be parsed (*REQ-1: Parser*) and populated to a *Semantic Model* (*REQ-2: Runtime independent Semantic Model*), i. e., an in-memory object model similar to the domain model. The semantic model shall capture the control- and data flow to represent the integration process and exception flow (*REQ-3: Capture flow semantics*). For instance, one important



(a) `Cap-1`, `Cap-2`, `Cap-4a` (b) `Cap-1`, `Cap-3`, `Cap-4b`

Figure 1. Message redelivery patterns in BPMN expressing capabilities `Cap-1`–`Cap-4`.

integration processing type is the *Message Redelivery on Exception* (MRoE; *REQ-4: Support MRoE*), which is ubiquitous in integration scenarios [13]. MRoE requires at least the following capabilities (`Cap-1-4`): `Cap-1` redeliver message $n \in N$ times, `Cap-2` process the exception, when retry limit is reached, `Cap-3` handle processing on redelivery attempts, `Cap-4a` continue after exhaustion or `Cap-4b` end the process after exhaustion of redeliveries. Figure 1(a) shows the message redelivery in BPMN fulfilling the capabilities `Cap-1`, `Cap-2` and `Cap-4a`, and Fig. 1(b) for `Cap-1`, `Cap-3` and `Cap-4b`. Eventually, the semantic model shall allow to generate code for multiple runtime systems (*REQ-5: Code generation*), which is packaged and deployed to the runtime system. We use *compilation* (i. e., code generation) over *interpretation* due to *REQ-2*.

3 From BPMN to Apache Camel in a Nutshell

In this section we sketch the idea of mapping from BPMN to runtime constructs. The lightweight integration system Apache Camel [2] represents the runtime system, which executes so called Camel routes (i. e., *Message Channel* [7] that can be combined to integration scenarios), described either in form of a Java DSL (used here) or several XML formats. Table 1 shows the mapping of BPMN elements used in IFlows to the corresponding Camel DSL statements. Notably,

Table 1. Camel Java DSL equivalents for BPMN elements

BPMN element	Camel DSL statement
Message Start Event / Message Receiving Activity	`from`
Plain Activity	`process`
Message End Event / Message Sending Activity	`to`
Sub-Process	`to`
Activity with Message Flows	`to`
Event Sub-Process	`errorHandler`
Transaction	`transacted`
Boundary Error Event	`onException`
Boundary Timer Event	`?timeout=value`

the message receiving / sending elements in BPMN represent *Message Endpoints* like HTTP, SOAP, FILE in the sense of [7], which are mapped to configurable `from` / `to` statements in Camel. BPMN activities are implementations of the `processor` interface in Camel, while BPMN sub-processes can be represented as separate Camel routes, which are addressable within the same Camel VM instance by `to` with additions like `to:direct` or `to:direct-vm`. The BPMN elements like event sub-process and boundary error event, used for the exception handling, find their counterparts only partially in the Camel `errorHandler` and `onException` statements, which are applicable on route or Camel context[1] instance level. The major issue is that the BPMN boundary element semantics cannot be adequately matched by route / context-level statements, since the matching Camel `try-catch` statement is incompatible with `onException` and `errorHandler` in version 2.x. Other BPMN boundary events like timer can be mapped to the Camel `timeout` on statement or route level. The specific MRoE capabilities of *REQ-4* can be mapped to Camel as shown in Table 2. While the MRoE has to be modeled in BPMN,

[1] A Camel context is a collection of routes, separated from other contexts at runtime.

Table 2. Camel Java DSL equivalents for capabilities.

BPMN pattern	Camel DSL statement
Retry pattern `Cap-1`	`maximumRedeliveries`
Retry pattern `Cap-2`	`process` / `to` (after onException definition)
Retry pattern `Cap-3`	`onRedelivery`
Retry pattern `Cap-4a`	`continued(true)`
Retry pattern `Cap-4b`	`continued(false)` (default, can be omitted)

we use the `onException` statement with additions like `maximumRedeliveries` for specifying the redelivery limit or `continued` to specify the behavior after the limit has been reached. The Camel `useOriginalMessage` statement (not shown) indicates whether the original message or the potentially modified one will be forwarded after an exception.

4 Basic Compiler patterns

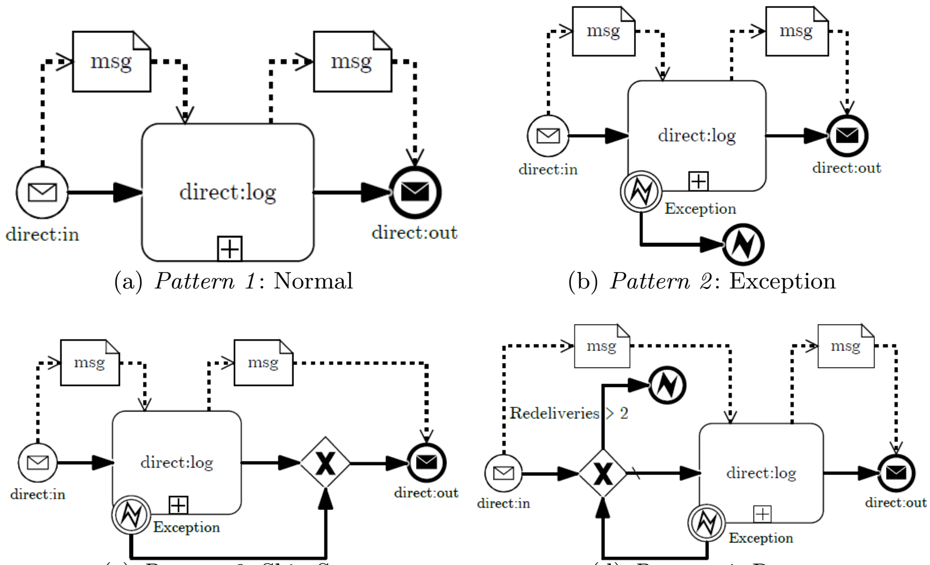
Starting with the normal processing, in this section we describe basic compiler patterns for the different MRoE processing types (cf., *REQ-4*). These patterns serve as building blocks for our compilation approach, since they enumerate the basic processing types within an integration process (cf., [13]) and their flow semantics (cf., *REQ-3*). The normal message processing (*Pattern 1*), depicted in Fig. 2(a), represents the integration process with control- and data flow (cf., *REQ-1*) for an integration process. The integration operations are indicated by a BPMN sub-process, which will become a separate Camel route with the Camel instance internal addressing `to:direct:log`. If an exception occurs during the processing of the sub-process, the process can be either stopped (cf., `Cap-4b`), as shown in Fig. 2(b) (*Pattern 2*), a message redelivery can be started (cf., `Cap-1`, `Cap-2`), sketched in Fig. 2(d) (*Pattern 4*), or the processing can be continued (`Cap-4a`), denoted in Fig. 2(c) (*Pattern 3*).

5 Compilation Models and Runtime Synthesis

In this section, we define a graph-based, (semantic) compilation model that fulfills *REQ-1–4*, explain the graph re-writing to a runtime graph representation and the generation of executable code (cf., *REQ-5*) by example of compiler *Pattern 4* and Apache Camel.

5.1 Compilation Models

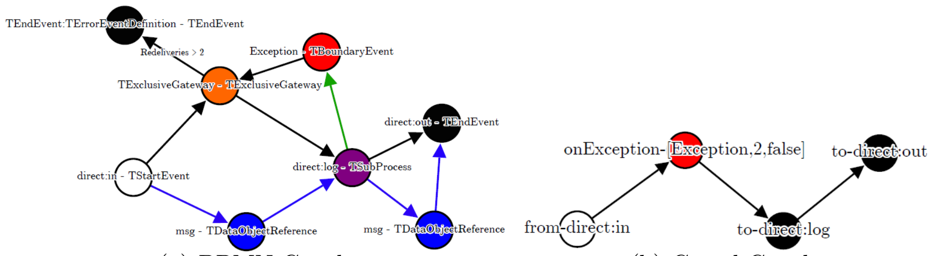
The key for the IFlow compilation to different runtime systems is a runtime-independent compilation model and approach (cf. *REQ-2*). Hence, we have decided for a compilation approach, where the semantic model is split into a logical (close to DSL) and a physical (runtime-near) representation. The *Logical*



(a) *Pattern 1*: Normal
(b) *Pattern 2*: Exception
(c) *Pattern 3*: Skip Step
(d) *Pattern 4*: Retry

Figure 2. Compiler Patterns 1–4: From normal processing (with BPMN *Sub-Process*) over exception handling to skip step and message redelivery.

Model (LM) is defined as directed property graph (e. g., refer to [14]), where the BPMN flow elements, data store and data object are vertices, and BPMN sequence flow, message flow and data associations are the edges. In this LM property graph, vertices and edges define semantic information about the represented elements, which are populated during the parsing (cf., *REQ-1*). Figure 3(a) shows the LM of the *Pattern 4* BPMN model from the `direct:in` node of type *TStartEvent* (white) to the `direct:out` *TEndEvent* (black), with the intermediate sub-process (purple) and the attached boundary error event (red) that uses an exclusive gateway for the message redelivery attempts that lead to another end event (e. g., of type *TErrorEventDefinition*) to stop the process after exhausted delivery. The data flow is denoted as *TDataObjectReference* nodes (blue). The LM is used to optimize the actual process model independent of the runtime along the given integration semantics (not further discussed). Then a runtime-specific re-writing logic is applied to translate the LM to its respective *Physical Runtime Model* (PRM). For instance, the PRM of an Apache Camel route is again a directed graph, where message endpoints (`from` (white) / `to` (black)), processors (`process`) and other route level statements (e. g., `onException` (red)) are nodes linked by edges, the control flow. Fig. 3(b) depicts the corresponding PRM for the discussed example. Notably, the resulting PRM for Camel does not make the data flow explicit, but assumes an implicit data flow handling in the Camel runtime. However, this observation helps to understand why the Camel DSL is not a (business) user facing model or LM for integration (P1), but rather a physical runtime model.



(a) BPMN Graph (b) Camel Graph

Figure 3. Compiler *Pattern 4* as Apache Camel model graph (right) from its BPMN graph (left).

5.2 Rule-based Logic to Physical Model Re-Writing

The translation from the runtime-independent LM to the specific physical runtime model has to be done for each runtime implementation by a system engineer. Hence, a pluggable, rule-based graph re-writing approach is preferred. Hereby, graph re-writing is conducted by graph node and edge traversers, which execute all applicable, registered rules (cf., Listing 1.1). Each rule specifies a `match` and an `execute` function. Listing 1.2 shows the implementation of the `match` function for detecting a MRoE (cf. *Pattern 4*). If the specified condition is evaluated to `true`, then the `execute` function for the current node is evaluated. Listing 1.3 shows the pseudo code for detecting the retry pattern: if a boundary error event is found during the `match`, the node is investigated further. The exception type becomes the name of the error event. Then, in case the leaving sequence flow leads to a gateway, preceding the activity, the redelivery count is extracted. If the outgoing flow leads to a succeeding gateway instead, no redeliveries are attempted and the route continues after the exception. Finally, the gather information is stored to the node and obsolete nodes are removed from the graph. Since the data flow of a Camel route cannot be configured explicitly, the BPMN message flow and data object associations are used to verify the correct behavior and the optimization of the LM. However, let us recall the `useOriginalMessage` statement from Sect. 3. With data flow information present, the case of "which message to use" in case of `continue==true` can be answered and applied if the runtime supports it.

Listing 1.1. Rewrite graph.

```
void rewrite (graph)
    rule1 =
        new detectFromEndpoints
    //...
    ruleX =
        new detectRetryPatterns
    runner =
        new ruleRunner(graph)
    runner.run(
        rule1, ..., ruleX)
```

Listing 1.2. `match` function.

```
match(node)
    return (node.getType()
    equals
    "BoundaryErrorEvent")
```

Listing 1.3. Detect retry patterns.

```
void execute(node)
    exception = node.getName()
    redeliveries = 0
    continued = false

    if node.leadsToPrecGtw()
        redeliveries =
            node.getPrecGtw()
            .getRedeliveries()
    if node.leadsToSuccGtw() OR
        node.getPrecGtw()
            .leadsToSuccGtw()
        continued = true

    node.add([exception,
        redeliveries,
        continued])
    node.getGraph()
        .rmObsolteNodes()
```

5.3 Code Generation

The resulting PRM contains all necessary information for the code generation. Listings 1.4 and 1.5 show the generated code for the compiler patterns *Pattern 3* and *Pattern 4* for comparison. The main difference between the two patterns lies in the behavior after an exception occured: *Pattern 3* continues with the processing (Listing 1.4, line 3), while *Pattern 4* starts with two message redelivery attempts (Listing 1.5, line 2, 3) and stops the execution through `handled(false)`, which ends the processing and throws the previously caught exception.

Listing 1.4. Camel DSL for *Pattern 3*

```
1 from("direct:in")
2  .onException(Exception.class)
3      .continued(true)
4  .end()
5  .to("direct:log")
6  .onException(Exception.class)
7      .handled(true)
8  .end()
9  .to("direct:out");
```

Listing 1.5. Camel DSL for *Pattern 4*

```
1 from("direct:in")
2  .onException(Exception.class)
3      .maximumRedeliveries(2)
4  .end()
5  .to("direct:log")
6  .onException(Exception.class)
7      .handled(false)
8  .end()
9  .to("direct:out");
```

6 Conclusion and Outlook

In this work we have collected basic requirements (cf., *REQ-1–5*) for the compilation of BPMN-based integration flows to integration runtime systems by example of Apache Camel. We sketched our idea for a runtime-independent, logical compilation model, identified compilation patterns for the case of message redelivery on exception and discussed the runtime-dependent re-writing to the physical runtime model with code generation. Our approach is comparable to the transformation from process model (i. e., IFlow), over executable workflow (i. e., Camel DSL) to IT infrastructure in Appel et al. [3] and the re-writes for security policies and compliance rules on μBPMN in Accorsi [1]. We decided for a "two-step" approach due to required compilation to different runtime systems (from one logical model) and separate optimizations on logical / physical level.

Future work will consider the optimization of the logical model along the integration semantics and the application of the compilation approach to other runtime systems. We will check the transformation of our logical model to colored petri nets for checking middleware designs for enterprise integration, e. g., Fahland et al. [5], and are highly interested in collaboration partners for the formalization of integration runtime systems, e. g., as in Dijkman et al. [4].

Acknowledgments Thanks go to Jan Sosulski for his contribution to the compiler implementation.

References

1. Accorsi, R.: On process rewriting for business process security. In: Proceedings of the 3rd International Symposium on Data-driven Process Discovery and Analysis, Riva del Garda, Italy, August 30, 2013. pp. 111–126 (2013)
2. Anstey, J., Zbarcea, H.: Camel in Action. Manning (2011)
3. Appel, S., Frischbier, S., Freudenreich, T., Buchmann, A.P.: Event stream processing units in business processes. In: Business Process Management - 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings. pp. 187–202 (2013)
4. Dijkman, R.M., Gorp, P.V.: BPMN 2.0 execution semantics formalized as graph rewrite rules. In: Business Process Modeling Notation - Second International Workshop, BPMN 2010, Potsdam, Germany, October 13-14, 2010. Proceedings. pp. 16–30 (2010)
5. Fahland, D., Gierds, C.: Analyzing and completing middleware designs for enterprise integration using coloured petri nets. In: CAiSE. pp. 400–416 (2013)
6. Fowler, M.: Domain Specific Languages. Addison-Wesley Professional, 1st edn. (2010)
7. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2003)
8. (OMG), O.M.G.: Business process model and notation (bpmn) version 2.0. Tech. rep. (jan 2011)

9. Prinz, T.M., Spieß, N., Amme, W.: A first step towards a compiler for business processes. In: Compiler Construction - 23rd International Conference, CC 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings. pp. 238–243 (2014)
10. Ritter, D.: Experiences with business process model and notation for modeling integration patterns. In: Modelling Foundations and Applications - 10th European Conference, ECMFA 2014, York, UK, July 21-25, 2014. Proceedings. pp. 254–266 (2014)
11. Ritter, D.: What about database-centric enterprise application integration? In: Proceedings of the 6th Central-European Workshop on Services and their Composition, ZEUS 2014, Potsdam, Germany, February 20-21, 2014. pp. 73–76 (2014)
12. Ritter, D., Holzleitner, M.: Integration adapter modeling. In: Advanced Information Systems Engineering - 27th International Conference, CAiSE 2015 (accepted), Stockholm, Sweden, June 8-12, 2015. Proceedings (2015)
13. Ritter, D., Sosulski, J.: Modeling exception flows in integration systems. In: 18th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2014, Ulm, Germany, September 1-5, 2014. pp. 12–21 (2014)
14. Rodriguez, M.A., Neubauer, P.: Constructions from dots and lines. Bulletin of the American Society for Information Science and Technology 36(6), 35–41 (2010)

Proposals for a Virtual Machine for Business Processes

Thomas M. Prinz

Chair of Software Technology, Friedrich Schiller University Jena, Germany
Thomas.Prinz@uni-jena.de

Abstract Business process management (BPM) promises an efficient and simple application of business processes. Although there are many approaches belonging to BPM, BPM suffers from the wide heterogeneity of tools and approaches and their missing integration into a homogenous workflow.
In this paper, we argue for a consistent system, which supports each step of BPM. The system consists of a virtual machine, a process intermediate representation, a compiler, and analyses. Thereby, the major focus lies on the derivation of requirements on a virtual machine for business processes and its intermediate representation. In a first conceptual approach, we present our proposals for such a virtual machine.

Keywords: Virtual machine, processes, business process management

1 Introduction

The state of the art in research of business process management (BPM) conveys a simple and efficient application in practice. Popular business process modeling languages, e.g., the Business Process Model and Notation (BPMN) [9] and the Web Services Business Process Excution Language (BPEL) [7], as well as verifiable process properties, which promise the absence of deadlocks for example (cf. soundness [1,13]), strengthen this impression. In addition to the prevalence of service-oriented architectures (SOA) in companies, there is no reason why the safe and dynamic application of BPM should not be able to work successfully: (1) the process is analyzed, (2) implemented, and (3) applied. During the process's use, it is (4) evaluated (monitored) and (5) improved continuously — the life cycle of a business process (cf. Fig. 1) [2].

However, Koehler et al. already recognized the lacks in the life cycle [8] such that the tool landscape of BPM suffers from its wide heterogeneity and its missing integration into a homogenous workflow. Take the development of a business process as example: In most graphical process modeling languages exist elements, whose semantics have obvious dependencies to the process developer's intension and are therefore not universally and basically definable. In addition, most common used process modeling languages (except BPEL) neither have clear definitions for documents nor for instructions to modify them. As result,

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

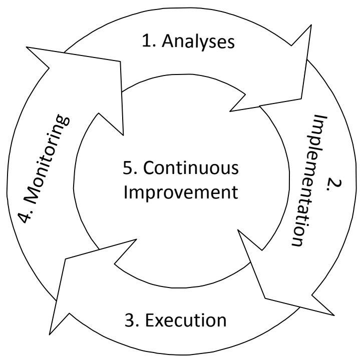


Figure 1. Process' life cycle

the majority of processes is performed manually and only an insignificant part of processes of an organization is executed by a process engine. In short: BPM would benefit from a system, which combines research approaches and which homogenizes the life cycle by new research results furthermore. In this paper, we focus on such a system and, especially, on requirements and on a conception of a homogenous virtual machine for the execution of business processes.

The remainder of this paper is as follows: At first, we explain our overall system's approach [11] (Section 2). In Section 3, we derive requirements of a machine, which are implemented in a first conception (Section 4). Eventually, we give an outlook on future work in Section 5.

2 Overall Approach

Our approach is a system, which is organized into a producer side (where the business process is developed) and a consumer side (where the process is then executed), inspired by Amme et al. [3] regarding mobile code. Our proposed system [11] consists of a compiler and a virtual machine (cf. Fig. 2), which support the development, analysis, storage, execution, maintenance, evaluation, and improvement of processes. We argue for a compiler since the development process of business processes has adjusted itself to the process of software development (cf. the life cycle of business processes to the spiral model [4]). Furthermore, the efficient development of software was improved through a well-defined computer architecture as well as the programming languages based on it. The consideration of state-of-the-art tools for business process development (for a list, visit `http://www.bpmn.org/`) shows that there is a wide heterogeneity of different execution techniques and that most tools are monolithically, i.e., it is hard to include additional analyses and to distribute processes between different execution engines.

Starting from these coniderations, the first step should be the definition of a (virtual) machine, which realizes the execution of a (technical) process. In

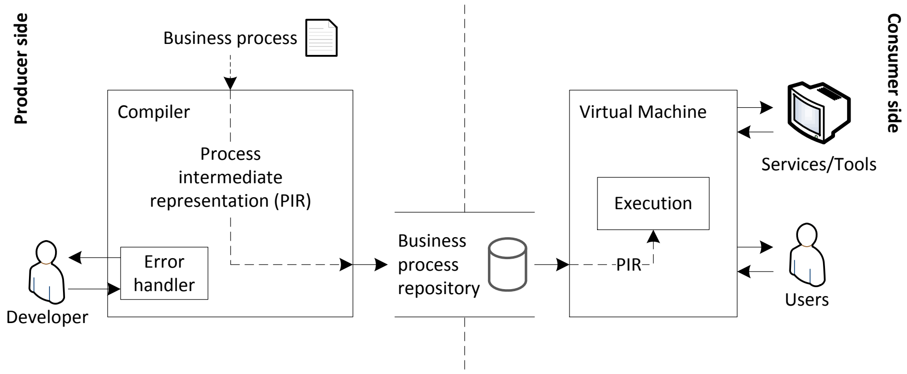


Figure 2. Overall system consisting of a compiler and a virtual machine

the second step, we should develop a technical and general process description language (process intermediate representation (PIR)). Afterwards, in the third step, existing process description languages should be modified in such that the automatic transformation into the PIR is possible. It is not the goal, in doing so, that the iterative derivation of a process from an abstract model to a technical process should be automated, since the inclusion of the process analyst is essential (cf. requirements analysis in software engineering). The goal is to define limitations for which such a transformation is possible at all and to allow a stepwise refinement of the business process into a technical one. In the last step, verifiable properties of processes based on the PIR should be found and some analyses should be developed which seriously support a solution-oriented failure diagnostic. Especially, in the context of business processes, such analyses must be applied to minimize the enormous costs in cases of malfunctions.

As mentioned before, in this paper, we focus on the first two steps: a conception of a virtual machine for the execution of business processes and the definition of a PIR. For this, we derive requirements for such a virtual machine in the next section.

3 Requirements on a Virtual Machine

One of the most famous virtual machines is the *Java Virtual Machine* (JVM). The JVM supports the execution of *Java Bytecode* (JBC), which is generated from *Java* programs by a compiler. The advantage is, that also other programming languages can be compiled to JBC and, therefore, programs based on multiple programming languages are possible. A process engine should follow that approach as it grows its flexibility. For this, *(R.1)* a virtual machine has to allow for the execution of a process code, i.e., the conception of such a machine affiliates with the language to be executed.

Popular process modeling languages have a quite large common subset of description elements on closer considerations. *(R.1.1)* A virtual machine should allow the handling of basic elements to model the control flow, e.g., exclusive decisions, parallelisms, inclusive decisions, and simple tasks. *(R.1.2)* These elements should be extended by explicit exception and error handling, high-language loops, events and signals, as they are frequently used in processes. Furthermore, *(R.1.3)* the inclusion of roles also is essential to ensure organizational structures as well as access permissions and common security. Eventually, *(R.2)* a process modeling language should naturally enable the usage of instructions and data structures for the modification of data and branch conditions. Such instructions are currently rare in previous languages as mentioned before. *(R.2.1)* The reception and sending of messages besides the usage of documents and conditions naturally are very important. With traditional programming languages in mind, *(R.2.2)* arithmetic operations, comparisons, objects, and method invocations on objects should be part of a process modeling language to modify and compare data. For this, documents are special cases of objects. Method invocations encapsulate the access on object fields and have to be restricted on state changes of objects, in such that side effects are rare and the analysis of processes is more simple. The start of sub processes is incumbent upon the "super" process.

(R.3) A virtual machine has to be able to load processes, which are ready to be started. As result, *(R.3.1)* the virtual machine should have a separate storage for processes. Besides the storage for processes, it needs a further data storage since a process being executed modifies objects. Furthermore, *(R.3.2)* the data storage should be separated (with reference to a single control flow) in a local and a global part as processes have commonly different parallel control flows with their own (private) data. *(R.3.3)* Each of those concurrent control flows should have its own local storage. It then has only the permission to access the local storages of those control flows leading to its execution, i.e., those local storages are global with regard to that single control flow. This is mandatory as some information is not safe on some points of the process. With this in mind, consider the process $P1$ of Fig. 3 in BPMN. This process defines a local object A in a task and starts two parallel sub processes $P2$ and $P3$. Both sub processes get the *global* object A and have access on it (like done in process $P2$). However, the process $P2$ has no permission to access the local object B of process $P3$ since $P3$ does not start $P2$ (neither direct nor indirect). Therefore, the instruction (3) *use* B is displaced. For this, *(R.3.4)* a virtual machine and the corresponding process intermediate representation should define the scopes of objects carefully.

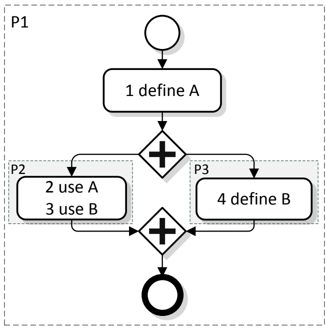


Figure 3. A process defines and uses data

The separation between local and global objects has less attention in previous research. The determination of local and global objects is simple in structured processes as it can be done like for traditional programming languages, e.g., by data flow analyses [5]. We propose the application of a process structure tree (PST) for unstructured or irreducible processes, who structures the process as a tree [14]. Although we worked on analyses for irreducible processes in previous work [10,12], we *(R.4)* plead in or for transformations to structured processes to simplify analyses or to make them possible at all. This step towards "structures" was already done for today's programming languages. However, the transformation of irreducible processes into structured ones is challenging since most algorithms (e.g., node splitting [6]) only work for sequential processes and have an exponential growth behavior of the entire graph.

The terms of local and global objects belong to a single control flow in processes and belong not to their spatial position. In general, *(R.5)* sub and partial processes of a process should be able to be executed on different physical machines. That belongs to the fact, that business processes are almost implemented by the use of SOA and that they work beyond company borders. As result, *(R.5.1)* a virtual machine has to abstract from the underlying physical network. For this reason, our considerations have a secure business network and standardized processes in mind as, otherwise, we have to know all details about processes which is commonly not the case. The difficulty is to identify communication partners in networks and to guarantee the correct usage of state-based services.

4 Conception of a Virtual Machine

The requirements mentioned before are implemented as a conception in the following. For this, Figure 4 shows the overall structure of our proposed virtual machine. Concluding from requirements *(R.1)* and *(R.3)*, the machine owns a *process loader* which loads a process and recognize the PIR part. We recommend to use a PIR in a tree representation as such a representation has many advantages [3] and leads to structured processes *(R.4)*. Furthermore, the representation should contain instructions *(R.1.1)* - *(R.2.2)*. Since the PIR is in a linearized file form, a *decoder* has to rebuild the PIR and, afterwards, a *verificator* checks the PIR for unallowed modifications with regard to security properties, e.g., soundness and reference safety. Then, the process is executed in the *process interpreter*.

As argued in previous work [11], a *dynamic semantic analyser* should monitor and analyze the process's execution to detect possible runtime process failures as early as possible. Those failures could be tried to be solved by a *runtime error handler* which interacts with the user or developer. So, a "rescue" of the process is still possible in failure situations and, afterwards, the process interpreter can continue its work.

The process interpreter gets the PIR and creates a new *process frame*. The structure of such a process frame illustrates Fig. 5. Each frame defines the state of each (sub) process and hence stores all necessary execution information *(R.3.1)*: A process object stack, a process extract, an in-/output, a process position pointer, and a process stack. The process stack (illustrated as arc in the figure) contains all process frames of started sub processes. Once a sub process starts,

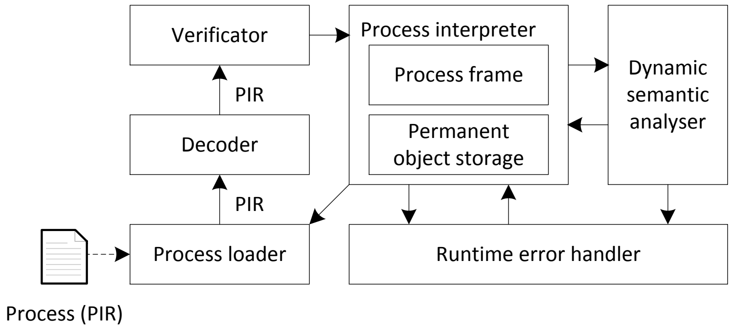


Figure 4. Proposals for a virtual machine for processes

the virtual machine creates a process frame for it. and pushs the frame on the process stack on the calling process. If the work of the sub process is done, its frame can be popped from the process stack. The "super" process can continue its work not until its process stack is empty (as, otherwise, it has to wait for the results of its sub processes). Although other implementations are possible, this stack-based approach reflects all hierarchic dependencies between different processes in a simple way. For example, Java uses a stack for its hierarchy of method invocations.

The process extract shows a subgraph of a tree representation (from the PIR) of the process that should be executed *(R.3.1)*. That tree representation is generated by the compiler with the help of the PST *(R.4)*. Each node of this tree has its own functionality (branch, parallelism, task, etc.) *(R.1.1) - (R.2.2)*. Through *link* edges in the tree, sequences are implicit and the tree can be traversed easily. Furthermore, that fact guarantees a simple reference safety, since global objects can be derived by a backward traverse from the current executed node to

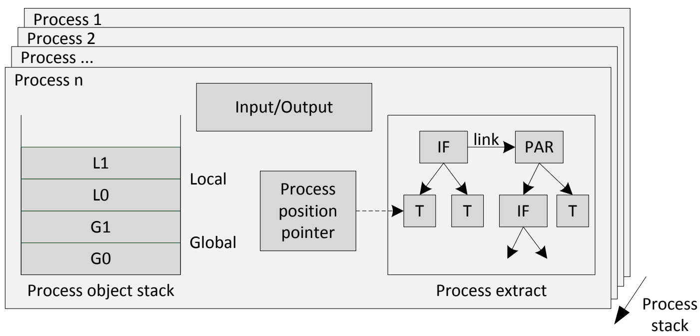


Figure 5. Structure of a process frame

the root of the tree *(R.3.4)*. The current executed node is marked by the process position pointer as the machine has to know which process's nodes are currently executed. The execution semantics of a node depends on its functionality. For example, if the current node produces parallelism, for each of its child nodes, the virtual machine creates a process frame, passes the process object stack, determines the subtree as process extract, and pushes the frame on the current process's process stack. After each sub process is finished, the process frames are popped from the process stack and the *link* node is executed afterwards. If an execution produces objects, they are pushed on the process object stack and they are locatable via this stack *(R.3.3)*. That stack contains all objects, which the (sub) process can access *(R.3.2),(R.3.4)*. Objects, which are stored over process borders, are stored within a permanent object storage (data base). Only processes, which have the permission, can access those objects.

If the execution arrives at an instruction for the reception or sending of a message and event, respectively, *(R.2.1)* the virtual machine uses the in-/output mechanism that handles the communication — regardless whether it talks locally or via a network *(R.5),(R.5.1)*. For this, each virtual machine can provide its processes by a RESTful web service, i.e., it automatically creates an unique uniform resource identifier (URI) for each process. Another process may then contact that URI via a network and gets an URI for the resource which points to the corresponding started instance of the process. Now, it is simply possible to communicate with a state-based process via a network. However, for this, it has to know its communication partners before.

5 Conclusion and Outlook

In this paper, we have extracted the need for a complete and consistent system for the development, analysis, and execution of processes in business process management. For this, we have focused on the conception of a virtual machine for the execution of processes. Besides the conception, we derived requirements for such a virtual machine.

In future work, these proposals have to be implemented and have to be compared to and extended for its application on business process management. Additionally, some parts of the machine (e.g., an efficient intermediate representation of a process) are in our focus of research.

References

1. van der Aalst, W.M.P., Hirnschall, A., Verbeek, H.M.W.E.: An alternative way to analyze workflow graphs. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Özsu, M.T. (eds.) Advanced Information Systems Engineering, 14th International Conference, CAiSE 2002, Toronto, Canada, May 27-31, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2348, pp. 535–552. Springer (2002), `http://link.springer.de/link/service/series/0558/bibs/2348/23480535.htm`

2. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: A survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) Business Process Management, International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2678, pp. 1–12. Springer (2003), http://dx.doi.org/10.1007/3-540-44895-0_1
3. Amme, W., Heinze, T.S., von Ronne, J.: Intermediate representations of mobile code. Informatica (Slovenia) 32(1), 1–25 (2008), http://www.informatica.si/PDF/32-1/11_Amme-Intermediate%20Representations%20of%20Mobile%20Code.pdf
4. Boehm, B.W.: A spiral model of software development and enhancement. IEEE Computer 21(5), 61–72 (1988), http://doi.ieeecomputersociety.org/10.1109/2.59
5. Hecht, M.S.: Flow Analysis of Computer Programs. Elsevier Science Inc., New York, NY, USA (1977)
6. Hecht, M.S., Ullman, J.D.: Flow graph reducibility. SIAM J. Comput. 1(2), 188–202 (1972), http://dx.doi.org/10.1137/0201014
7. Juric, M.B.: Business Process Execution Language for Web Services BPEL and BPEL4WS 2Nd Edition. Packt Publishing (2006)
8. Koehler, J., Gschwind, T., Küster, J.M., Völzer, H., Zimmermann, O.: Towards a compiler for business-it systems - A vision statement complemented with a research agenda. In: Huzar, Z., Kocí, R., Meyer, B., Walter, B., Zendulka, J. (eds.) Software Engineering Techniques - Third IFIP TC 2 Central and East European Conference, CEE-SET 2008, Brno, Czech Republic, October 13-15, 2008, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4980, pp. 1–19. Springer (2008), http://dx.doi.org/10.1007/978-3-642-22386-0_1
9. OMG: Business Process Model and Notation 2.0. formal/2011-01-03 (2011), http://www.omg.org/spec/BPMN/2.0
10. Prinz, T.M., Amme, W.: Practical compiler-based user support during the development of business processes. In: Lomuscio, A., Nepal, S., Patrizi, F., Benatallah, B., Brandic, I. (eds.) Service-Oriented Computing - ICSOC 2013 Workshops - CCSA, CSB, PASCEB, SWESE, WESOA, and PhD Symposium, Berlin, Germany, December 2-5, 2013. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8377, pp. 40–53. Springer (2013), http://dx.doi.org/10.1007/978-3-319-06859-6_5
11. Prinz, T.M., Heinze, T.S., Amme, W., Kretzschmar, J., Beckstein, C.: Towards a compiler for business processes - a research agenda. In: de Barros, M., Rückemann, C.P. (eds.) SERVICE COMPUTATION 2015, The Seventh International Conferences on Advanced Service Computing. pp. 49–55. IARIA, IARIA XPS Press, Nice, France (March 22 - 27 2015)
12. Prinz, T.M., Spieß, N., Amme, W.: A first step towards a compiler for business processes. In: Cohen, A. (ed.) Compiler Construction - 23rd International Conference, CC 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8409, pp. 238–243. Springer (2014), http://dx.doi.org/10.1007/978-3-642-54807-9_14
13. Sadiq, W., Orlowska, M.E.: Analyzing process models using graph reduction techniques. Inf. Syst. 25(2), 117–134 (2000)
14. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. Data Knowl. Eng. 68(9), 793–818 (2009), http://dx.doi.org/10.1016/j.datak.2009.02.015

A Method for Modeling and Analyzing Business Processes for Knowledge Carriers

Timm Caporale

Institute of Applied Informatics and Formal Description Methods (AIFB)
Kaiserstrasse 89, 76133 Karlsruhe, Germany
timm.caporale@kit.edu

Abstract Driven by external changes, organizations have to adapt and improve their business processes more often and at short notice. Thus, there is a need for effective and efficient methods for business process management. The modeling and analysis of existing business processes thereby represents the basis for the subsequent adjustment process and is significant for its later realization. Existing and current approaches attempt to support the modeling and analyzing of the processes in order to make them more effective and efficient in many different ways, for example through the definition of more extensive languages, new perspectives, or by developing new supporting software systems. Nevertheless, the modeled processes often do not comply with the expectations of the parties involved. Unlike the existing approaches, the presented method should allow to shift the modeling process towards the knowledge carriers, so that workflows and business prcesses can be modeled more precisely.

1 Motivation

Organizations are increasingly influenced by external conditions and changes such as the global competition, complex business models and greater customer focus, and must respond flexibly to them. Therefore, specific strategies and flexible business processes are needed to adapt quickly to continuously changing internal and external conditions. Usually business processes are modeled, analyzed, adjusted and established iteratively in the company. A major challenge is to make this "process of change" manageable. Different methods, tools and techniques have been developed in the field of business process management to support the change process. When modeling a business process, the manual and (semi-)automatic activities, which are carried out by and with specific resources for a particular purpose, are documented [9].

The strategies of most existing approaches attempt to use the implicit knowledge of the resources involved in the modeling and adjustment processes. This is usually done through a process analyst, who observes knowledge carriers during their work and asks questions about their activities. To successfully complete this process step, the process analyst must have a lot of knowledge about the methods, models and languages of business process modeling.

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

Apart from the obvious problems that occur due to the continuous observation of the process participants, a lack of knowledge about the activities can lead to inaccurate or incomplete business process models [7].

2 Solution

Thus, the here presented idea is to shift the modeling activities towards the knowledge carriers, because only the knowledge carriers themselves have complete information about input, sequence and output of their own activities. The paper proposes a new method for modeling business process through knowledge carriers.

In order to enable the knowledge carriers to model their own processes without having the knowledge of the process analyst about business process modeling techniques, the modeling process needs to be supported by a software system in a way, that the knowledge carriers are capable to model their business processes in an appropriate form on their own. To achieve this goal, the following three major research questions have been identified:

1. What information must be retrieved from the knowledge carriers in order to yield significant process models?

Similar to classical methods (e.g. the expert interviews), information about required resources, people involved in the process and the intermediates and final products must be identified.

2. How can a process model be created without knowledge about modeling languages and modeling patterns?

The method used to answer research question 2 combines three techniques:

- First, the knowledge carriers get a set of requirement templates for the corresponding modeling patterns, which enable them to describe their activities in natural language. The natural language is restricted to the syntax of the requirement templates.
- Secondly, a software tool receives the sentences, created with the help of the requirement templates. The sentences can be typed in or can be recorded and analyzed by a voice recognition tool. The (recognition) tool then transforms the sentences into the corresponding modelling patterns used in the process model.
- Thirdly, to avoid errors, the tool displays both the entered text and the recognized patterns. Using this approach, every knowledge carrier models only his own activities. The ability to know and use a modeling language is not required at all. Techniques from the field of Natural Language Processing, a formal model for the pattern transformation, and software engineering design principles are used to achieve this goal. This technique aims abstract process models for documentation.

3. How can the process models be analyzed?

The idea is here to use a formal object model for roles, resources, intermediate and final products. Based on this model, an algorithm should be able to link the created process models with each other semi-automatically using similarity-based analysis methods. This enables the business process analyst to identify higher-level business processes and analyze these on schema level.

3 Conclusion

The paper proposes a new method for modeling and analyzing business processes through knowledge carriers. The topic as such is not new and subject to various types of research. The paper proposes using natural language processing techniques to transform domain knowledge text into abstract process models. The presented method should increase the information content of the business process models of an organization and thus the documented process knowledge, too. Problems of classical process documentation should be reduced through the simplified modeling and analysis process. Moreover, it is conceivable that the total cost of process modeling and analysis can be reduced by this method, and thus the demand for better methods and shorter adaptation cycles is fulfilled. This can be a costeffective support for change processes for small and medium sized enterprises.

4 Related Work

Existing and current approaches attempt to improve the modeling and analyzing of business processes in several ways.

Some, for example, present a more extensive language to standardize basic processes through new symbols in the process model [10]. In fact, some evaluations describe this approach as making the modeling process even more complex through too many rules and symbols [11].

Another approach is to discover business processes from other sources than the knowledge carriers. With Process Mining [1] it's possible to discover business processes from log files. But the prerequirements to make Process Mining work are not always fulfilled. For many Use Cases, when information is only available from knowledge carriers, Process Mining is not practicable.

Fleischmann [2] changes the point of view on the business process. He places the subject into the center of the modeling process and discovers inputs and outputs from this point of view. But still much effort is necessary to interview the knowledge carriers and model correct business process models.

Other research groups develop lightweight and agile software systems [12,6] to provide a more effective and efficient process modeling, which is the right way from our point of view.

One more approach is, to bring process modeling experts and knowledge carriers on one table like [8,5].

But in contrast to our approach, none of these try to really give the knowledge carriers the ability to model business processes independently. Shifting the modeling activities towards the knowledge carriers and making the interviewing and observing process unnecessary is the central idea of our approach. Research in the field of Natural Language Processing like [3,4] enables this method.

References

1. Van der Aalst, W.M., Weijters, A.: Process mining: a research agenda. Computers in industry 53(3), 231–244 (2004)
2. Fleischmann, A., Schmidt, W., Stary, C., Augl, M.M.: Agiles prozessmanagement mittels subjektorientierung. HMD Praxis der Wirtschaftsinformatik 50(2), 64–76 (2013)
3. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Advanced Information Systems Engineering. pp. 482–496. Springer (2011)
4. Keuter, B.: Bidirektionale Abbildung zwischen Geschäftsprozessmodellen und IT-Kommunikationssystemen. KIT Scientific Publishing (2014)
5. Lübbe, A.: Tangible Business Process Modeling: Design and evaluation of a process model elicitation technique. Ph.D. thesis (2011)
6. Margaria, T., Steffen, B.: Business process modelling in the jabc: the one-thingapproach. Handbook of Research on Business Process Modeling pp. 1–26 (2009)
7. Mazarakis, A., Braun, S., Kunzmann, C., Schmidt, A.P., Stocker, A., Richter, A., Koch, M. (eds.): Motivation und kulturelle Barrieren bei der Wissensteilung im Enterprise 2.0 (2011)
8. Mevius, M., Ortner, E., Wiedmann, P.: Gebrauchssprachliche modellierung als grundlage für agiles geschäftsprozessmanagement (2014)
9. Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen: Zugl.: Karlsruhe, Univ., Habil.-Schr., 1995. Teubner-Reihe Wirtschaftsinformatik, Teubner, Stuttgart (1996)
10. White, S.A.: Introduction to bpmn. IBM Corporation (2004)
11. Wohed, P., van der Aalst, W.M., Dumas, M., ter Hofstede, A.H., Russell, N.: On the suitability of BPMN for business process modelling. Springer (2006)
12. Xie, L., Xu, L., de Vrieze, P.: Lightweight business process modelling. In: E-Business and E-Government (ICEE), 2010 International Conference on. pp. 183–186. IEEE (2010)

Personal BPM – Bringing the Power of Business Process Management to the User

Jonas Lehner

Institute of Applied Informatics and Formal Description Methods (AIFB)
Karlsruhe Institute of Technology (KIT)
Kaiserstr. 89, 76133 Karlsruhe, Germany
jonas.lehner@kit.edu

Abstract External and internal factors force companies to be flexible in their business processes. However, the information systems used in the companies are often too complex and the implemented process models are too rigid to quickly adopt changing determining conditions. This leads to a loss of efficiency since various tasks that would be suitable for the execution by workflow management systems must be performed manually. The presented approach for a solution solves this problem by offering a method to exploit the automation potential by using suitable flexible tools and a target-group oriented procedure model. Compared to existing approaches this method does not try to solve the problem on a company level. Instead the focus is on the individual employee who will be enabled to improve his personal workflows.

Keywords: Personal BPM, Personal Workflow Management, JSON Nets

1 Introduction

Process models are often developed by top management and are imposed on employees (top-down). Usually this is performed by a process modelling specialist who is not equipped with the necessary domain knowledge. This may lead to a situation where the employee is faced with a process model that does not represent his actual needs and therefore does not support him in his daily business. In the worst case it will even constrain him in the fulfilment of his duties. Bandera et al. identify the “perceived gaps between process design and process execution” as one of the top operational issues in BPM [1]. This means that many processes, which have a great potential for automation, are not supported by central IT systems. Hence, it would be useful to offer a way to create user-level workflows for personal use to improve the productivity of individual members of an organization.

Companies have to deal with internal and external drivers and need to be flexible in their business processes which may lead to a problem since the information systems used in the companies cannot respond quickly to changing conditions. Hence, optimization potential is not used because employees have to perform tasks manually instead of using a workflow management system. Indulska et. al see problems with change management and a lack of governance [2].

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

2 Personal BPM

Personal BPM tries to address these problems by providing a user centred perspective on business process management and the execution of workflows. In this context, the *user* is understood to be any member of an organization, who is not inevitably a modelling expert. He has to have a certain level of understanding regarding the use of IT. To benefit from the automation, a substantial part of his work should be done on a computer. Examples for users in this context are agents in an insurance or scientific researchers in a university.

Following the BPM lifecycle (often described, e. g., [9]) the user needs to be supported in different activities:

1. The user has to understand which activities are eligible for automation and has to identify the dependencies between single tasks (*process design*). Here, the user has to be supported by an appropriate procedure model.
2. He then has to implement the workflow (*system configuration*). On the one hand this has to be simple so that the user is not deterred and on the other hand the system has to be flexible so that it meets the user's needs.
3. The execution of workflows has to be simple and should require as little manual activities as possible (*process enactment*).
4. To allow continuous process improvements there has to be a tool to measure key data like process time (*diagnosis*).

Business process management is widely used by companies. But as shown before its benefits does not always reach the operational level and therefore the single employee, which leads to the following main research question:

RQ0 *How can the benefits of using BPM be transferred to the operating range of individual members of an organization?*

This main research question can be divided into four questions concerning different aspect:

RQ1 *What language is suitable for modelling business processes in this context?*
Business processes are usually modelled using graphical languages like BPMN, EPC or Petri nets. This is suitable for modelling specialists, while domain specialists may have problems in understanding and using it.

RQ2 *How does a workflow management system have to be like that is flexible enough to be used by domain specialists?*
Workflow management systems are complex software systems that are usually configured by IT experts and cannot be easily customized by the end user.

RQ3 *How can the development process for new features be supported?*
A system that has to meet the requirements of the users has to be expandable in its functions. Hence, developers of additional features have to be encouraged by the architecture of the system.

RQ4 *How is the productivity of individual employees affected by the deployment of Personal BPM?*
If using a software system takes more time than doing something manually it should not be utilized. Therefore an evaluation has to be performed to show if it is useful to use Personal BPM in daily business.

Business processes as an ordered set of activities to achieve a specific operational objective can be modelled with different languages, e. g., BPMN, UML and Petri nets. XML nets represent a variant of higher Petri nets, which can model business processes based on the exchange of relevant information objects [3]. In this respect, the language is suitable for the application described here. However, the use of XML requires a relatively rigid data schema that limits the flexibility of a system which uses this language. Therefore, it is advisable to select a data format that allows to map any data schema and is easy to handle.

3 Related Work

The idea of using Petri nets to describe runnable workflows is often described (e. g., [8]). However, it is usually used by a modelling specialist not by a normal employee.

One approach to close the gap between modelling specialists and domain experts is described by Luebbe and Weske [4]. They use methods of design thinking to make process modelling more understandable for the user. In so-called "Tangible Business Process Modelling" plastic elements, which represent BPMN iconography, are used to model business processes in a playful manner. Although this approach helps to improve communication between knowledge providers and modelling experts the modelled processes remain inflexible with respect to changing conditions.

Petric and van der Aalst present an approach for flexible business process management that addresses the problem of managing dynamic processes in rapidly changing organizations by shifting from an imperative paradigm to a declarative one [5]. This means, it is rather declared what should be done without specifying how something should be done. While this may help to model a company's processes from a high level point of view, it is hard to use these models to automate tasks as concrete execution instructions are missing.

In contrast, there are a couple of services, that use some kind of personal workflows like IFTTT (http://ifttt.com) or the mobile app "workflow" (http://workflow.is). They allow users to manage different social media services and connect them with simple rules, e. g., save a photo in dropbox, that a user posted on facebook or change the colour of a smart home light bulb when a new email is received. But these services are not suitable for a business context since they only allow simple linear processes and support a closed set of services.

There is a common approach that uses personal task management to enable end-user driven business process composition [6,7]. In contrast, the idea described in this paper focuses on data flowing through the process rather than concentrating on the tasks.

References

1. Bandara, W., Indulska, M., Chong, S., Sadiq, S.: Major issues in business process management: An expert perspective. ECIS 2007 Proceedings (2007)

2. Indulska, M., Chong, S., Bandara, W., Sadiq, S., Rosemann, M.: Major issues in business process management: An australian perspective. ACIS 2006 Proceedings (2006)
3. Lenz, K.: Modeling interorganizational workflows with xml nets. In: System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on (2001)
4. Luebbe, A., Weske, M.: Bringing design thinking to business process modeling. In: Meinel, C., Leifer, L., Plattner, H. (eds.) Design Thinking, pp. 181–195. Understanding Innovation, Springer Berlin Heidelberg (2011)
5. Pesic, M., van der Aalst, W.M.P.: A declarative approach for flexible business processes management. In: Eder, J., Dustdar, S. (eds.) Business Process Management Workshops, Lecture Notes in Computer Science, vol. 4103, pp. 169–180. Springer Berlin Heidelberg (2006)
6. Stoitsev, T., Scheidl, S., Flentge, F., Muhlhauser, M.: Enabling end-user driven business process composition through programming by example in a collaborative task management system. In: Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on. pp. 157–165 (2008)
7. Stoitsev, T., Scheidl, S., Flentge, F., Mühlhäuser, M.: From personal task management to end-user driven business process modeling. In: Dumas, M., Reichert, M., Shan, M.C. (eds.) Business Process Management, Lecture Notes in Computer Science, vol. 5240, pp. 84–99. Springer Berlin Heidelberg (2008)
8. van der Aalst, W. M. P.: The application of petri nets to workflow management. Journal of circuits, systems, and computers 08(01), 21–66 (1998)
9. van der Aalst, W. M. P., ter Hofstede, Arthur H. M., Weske, M.: Business process management: A survey. In: van der Aalst, W. M.P, Weske, M. (eds.) Business Process Management, Lecture Notes in Computer Science, vol. 2678, pp. 1–12. Springer Berlin Heidelberg (2003)

The Business Process Game

Nico Herzberg[1] and Matthias Kunze[2]

[1] SAP Germany,
nico.herzberg@sap.com
[2] Hasso Plattner Institute (University of Potsdam),
matthias.kunze@hpi.de

Abstract Recently, gamification – the augmentation of work with game elements – attracted tremendous interest and has become one of the most important trends in digital business strategy[1].
In this paper, the authors elaborate on the opportunities to gamify operative business process management, in particular process enactment, and show various ways to engage process participants in their daily work, support them achieving their goals, and develop their skills. A concrete implementation of gamification in business process management is shown on a use case and a detailed discussion of future research is given.

1 Gamification in Business Process Management

Gamification is the discipline utilizing game elements, such as points, badges, and leaderboards, and experience design, e.g., game play, play space, and story line, to engage and motivate people to achieve their goals [1,2]. It targets at customers, communities of interest, and employees – in particular knowledge workers – as players and focuses their goals through player centered design [5]. To create value with a gamified solution the sponsoring organization needs to align supported player's goals with their own goals.

In this paper, we concentrate on the fact that gamification strives to engage knowledge workers to reach their goals by incorporating game elements into the context of their tasks and responsibilities within an organization's business processes. Motivation is seen as a means to support the worker in changing behaviors, in acquiring new skills, and building on existing expertise and performance.

With regards to business process management (BPM), we focus on the process participant, who carries out parts of business processes. Besides the execution of tasks in the context of the process, this includes to understand the underlying process model, its technical and organizational environment, i.e., resources and interaction partners, as well as documenting the results of the operations carried out. In contrast, other research approaches, for example, described by Santorum et al. [6,7] resort to social aspects, motivation, and gamification of process formalization and analysis, process modeling, and simulation.

Hence, by introducing game elements to operational aspects of BPM – becoming skilled in new processes, enacting them, and their improvement – we aim at

[1] http://www.gartner.com/newsroom/id/2702117

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

the process participant as our main game character or player. For this purpose, we introduce the following game elements.

Process scores Participants can receive scores for each business process they are involved in. The sum of scores for one process represents the participant's progress in mastering the process.

Badges, leaderboards Upon exceeding a score thresholds, participants are awarded with badges to publicly show their progress. Often, new features and responsibilities are associated with earned badges.

Points For each earned process score, the participant receives a point in their personal account. In contrast to process scores, these points can be traded for certain features or benefits.

Challenges Challenges are particular efforts that need to be carried out to gain additional scores.

Challenges, in particular, offer an opportunity to foster the collaboration between different process participants. In general, collaboration requires to hand-over work to other participants, which often results in delays in a business process as the receiving person needs to familiarize themselves with the incoming work. Hence, by challenging collaboration, we envision participants to work together more closely, e.g., in concise personal meetings.

2 Stages and Badges in Mastering a Business Process

As mentioned above, we aim at supporting process participants in mastering business processes. Therefore, we have identified three principal stages that describe, how familiar a participant is with a business process. Badges are associated with each stage, as stated in Figure 1. Here, we provide only the main badges that need to be earned to enter the next stage. To implement the approach, more badges should be awarded while traversing each stage, respectively.

Figure 1. Stages and badges of mastering a business process

Onboarding. At the beginning of the onboarding phase, the participant has no experience at all in carrying out the business process. This can be the case if the participant has been assigned new responsibilities or the process has been newly introduced or recently re-engineered. This phase is closely related with the first achievement, the *apprentice* badge, which is earned by familiarizing oneself with the business process. Process scores are earned by collecting information, and thus acquiring knowledge, about the process.

On the one hand, passing this stage can be achieved by setting up artificial cases in the fashion of a role-playing game: Apprentices are faced with a simulation of historic cases of the business process and need to carry them out. They need to take the right choices, access the right information, interact with adequate resources, and meet predetermined deadlines. On the other hand or subsequent to the role-play, the apprentice tackles real cases under the supervision of more experienced participants who mastered the same process such that they earned the *champion* badge.

Enactment. Once apprentices have gathered sufficient experience in the process, i.e., their process score surmounts a certain threshold, they are awarded with the process *master* badge and are ready to perform the business process without the supervision of a mentor. Nevertheless, mentors remain available for assistance, thus fostering collaboration between colleagues.

The enactment stage is the main stage and will be discussed further in Section 3. Process scores are earned by individual or collaborative performance and improvement. That is, getting better at carrying out operations, e.g., increasing the quality of results and the timeliness of documentation, and getting faster, e.g., reduce waiting and execution times or executing more cases. Key performance indicators can be used to align operational goals of business processes with the performance of process participants. However, such measurements must be designed carefully and transparently as not to counter collaboration or other quality or performance goals.

Improvement and Mentoring. Burke [1] states that gamification works best with intrinsic motivation, i.e., the emotional urge to personal development. One of these intrinsic factors is personal assistance, i.e., if persons are able to help, they receive implicit and explicit praise. Another intrinsic driver is the establishment of a positive impact on one's environment, e.g., a business process. Both require a high skill in process mastery, and therefore such process participants are awarded with the *champion* badge.

So far, we envision two ways to leverage intrinsic motivation. First, a very experienced process participant can become a mentor, who advices *apprentices* in the onboarding stage. Second, process participants can contribute to the improvement of business processes, both actively and passively. Actively, they can propose changes to the business process to improve its outcome or performance. Passive contribution can be carried out by "betting" points on a proposed change. If the change is applied successfully, passive contributors earn the amount of their stake and the proposer earns an amount that relates to the sum of stakes put into the proposal. If a business process is re-engineered, existing champions may be downgraded as they need to first familiarize themselves with the new process. However, the proposer and supporters of a change, who certainly understood it well, remain in the position of a process champion.

When applying gamification to BPM, one needs to bear in mind that the execution of a business process is, in general, a collaborative effort. That means that different people with different roles are involved as well as interaction partners, e.g., from other departments or organizations. However, as gamification aims at motivating individuals, any means to employ it in business processes requires well defined and transparent measures for the individual performance of process participants. We elaborate on this in the next section.

3 Gamified BPM in Action

In the following, we present a gamified BPM solution with the example of a process for specifying a job that we designed with a German health insurance provider, see Figure 2. The process is mainly driven by the *organization development* (expanded pool) but needs input from other departments of the company with which it interacts. The principles of gamification can be applied to onboard new process participants, to execute the process efficiently, and to set a scene for process improvement, as elaborated on in Section 2. In this section, we focus on the efficient execution of the process.

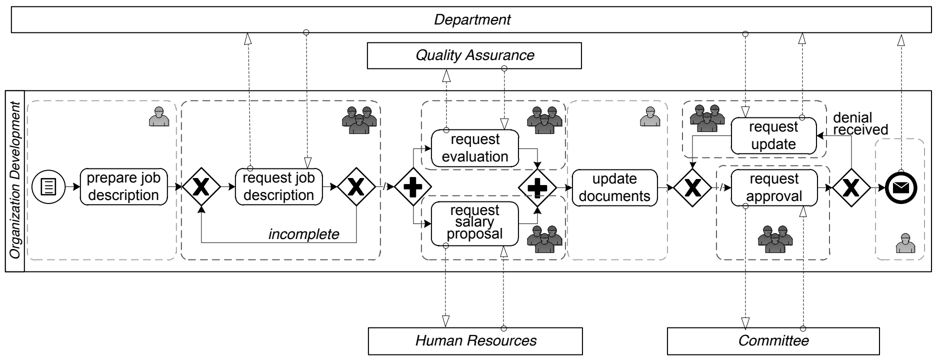


Figure 2. Business process model for specifying job descriptions at a German health insurance provider, including individual and collaborative tasks to earn process scores.

Once a new job specification is required, the *job description is prepared* for further interaction with the other departments. This is an individual task by an employee of the organization development department and thus, gamification targets on that individual (shown as a dashed line area with one individual) for these steps. In contrast, the following steps involve collaboration between the organization development employee and the department for which the job description is created (shown as a dashed line area with a group). In an iterative fashion between the two participants, the job description is to be completed successfully. Motivation of only one individual would lead to a decrease in quality, e.g., the organization development employee could quickly accept the job description without proper proofing for a quick completion and claim the lack of quality is due to the interaction partner. Therefore, and to sport the successful collaboration in the context of the business process, a gamification solution for several parties is required.

The completed job description must be evaluated by the quality assurance. Again, this is a collaborative effort as the employees of the quality assurance department and of the organization department should be stimulated to produce a joint result quickly. In parallel, the salary proposal from the human resources office is incorporated (collaborative). After these two steps, the documents are updated (individual) before approval is requested from the committee (collaborative). If the job description is approved, it can be sent to the department to be implemented (individual), otherwise it needs to be updated again (collaborative).

The activities *preparing the job description*, *updating the document*, and *sending out the approved job description* are an individual process participant's responsibility for earning process scores for the particular process to reach the next badge. In contrast, working on the other activities in the process depends on the work of other departments' employees as well. Thus, process scoring involves the performance of other people.

The concrete playground for the shown process is designed with the following levels that are rewarded with a badge when successfully completed, cf. Figure 2.

Introduction Learning about the process theoretically by studying all process documentation and understanding the process model. To become an *apprentice*, this level is completed by successfully passing a challenge, i.e., a test that asks questions about the process.

Advised process execution The process *apprentice* is responsible for executing the particular process, but is supported by a *champion* in the role of a mentor. Both of them can score by efficient process execution. The *apprentice* needs 100 process scores to become a *master*.

Personally improve execution performance The process *master* is responsible for executing the particular process and can score with an efficient process execution including individual and collaborative efforts. The *master* needs 1000 process scores to become a *champion*.

Process execution, improvement, and mentoring The process *champion* is responsible for efficient process execution, process improvement based on process model level, and mentoring process *apprentices*. *Champions* score in all three areas and get additional points when improving the process. To become a *guru*, *champions* need to score 1000 points year by year.

Process Development Being a *guru* enables an employee to access a pool of specific tasks of the business process management office, e.g., the design of completely new business processes or the consolidation of business process duplicates and variants that are the result of merged departments or companies. These specific tasks allow employees to reach the next level of their personal promotion.

Scoring while executing the business process is possible by performing the single activities efficiently and meeting KPIs, which are set as follows:

- Preparing the job description once it is requested within two days will result in 2 scores, within three days in 1 score, and in more than three days in 0 scores.
- Complete the job description with the department within three days will result in 2 scores, within five days in 1, and above in 0 scores.
- Get the evaluation done within one week will be rewarded with 1 score.
- Receiving the salary proposal within one week will be rewarded with 1 score.
- Updating the job description document with the salary information and with the changes proposed by the quality assurance within two days will result in 1 additional score.
- Receiving an approval/denial by the committee within two weeks (they are meeting only twice a month) will result in 1 score.
- Getting an approval by the committee in the first attempt is a challenge for all process participants contributing so far. Hence, 5 extra scores are given to each participant involved in this case prior to approval.
- Working on a requested update after a job description denial within two weeks will result in 1 score.
- Sending out the approved job description within 2 days will result in 1 score.

Meeting KPIs within a business process execution environment could be measured by applying approaches such as proposed by Herzberg et al. [3,4].

Rewarding collaborative tasks fosters the interaction between participants and encourages them to team up, e.g., in phone calls or personal meetings, instead of just handing over work items. Furthermore, the interaction partners of the business process can earn points with their immediate contribution.

Points earned in either of the above ways, or by means of successful process improvement, if the participant is a champion, can – in turn – be spent on supporting particular process model improvement proposals, which other champions propose. Thereby, process model improvements are evaluated by all process champions. Promising process model improvements will be promoted by the support of many process champions, non-promising improvements will be downgraded. Successful process model improvements that result in value for the company will be rewarded by points, whereas improvements that are not implemented or not successful will result in the loss of the spent points. This, however, requires that process management respects the impact of improvement promotions by champions.

4 Game Credits

In this paper, we outlined how the principles of gamification can be applied in business process management to support onboarding new process participants, engaging participants in efficient process execution, and improving the process. Completing process activities and accomplishment of challenges is rewarded with process scores and the development of individuals is awarded with badges. These unlock new levels of responsibility, e.g., to mentor new process participants or vote for process improvement proposals. A case study has been presented using the example of a job description process designed for a German health insurance provider.

We already envision various directions for future research. First and foremost, a study needs to be conducted to evaluate how and to what extend knowledge workers get influenced by a gamified business process execution environment. We have discovered arguments that support a positive effect on process performance, but also ones that suggest the opposite, i.e., overall performance decrease.

One potential reason for a negative impact is the risk of tricking the system. If, for instance, only the execution of few pilot business processes is gamified, employees may concentrate their effort on these processes, while non-gamified work falls short. Also, scoring measures may be cheated, if they are not carefully designed.

Additionally, rewarding of efforts shall be investigated in more detail. This includes aspects such as rewarding collaborative efforts in more detail, e.g., by distributing scores received as a team to the individual team members according to their respective contribution to the joint achievement. Also, negative rewards in case of insufficient performance need to be explored.

Finally, we argue that, for gamification to become successful, it should be incorporated into the lifecycle of operative business process management, and not only in its execution phase. We envision modeling aspects that allow the specification of gamification mechanisms and measures directly in the business process models, such that they are also implemented when the model is put into operation.

References

1. B. Burke. *Gamify: How Gamification Motivates People to Do Extraordinary Things*. Bibliomotion, Incorporated, 2014.
2. Sebastian Deterding. Gamification: Designing for Motivation. *interactions*, 19(4):14–17. ACM, 2012.
3. Nico Herzberg, Matthias Kunze, and Mathias Weske. Monitoring business process interaction. In *ICSOC 2012 Workshops*, pages 228–240. Springer, 2013.
4. Nico Herzberg, Andreas Meyer, and Mathias Weske. An Event Processing Platform for Business Process Management. In *EDOC'13*, pages 107–116. IEEE, 2013.
5. Janaki Mythily Kumar and Mario Herger. *Gamification at Work: Designing Engaging Business Software*. Interaction Design Foundation, 2013.
6. Marco Santorum. *A serious game based method for business process management*. In RCIS 2011, pages 1–12. IEEE, 2011.
7. Marco Santorum, Agnès Front, abd Dominique Rieu. *ISEAsy: A Social Business Process Management Platform*. In *Business Process Management Workshops*, pages 125–137. Springer, 2014.

BPM in German Companies – Information Gathering

Felix Baumann and Dieter Roller

Institut für Rechnergestützte Ingenieursysteme, Universiät Stuttgart

Abstract This position paper outlines a method devised to gather information on companies involved in BPM from publicly available data. The method is based on searching for job openings on job search engines which are related to BPM. The assumption is that companies searching for new employees with skills regarding BPM are either involved in BPM or are about to be involved in BPM.

Keywords: BPM, BPM in Germany, Crawler, Publicly Available Data, Web Scraping

1 Motivation

The statements made in the literature on the use of BPM in businesses are mainly based on statements which are provided by companies or their employees in the form of interviews and surveys [3] [1]. These surveys have partially the shortcoming that the company would like to represent itself positively or employees give answers in the surveys in accordance to the social desirability response set. Our idea is to identify companies involved in BPM by using their search for new employees on job sites. These sites are public as these companies need to address potential future employees. We can use this knowledge of the searching companies to make assertion on the prevalence of BPM within the German economy. For this purpose, publicly available sources, particularly job search engines are used to gather this information employin web scraping technology[5],[2]. False positive results can be excluded because companies will not search actively for new employees on topics/areas that are irrelevant for the organisation. There might be a bias in this case because we might miss some companies searching for employees when they don't use the correct keywords in their job descriptions when searching for the appropriate staff. After we have identified the companies/organisations that apply BPM or are about to apply BPM we can use the job advertisements itself in order to gain information on the necessary competencies/skills and related fields of work. Additionally information on the duration of the existence of each job advertisement can lead to information on the availability of BPM experts. In a further step, it is possible to correlate the firms found with publicly available information (self-representation on the Internet and corresponding sites like XING) and to draw conclusions about the geographical distribution of the companies, its business and number of employees. Thus it would be possible to support previous research results regarding the use of BPM, such as [7], [6], [4].

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

2 Technical Base

This work is based on the automated search and indexation of job advertisement with relation to BPM. As the sources of information the homepage of the Bundesagentur für Arbeit [1] - with currently 829516 listed open positions (as of 2015-01-27) - and the German homepage of Stepstone [2] - with currently 56398 listed open positions (as of 2015-01-27) - were chosen. The homepage of the Bundesagentur für Arbeit has been chosen because it is the governmental institution handling unemployed persons or persons currently searching for a job. Furthermore this site integrates other job search engines within its own search engine and makes those data sources available through a single interface. Stepstone has been chosen as a data source to counterbalance the governmental page of the JobBoerse as a privately owned entity. Stepstone is by its own accord Germanys job search engine number 1 ("Deutschlands Jobbörse Nr.1"). Other job search engines e.g. the German homepage of Monster[3] have not been used because they either did give no account on the number of jobs they are listing or are available through the search engine of the JobBoerse - e.g. XING[4] - or would only give replicates of jobs found on the two search engines used.

2.1 Research questions

We have identified the following set of questions that we want to answer.

1. Questions related to enterprises: (a) Is this method valid to acquire data on enterprises employing BPM? (b) Which enterprises do use BPM? (c) What kind of companies are these? (d) Where are they geographically located? (e) What is the size of these companies? (f) In which domain(s) are these companies located?
2. Questions related to the qualification of BPM practitioners: (a) What are qualities applicants are expected to have? (b) Is there an overlap between BPM and other fields of work like cloud computing? (c) What positions are BPM practitioners employed in? (d) What is the level of experience required? (e) Is the required skillset changing? (f) Do enterprises look for the same qualities as described in scientific literature[8]?
3. General questions: (a) What is the duration of each job advertisement? This could be indicative of abundance of of qualified employees? (b) Are the job openings reccurent (for the seemingly same position)? (c) Is there a general trend of enterprises employing BPM?

2.2 Implementation

Starting with the analysis of the datasources, we have devised a crawler able to search those datasources for a given keyword using the search functionality

[1] `https://jobboerse.arbeitsagentur.de`
[2] `http://www.stepstone.de`
[3] `http://www.monster.de`
[4] `http://www.xing.de`

provided by said sites. The data crawler consists of scripts which split the work into the following parts

1. Form the search query
2. Filter the results accordingly to the searched for information
3. Gather further information, e.g. URL of employer
4. Save the results in a persistent way

The crawler uses a variety of Linux software in order to fulfill the specific tasks (e.g. wget[5] and curl[6]) and to filter the results by using streamline processing e.g. awk[7] and sed[8]. The respective HTML pages are stored for documentation purposes twofold. They are stored as HTML files and audit-proof as PDF. The information extracted are stored in corresponding XML files and additionally in a SQLite[9] database.

2.3 Data Acquisition

The actual keywords/phrases are being sent to the respective data sources/search engines daily. The crawler handles the communication with the search engines and stores the data persistently. Data collection has started 2014-09-18 and is ongoing. Currently there is 131 days worth of data available. There are 9 search queries with 9 keywords/phrases going to both search engines. The actual search phrases in use are: 1. “yawl” 2. “epk” 3. “epc” 4. “business process management” 5. “prozess modell” 6. “prozessmanager” 7. “sysml” 8. “bpmn” 9. “bpm”

2.4 Preliminary Data Analysis

The current data set consists of 767314 HTML pages out of which 104791 (13%) are originating from JobBoerse, 289065 (37%) from Stepstone and another 204737 (26%) pages from third parties accessed through the JobBoerse search engine. The remaining 24% of the pages is broken or otherwise impaired. These HTML pages have not been filtered for duplicates as of yet but there are certainly double entries that will be filtered out. Searching for unique job titles yields 7231 results. On average there seems to be 55 job openings per day spread over 2 search engines and 9 different keywords/phrases. Taking those different keywords and the different search engines into account it results in 3.06 job openings per day per keyword gathered.

2.5 Challenges

After the preliminary analysis the following challenges have arisen:

---

[5] `https://www.gnu.org/s/wget`
[6] `http://curl.haxx.se`
[7] `http://www.gnu.org/s/gawk`
[8] `https://www.gnu.org/software/sed`
[9] `http://www.sqlite.org`

– Job openings by consultancy companies. Job advertisements by these companies can not necessarily be correlated to a specific enterprise which does the actual BPM work. Employees might be loaned to other companies or work as consultants for other companies. These job advertisements are identifiable by the company name and its classification as a consultancy company.
– Variations of the labour market. There might be other factors e.g. crises, wars or weather influencing the overall labour market. Taking other search phrases from other areas of information technology related areas into the data collection can reduce the effect of these variations as they can function as a baseline. Suggested terms include 1. cloud computing 2. data mining 3. java (programmer) 4. service engineer .

3 Conclusion and Outlook

The main contribution of our work will be an extensive set of data relating to companies involved in BPM, especially their geographic distribution, company size, domain and business. Furthermore the contribution will be a dataset of skills and related areas of business of people practising BPM in the real world. This data can be extend where necessary and used to support existing research in BPM using surveys and interviews based on participant selection. This is an ongoing experiment and we will be able to produce results without the need to halt data collection. A longer duration of data collection is necessary to track variations and trends in BPM and the associated job advertisements.

References

1. Cebeci, C., Kol, E.: Analysis for the Implementation of the Business Process Management in Selected Turkish Enterprises. International Journal of Economics and Financial Issues 3(2), 420–425 (2013)
2. Herrouz, A., Khentout, C., Djoudi, M.: Overview of Web Content Mining Tools. The International Journal Of Engineering And Science (IJES) 2, 106–110 (7 2013), http://adsabs.harvard.edu/abs/2013arXiv1307.1024H
3. Imanipour, N., Talebi, K., Rezazadeh, S.: Obstacles in Business process Management implementation and adoption in SMEs (1 2012), http://ssrn.com/abstract=1990609
4. Knuppertz, T., Schnägelberger, S., Clauberg, K.: Umfrage Status Quo Prozessmanagement 2010/2011. Tech. rep. (2011), http://www.bpmo.de/bpmo/export/sites/default/de/know_how/content/downloads/Status_Quo_Prozessmanagement_2011.pdf
5. Marres, N., Weltevrede, E.: Scraping the Social? Issues in live social research. Journal of Cultural Economy 6, 313–335 (4 2013), http://dx.doi.org/10.1080/17530350.2013.772070
6. Müller, T., Thome, R., Vogeler, K.: Zukunftsthema Geschäftsprozessmanagement. Tech. rep. (2011), http://www.pwc.de/de/prozessoptimierung
7. Wolf, C., Harmon, P.: The State of Business Process Management 2012. Tech. rep. (2012), http://www.bptrends.com/bpt/wp-content/surveys/2012-_BPT%20SURVEY-3-12-12-CW-PH.pdf

8. Xu, L.D.: Enterprise Systems: State-of-the-Art and Future Trends. In: IEEE Transactions on Industrial Informatics. vol. 7, pp. 630–640. IEEE (09 2011), http://dx.doi.org/10.1109/TII.2011.2167156

A Framework for Interactive Exception Management of Long-term Human-Involved Business Processes

Johannes Kretzschmar and Clemens Beckstein

Institute for Computer Science
Friedrich Schiller University Jena
E-Mail: {johannes.kretzschmar, clemens.beckstein}@uni-jena.de

Abstract While contemporary business modelling languages offer the possibility of describing processes on an abstract operative level there is no computer assisted support in monitoring and exception handling of such workflows. In this paper we discuss an approach to determine a formal representation of workflow execution states by applying methods of artificial intelligence (AI) planning. The hereby introduced framework *BPdoc* generates a questionnaires for human workflow participants based on semantically enriched business models and transforms the observations into a sufficient propositional state representation. The framework is triggered externally in exceptional cases and can anticipate future impacts on process soundness and correctness. By this, it serves as a foundation for further failure handling by a business administrator or by computer assisted methods.

Keywords: BPM, BAM, failure detection, exception management, AI planning

1 Introduction

Nowadays, Information Technology is an integral part of Business Process Management (BPM). Various formal languages and process engines allow the modelling and execution of complex workflows. Especially the ability of automatically monitoring process execution and visualizing comprehensive runtime information is an important benefit for upper management and process administration. The data gathered by business activity monitoring (BAM) services are essential for the optimization of workflow descriptions during the BPM life-cycle [1]. BAM provides useful information about bottlenecks or unreliable actions by evaluating triggers, counters, metrics or various performance indicators. The monitoring rely on data provided by a BP engine executing a workflow instance. Thereby BAM only works for processes with a precise executable description.
Contemporary workflow modelling languages like the Business Process Model and Notation (BPMN 2.0) [2] offer rich vocabulary to describe processes on a abstract operative level. Languages of this type are necessary for modelling processes in highly dynamic environments with human participation and nested,

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

complex interactions. The corresponding workflow models are generally meant for formalization, visualization and optimization. They are not executable by a BP engine and cannot be handled by BAM services for this reason.
In this paper, we introduce an approach to provide BAM-like services for such abstract workflow descriptions. The framework named *BPdoc* is a semi-automatic computer assistance for determining a current execution state, identifying failures and estimating the impact on future business activities as well as process goals. *BPdoc* operations are primarily based on methods and techniques of artificial intelligence (AI) planning.

AI Planning for Process Management

AI planning techniques are already applied to the field of BPM for evaluating, assisted developing and repairing process models. This development was primarily driven by processes as service-choreographies and related semantic service description standards [3]. But the high complexity of planning algorithms limits the application to very specific use-cases. In classical planning, there are assumptions of the domain models concerning time, execution, observability and influence aspects. These assumptions guarantee feasible planning algorithms, but in most cases there is a huge gap of expressiveness between such domain models and common workflow applications.
AI planning is based on a formal descriptive domain model [4]. This domain model provides the vocabulary to describe a state in a specific domain. This can be done, for example, by a set of propositions. A proposition is an element of a state-representing set, if the corresponding property is observable. Another part of a planning domain is a set of actions, which might be applied to states. Every action is annotated by preconditions and effects. A precondition controls, whether an action is applicable in a state and the effects describe the impact. With this information, an algorithm can calculate a new state by executing a set of actions in a given state. Therefore, the applicability, overall effects and goal reachability of processes can be computed by using methods of AI planning.
The research topic of using AI planning techniques in the field of BPM is not the focus of this paper, because we are not using the capability of planning for generating workflows from scratch. We are concentrating on the two methods from planning for applying actions (soundness) and checking states for propositions (correctness). These are essential parts of a planning algorithm for unfolding the search-space and evaluating the correctness of a solution. By leaving the costly planning algorithm aside, we also may narrow the mentioned gap and cover a wider range of workflow applications and planning domain models.

2 The BPdoc Framework for Exception Management

In this paper, we are focussing on workflows with a high abstraction level. This type of workflows is characterized by a long-lasting runtime, involving human participation, highly dynamic domain and they rely on real world resources. There exists no proper runtime monitoring for these kind of processes, because it is impractical to describe such processes on an executable level. Not to speak of the problem to install sensor technology observing the real world domain and define proper triggers and performance parameters.

We propose the framework *BPdoc*, which is able to determine a representation of the current workflow execution. This approach relies on a given semantically annotated process model, a domain description and a proposition-based representation of the state where the process was deployed. Besides the preconditions and effects, every action is annotated with a specific task associate. As shown in figure 1, *BPdoc* consists of a controller and an evaluator. The controller can generate, send, receive and process messages to and from every process participant.

For deriving a current state representation, the *BPdoc*-controller has to retrieve the state of real world workflow execution first. Each process participant sends a status messages to *BPdoc* after accomplishing a task successfully. After every message, the controller uses the *BPdoc*-evaluator to calculate an ad hoc proposition-set-representation by applying the effects of the corresponding operator from the business model. Therefore, *BPdoc* always holds a representation of the current state of workflow exeution. This representation is delayed in reference to the busines environment state, because there are only messages after every successfully accomplished task and participants may continue after sending their status messages.

The framework does not rely on continuous runtime-update messages. This approach also works by asking all participants at once, for the tasks they are currently performing and determining all the finished tasks from the process model. In cases of unordered or parallel tasks in plans, this information may be incomplete and *BPdoc* has to ask further questions. This method is, compared to the runtime-update method mentioned before, hereby more efficient, because the framework is only supposed to assist in exceptional cases and is no runtime monitoring tool.

3 Exception and Failure Detection

BPdoc gets triggered by participants or administrators in case of an unexpected event. By default, the process model and the derived propositional state only holds information about successfully completed tasks. So, the main difficulty is the formalisation of real world observations into the set-theoretic state representation of the domain model. Due the abstract and non-executable workflow models we are looking at, an automatic translation is only possible in few particular cases. To overcome this problem, *BPdoc* tries to gather information from the human process participants, who are inherent process observers on a small scope, too. But the capability of observation and autonomous failure detection counterweights a lack of formalization. Not every process participant can be assumed to be a formal knowledge engineer. We solve this problem by generating simple questions, where formal propositions according to the (semantic) process model are checked. These questions are answered in `YES` / `NO`-responses by process participants. This binary pattern correspondents to the propositional set state representation, as mentioned in section 1. This schema enables *BPdoc* to be triggered and work in two main exception cases: the non-applicability of actions and unfinished or failed tasks.

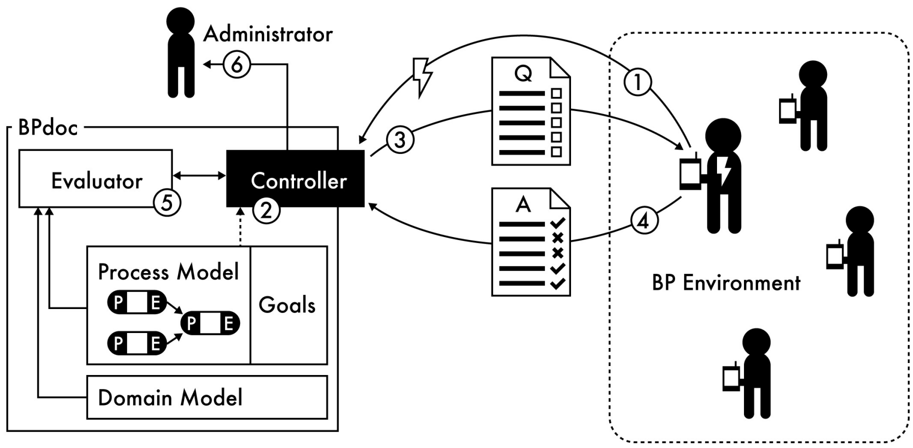


Figure 1. The interaction of *BPdoc* and the process environment

Non-Applicability Exception

This type of exception occurs, if a participants realizes that an assigned task cannot be started due missing requirements and manually triggers the *BPdoc*-controller (see figure 1 (1)). In this case, the framework has to determine the apparent discrepancy of the current state and the required preconditions of the task (2). This can by done by generating a set of questions concerning the propositions of the precondition (3). The participant has to verify these propositions by answering the question (4). By transforming the response into a virtual effect-set and applying these propositions, *BPdoc* generates an updated state representation, which reflect the current state of execution as observed by the process participant. Further *BPdoc* can test the remaining process for soundness and correctness (5) by applying outstanding workflow actions to the actual state. With help of *BPdoc*, the administrator gets information about earlier actions (6), which were not fulfilled as presumed and led to missing preconditions, an actual state representation and possible future flaws concerning the process execution or goal reachability.

Unfinished Task Exception

This type of exception arises, if a task fails in progress and the accountable participant triggers the *BPdoc*-controller. The execution semantics of state-of-the-art workflow models as well as classical planning always assume that a state remains after executing failed actions. But especially in the field of abstract human-involved workflows, activities often consist of multiple complex tasks. In these cases we have to assume, that even irregular performed actions may effect the state in an unforeseen way.
BPdoc tries to identify these side effetcs by generating questions based on the precondition and effect set of the failed operator. It is necessary to figure out, which propositions of the precondition are still existent and which effects have already occurred. The questions are generated and asked according to the non-applicability-exceptions and lead to an updated representation of the execution state. Similar to the first case, the administrator gets information about the failed execution of an action and the consequences for the process execution.

4 Conclusion

We showed that BAM-like services are also possible in processes with abstract and operative descriptions without any dedicated monitoring BP engine. The proposed framework *BPdoc* requires a semantic description of process elements and appropriate observers of the real world environment. A simple questionnaire is automatically producible and raises utilizable information corresponding a planning domain representation. Once triggered, *BPdoc* is able to determine a formal representation of the current workflow execution state and to infer prospective effects on a process. By this, the framework delivers practical information, which are essential for process repair by either a workflow modeller or an AI planner.

Our approach reveals a strong link between a domain model and failure model of processes: The more propositions a domain model holds, the finer grained is a potential detectable failure model. The *BPdoc*-framework fails to handle a specific exception, if the propositions of the domain model do not suffice to generate questions, whose answers determine the situation. So, a triggered controller but no significant change in state of execution may imply an insufficient domain model and delivers a positive refactoring impulse for the process modeller. By this, *BPdoc* may be a useful tool in the analysis and modelling part of every BPM-lifecycle.

5 Future Work

In this approach we have assumed workflow models with an control flow corresponding to a partial order plan. Modern process description languages like BPMN enable much more constructs, like loops, OR and XOR gateways. We need to stretch the plan as well as the domain model to enhance *BPdoc* handling more comprehensive workflows. Our main research in this direction is the rule language for describing aspects of the process domain. An adequate language would enable a modeller to keep the set of propositions small and would support *BPdoc* in generating efficient specific questions.
Until now, we have focused on the fundamental possibility of translating an dialogue with human process participants into a corresponding domain model. There should be a dedicated research about the usability of such a system and how it could be improved. An effective and efficient communication should be as easy and user-friendly as possible, without loosing its expressiveness.

References

1. McCoy, D., Schulte, R., Buytendijk, F., Rayner, N. and Tiedrich, A., *Business Activity Monitoring: The Promise and Reality.* Gartner, Gartner's Marketing Knowledge and Technology Commentary COM-13-9992, 2001.
2. *BPMN 2.0 specification.* `http://www.omg.org/spec/BPMN/2.0/PDF/`, 2011 (accessed March, 12th 2015)
3. Rao, J.,Su, X., *A Survey of Automated Web Service Composition Methods.* In Proceedings of 1st international workshop semantic web services and web process composition (pp. 43–54), 2004
4. Ghallab, M., Nau, D. and Traverso, P., *Automated Planning, theory and practice.* Morgan Kaufmann, 2004

Challenging Service Extensions for Electric Vehicles in Massively Heterogenic System Landscapes

Sebastian Apel, Thomas M. Prinz, and Volkmar Schau

Department of Computer Science
Friedrich-Schiller-University Jena
Jena, Germany
{sebastian.apel, thomas.prinz, volkmar.schau}@uni-jena.de

Abstract The national goal of the federal government of Germany is to have one million fully electric vehicles in use until year 2020. Since an average tour of vehicles in inner-city logistic is about 80km, it is promising that such logistic companies use electric vehicles for their tours. However, there are some reasons for doubts in companies: unknown behaviors, inaccurate range forecasts, and high costs.
The research project Smart City Logistik will provide a service to optimize, monitor, and analyze electric vehicles in logistic scenarios for abolishing those barriers. There, the integration of a service system in existing IT-solutions by using predefined interfaces is challenging. This service system needs special implementations for each company solution, what could be avoided by using a more standardized way in logistic softwares. The overhead can be reduced by a harmonized terminology in combination with a standardized service bus to enable specific internal service extensions in existing logistic solutions.

Keywords: Reference architecture, electric vehicles, logistic, interface, transport management system

1 Introduction

To enable inner-city logistics with electric vehicles, the German Federal Ministry for Economic Affairs and Energy (BMWi) has started, amongst others, the collaborative research project Smart City Logistik (SCL). The project combines an intelligent tour planning system with dynamic range prediction, based on detailed knowledge of major influencing factors. The SCL service system is able to support dispatchers and drivers with real-time feedback, by optimizing the driving style, and by providing alternatives for a successful end of planned tours.

That SCL system should extend existing IT-solutions by using their data and by offering standardized interfaces for service requests rather than substitute them. However, the use case of proceeding external data about orders, customers,

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

drivers, and fleets to calculate optimized and energy efficient tours is not common – but in case of increasingly rapid innovation cycles more necessary.

As part of the project, SCL develops a reference architectural design for data exchanges between independent service systems in logistical scenarios based on a harmonized terminology. This design will describe how to enable or extend an existing IT-solution to provide data, how to receive them by newly created services, and how to embed the results back to existing IT-solutions. Furthermore, this design describes how to provide these results for other, more specific services.

To achieve this goal, the paper starts with an explanation about the software landscape in logistic IT-solutions, how they are clustered and how they are used in companies (Section 2). Followed by that, there are multiple use cases as examples in which these state-of-the-art solutions will lead to problems (Section 3). Afterwards, a first draft for a reference architecture for service-based extensions in existing IT-solutions is derived in Sect. 4. Eventually, the paper considers related work (Section 5) and concludes with a short outlook into future work (Section 6).

2 Software Landscape in Logistic Companies

Enterprise resource planning (ERP) describes a wide range of tasks handled by companies to manage their resourcese. g., financial accounting, data services, manufacturing, and human resources, as well as customer relationship management and supply chain management (SCM). One focus of companies with logistic background is the SCM while ignoring human resources and financial accounting at this point. This part of ERP describes a whole range of mechanisms and theories for supply chains. In detail, these chains start with acquisition, followed by production and distribution, and are finalized by sales [2]. SCM covers the whole production process and the logistic to handle these workflows. As a part of this, there are system solutions (transport management system (TMS) and warehouse management system (WMS)) to cover special parts of the SCM. These TMS and WMS handle requirements found in inner-city logistic scenarios like package, food, drug, and express delivery, as well as services, e.g., automate maintenance and restocking. A TMS covers functionalities about fleet, employee, order, and billing management. Furthermore, it contains tour and route calculation, tracking, statistics, and dispatching. A WMS on the other hand handles management of warehouses, positions, and amounts of stored articles, and helps to manage dispatching centers.

These functionalities of TMS and WMS will be combined to get a solution for each company to cover their needs. A closer look at these solutions will show a number of more then 50 different tools [1, 5]. Each tool has an unique set of interfaces and has more or less of the before described elements of TMS and WMS theories. Such interfaces mostly covers common DATEV and financial accounting interfaces, along with less interfaces based on electronic data interchange (EDI) standards[1] or proprietary REST, SOAP or COM solutions [5]. These interfaces

[1] A standardized set of document descriptions for data exchanges.

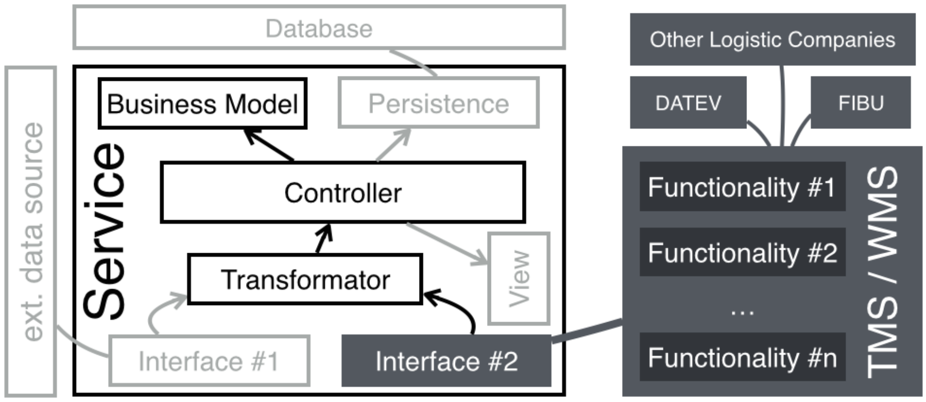


Figure 1. Abstracted service architecture based on the Model-View-Controller pattern.

are mostly designed to interact with other companies to realize transportation chains and managing orders.

An abstract Model-View-Controller based service to communicate with a TMS/WMS is illustrated in Fig. 1. This service shows an interaction with external datasources (like weather, traffic, and electricity rates information) on the left hand side, which is mostly unproblematic because of their public availability. The interaction with a TMS/WMS as illustrated on the right hand side is challenging, but necessary to enable ideas as described before.

3 Use-Cases for Services

In case of logistic, there is a wide range of innovations e. g., by using electric or hybrid vehicles. Those innovations could lead to required new services. Such services must be used to enable the innovations in existing environments. However, the current landscape in software solutions for logistic companies does not allow for their easy embedding as mentioned before. There are several use-cases for such services currently considered in research projects.

The first example is the optimization of energy consumptions and costs by using electrical cars in company fleets as energy buffer[2]. These buffers are used, when the provider's energy price is above a predefined threshold. To enable such solutions, the company needs an energy management system deciding which source of energy is used. In case of prices greater than the threshold, buffered energy can be consumed. In case of prices lower than this threshold, provider energy can be used. In case of a planned tour for a specific vehicle, that energy management system must ensure, that the vehicle is prepared in time. To realize such a management system, the service needs to know when vehicles have to start tours. For example, by using a read access to vehicle time sheets and planned tours from an existing TMS/WMS solution.

[2] iZeus Smart Grid `www.izeus.de`

Another example is the management of charge scheduling[3]. A lot of current installed energy systems are not suitable to handle the full amount of required power in case of simultaneous fleet charging. As a result of this, it is necessary to schedule the charging processes to avoid network peeks. Again, this could be done by using a special management system which decides, which vehicle is allowed for recharging now. Using a read access for vehicle time sheets and fleet information in existing IT-systems would enable such decisions.

A third example is an optimized routing algorithm for electric vehicles as suggested in the SCL project[4]. Currently, combustion engine based vehicles can refill their fuel when necessary. Because of missing charging infrastructure and a high amount of time to charge, this is not suitable for electric vehicles. To avoid stranded vehicles, a forecasting system should be used to plan energy-aware tours. Such a system could be used as extension by using data about vehicles, drivers and orders from a TMS/WMS. That extension would work as an external service with read access for required data. Additionally, this solution could push back the results to the TMS/WMS service, which requires an additional write access for tour data.

The examples illustrate situations in which an extension of TMS/WMS is necessary. In practice, this requires an individual extension in each system instance, developed by the original author of the applied solution. As an alternative, an external service can be based on available interfaces. In this case, there would be no standardized communication, the service has to use available REST, SOAP and COM interfaces and provide an individual implementation to access the data. Furthermore, this means an individual interface implementation in each TMS/WMS solution for each example.

4 Towards a Reference Architecture for Logistic Service Communications

A reference architecture is profitable for the integration of new services especially to reduce complexity in data exchanges. Such an architecture may define a communication structure to request internal process data between different services and to inform about state changes. We have derived those parts of the architecture in several steps for our own first approach:

The first step was to analyze business processes in inner-city logistic companies. Analyses were done by interviewing actors in inner-city logistic companiese. g., dispatchers and drivers, about their everyday work. Questions comprise among others for example: Which tools and software are used? Which workflows will be handled and which exceptions could occur?

The result of those interviews are business processes about tour planning, handling orders, managing picking, packing and delivery, and the knowledge about used software and tools.

[3] Econnect Duisburg `www.econnect-germany.de`, enviaM in sMobility `www.smobility.net`, and SAP in iZeus

[4] SmartCityLogistik `www.smartcitylogistic.de`, and PTV Group in iZeus

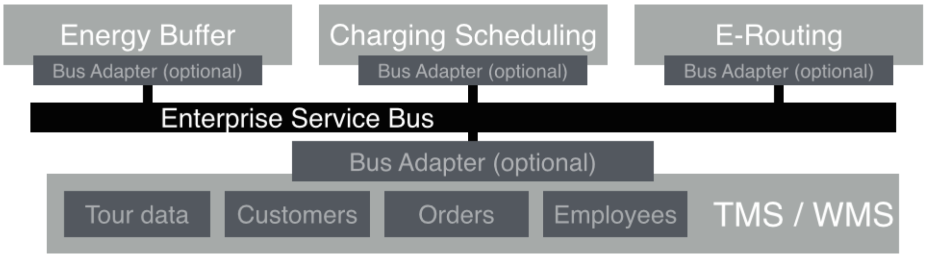


Figure 2. Example of a bus architecture in case of extending a existing TMS/WMS. An optional adapter could be implemented as a proxy in case of missing support.

The analyses about the IT landscape in logistic companies and the interview results were used to create a knowledge base about entities and their relationships in the second step. This knowledge base is context dependente. g., the understanding of a tour can differ in international logistic and inner-city logistic. In our case, the context is based on inner-city logistic. This base leads, by using a suitable modeling method, to a well-controlled vocabulary, which will be used as a base for a data description language (DDL). In case of the SCL project the SCL-DDL covers all terminologies based on their well-structured entities and their relationships.

We got a loosely coupled set of services and funtions with individual dependencies and communciation flows in the third step by using the following ideas: (1) A single IT solution in a logistic company is a subset of functions described in TMS and WMS theory with internal dependencies. (2) Additional functions (services) enable new innovations by using dependencies to internal data represented by functions of TMS and WMS. (3) The union of all functions and services of steps (1) and (2) builds a loosely coupled IT solution for logistic with individual dependencies and communication flows. A communication bus can be used to handle such a set of functions: Large companies use a communication bus, see enterprise service bus (ESB), for arranging a large set of interacting systems. Vehicles uses bus based systems to manage their vehicle systems (see CAN-BUS). By using a domain specific DDL and a standardized communication protocol, the bus can be used to interact between different services and functions.

A TMS/WMS solution could implement the bus feature and an additional service to handle the communication between services. In case of TMS/WMS solutions which do not implement the bus, an adapter may connect a subset of their available interfaces to the bus. However, a single service system must not be specialized to interact with one IT-solution. A single service system only needs an implementation for interactions with the standardized ESB infrastructure. Figure 2 shows how to use that bus in combination with the previously described use cases.

5 Related Work

Several approaches improve the modeling of logistic based architectures or specify the content and meaning of communications between companies. Often, such approaches describe single systems, architectures, or object models to manage situations within a single company instance [3, 4, 6, 7]. The integration of new services would lead to specific extentions.

In the case of standards, RosettaNet and UN/EDIFACT should be mentioned. RosettaNet offers a wide range of specific language, business objects and business processes [8]. United Nations Electronic Data Interchange For Administration, Commerce and Transport (UN/EDIFACT) is one of several international EDI standards. UN/EDIFACT offers a list of specifications for the description of a wide range of documents for e-commerce. This includes documents about routing instructions, orders, purchases, shipments, and billing for example. However, those documents describe data interchanges between companies, but not how to transport them.

6 Conclusion and Outlook

Electric vehicles in inner-city logistic scenarios could be a perfect synergy. Such vehicles are nearly soundless, may have zero emissions, and could help managing the raising number of freight transports without lowering living qualities in urbane areas. Because of range restrictions, missing e-specific infrastructures and long recharging rates, there is a need to handle those restrictions by supporting planning and monitoring processes.

The SCL project manages a system for optimization of plans, monitors, and analyses of tours for electric vehicles. The integration of that system in existent environments is currently an individual implementation which has to be done for each system. Alternatively, the logistic companies may wait for an integration by the producer of their systems, which leads to large delays or high costs in enabling new innovations.

The efforts in reaching one million electrical cars in 2020 leads to a wide range of different e-specific researching projects to help to provide solutions to enable them. Especially in case of inner-logistic it is possible to manage the restrictions of these cars. Unfortunately high acquisition costs and low profit margins will slow down the innovation processes. Reducing complexity and defining more common standards for internal process data in logistic IT-solutions could help to overcome this situation.

References

1. Busch, A., Dangelmair, W., Pape, U., Rüther, M.: Marktspiegel Supply Chain Management Systeme. Gabler, Wiesbaden (2003)
2. Chopra, S., Meindl, P.: Supply Chain Management, vol. 3. Pearson Education, New Jersey (2007)

3. C.Quiroga, N.Koncz, E.Kraus, J.Villa, J.Warner, Y.Li, D.Winterich: Guidance for developing a freight transportation data architecture. NCFRP Report 9 (2011)
4. Dürr, E., Giannopoulos, G., Brugge, R.: Towards an european intermodal freight architecture, ftp://ftp.cordis.europa.eu/pub/telematics/docs/tap_transport/thessalonikivs4.pdf
5. Die Qual der Wahl. Logistik Heute pp. 36–37 (May 2005)
6. Moore, M., Kumara, S., Richbourg, R.: An architecture for logistics replanning. Expert Systems with Applications 11(2), 177–190 (1996)
7. Re, S.D., Campagna, A., Nanni, U.: A reference architecture for freight transport manangement systems, http://freightwise.tec-hh.net/A-reference-architecture-for-freight-transport-management-systems.pdf
8. RosettaNet: Overview - Clusters, Segments, and PIPs (February 2003)

A Domain-Specific Modeling Tool to Model Management Plans for Composite Applications

Oliver Kopp[1], Tobias Binz[2], Uwe Breitenbücher[2],
Frank Leymann[2], and Thomas Michelbach[2]

[1]IPVS, [2]IAAS, University of Stuttgart, Germany
lastname@informatik.uni-stuttgart.de

Abstract TOSCA is a standard to describe composite Cloud-applications and their management in a portable fashion. Thereby, BPMN4TOSCA is a proposed extension for BPMN to ease modeling of management plans. This demonstration presents a web-based modeling tool that supports an updated version of BPMN4TOSCA. The updated version supports direct wiring of data of tasks and events without the need of separate data objects.

1 Introduction

The *Topology and Orchestration Specification for Cloud Applications* (TOSCA [6]) is an OASIS standard for automating provisioning, management, and termination of applications in a portable and interoperable way. To enable this, TOSCA employs two concepts: (i) application topologies and (ii) management plans. An application topology describes the software and hardware components involved in an application and the relationships between them. Management plans capture knowledge to deploy and manage an application and are typically modeled as BPMN or BPEL workflows.

The current OpenTOSCA toolchain starts with Winery [4]. Here, a user specifies required types and models the application topology. Currently, management plans have to be created manually and uploaded as archive into Winery. The whole application package including the topology, required types and the management plans is exported as CSAR and imported in the OpenTOSCA runtime [1]. The runtime deploys the CSAR and adds the application to the Vinothek [2]. Therein, a user can instantiate an application with one click.

In this paper, we present tool support for modeling management workflows in Winery. It is based on BPMN4TOSCA, a domain-specific extension of BPMN to ease the modeling of management plans [3]. BPMN4TOSCA consists of four new elements: The TOSCA Topology Management Task, the TOSCA Node Management Task, the TOSCA Script Task, and the TOSCA Data Object. The TOSCA Topology Management Task is used to interact with the OpenTOSCA runtime. For instance, to retrieve the current service template. The TOSCA Node Management Task is used to invoke management operations on a node. The TOSCA Script Task is used to invoke a script on a node. The TOSCA Data

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

Object provides integration of properties of nodes and relationships as data. If data should be read from or written to a property of a node or relationship, the TOSCA Topology Management can be used or directly a TOSCA Data Object.

We developed a prototypical BPMN4TOSCA-based modeler [3]. In the meanwhile, we improved the OpenTOSCA container to natively support script execution [8] and, therefore, the TOSCA Script Task is obsolete. Furthermore, we discovered that BPMN4TOSCA is still too complex due to directly visible data dependencies; especially if complex application structures have to be handled in non-trivial management plans. Therefore, we reduced the complexity by replacing the TOSCA Data Object, which is the reason for the increased complexity, by a direct wiring of data dependencies between (i) tasks and (ii) properties of the components and relations in the TOSCA model.

First, in BPMN4TOSCA management tasks are modeled by specifying the component and the operation to be invoked on this entity. In the revised version of BPMN4TOSCA, input parameters can be directly wired with output parameters of former tasks. Thus, there is no need for an explicit data object. Although this is a common way to specify data flows, BPMN4TOSCA provides a new means to ease accessing properties of the components and relations described in the application topology model: To specify the input parameter value of an operation invocation, a property of an application component or relation in the topology can be referenced directly. As a consequence, if the component or relation is not instantiated when invoking this operation, the specified property is extracted out of the TOSCA model. If the component or relation is already running, the property is taken from the instance model of the application that holds current runtime information about each component and relation of the application. The same way, properties can be specified for output parameters: After the invocation, the output is written into the specified property of the instance model. Thus, the improved version of BPMN4TOSCA simplifies modeling data flow significantly by enabling a direct access to the instance model of the application, which changes the data flow from a task-centric perspective to a topology-plan perspective.

The extension itself does not follow the idea of data transfer in BPMN. We believe, however, that our extension is in line with the use of BPMN in executable workflows. For instance, camunda BPM also does not use data objects, but uses a global hashmap to store data [7].

Existing BPMN Modeling tools such as bpmn.io[1], Signavio[2], or the web-based modeler of Stardust[3] neither provide a tight integration with application topologies nor allow for a direct wiring of data dependencies. Therefore, we tightly integrated a plan modeler supporting the new version of BPMN4TOSCA in Winery [4], a web-based modeling tool for TOSCA-based application descriptions. In the next section, we describe the modeling of a TOSCA management plan using the presented concept and plan modeler tool. In Sect. 3 we outline next steps.

[1] `http://bpmn.io/`
[2] `http://www.signavio.com/`
[3] `https://www.eclipse.org/stardust/`

2 Modeling a Management Plan in BPMN4TOSCA 2.0

We call the improved version of our management-specific workflow language *BPMN4TOSCA 2.0*. The modeling of a management plan in TOSCA basically encompasses two tasks: (i) specifying the application topology model and (ii) modeling the management plans, for example, a so-called *build plan* that deploys the application. To model the topology, Winery provides a repository of available node and relationship types, for example, an *UbuntuVirtualMachine* node type and *hostedOn* relationship type. These types typically specify available operations, for example, the VM type provides an operation *runScript* to execute low level management commands. Afterwards, management plans can be modeled. To describe management logic, Node Management Tasks specify which operation shall be invoked on which node or relation. Based on that, the input and output parameters are shown. For the input parameter, there is the choice between String, Plan, Topology, Deployment Artifact, Implementation Artifact, and Concat. A string is taken as is as input parameter. When specifying "Plan," all elements preceding the current tasks are queried for their output parameters and offered as choice. In Fig. 1, "Size" of the StartEvent has been chosen. In case "Topology" is chosen, all properties of all node templates and relationship templates are shown for selection. In case of "Deployment Artifact," all deployment artifacts of all node templates are offered and one of them can be chosen. In case of "Implementation Artifact," all implementation artifacts of all node and relationship templates are offered and one of them can be chosen. One use case for the latter two is to specify files to be transferred to a node. For instance, a SQL script to be executed can be chosen here. In case of "Concat," a combination of the other options can be specified. This enables a simple transformation of output variables. For instance, a location path can be appended to the IP address returned by an operation. In the case of output parameters, only Plan and Topology are offered. When choosing "Plan," the parameter is offered as output for subsequent activities. When choosing "Topology," the parameter additionally is written to a property in the topology. Figure 1 shows available properties of an example topology. The current prototype of the BPMN4TOSCA 2.0 plan modeler is available via `http://dev.winery.opentosca.org`. A detailed description of the concepts, the architecture, and the implementation is provided by Michelbach [5].

3 Next Steps

The modeler provides a basis for modeling management plans. The prototype enables us to undertake a user evaluation to quantify the improvements of BPMN4TOSCA 2.0 in comparison to BPMN4TOSA 1.0. When using the modeler in our projects, we did not need complex transformations of the output parameters. An analysis of possible management plans is required to justify the absence of transformation capabilities.

Acknowledgments The research leading to these results has received funding from the German government through the BMWi projects CloudCycle

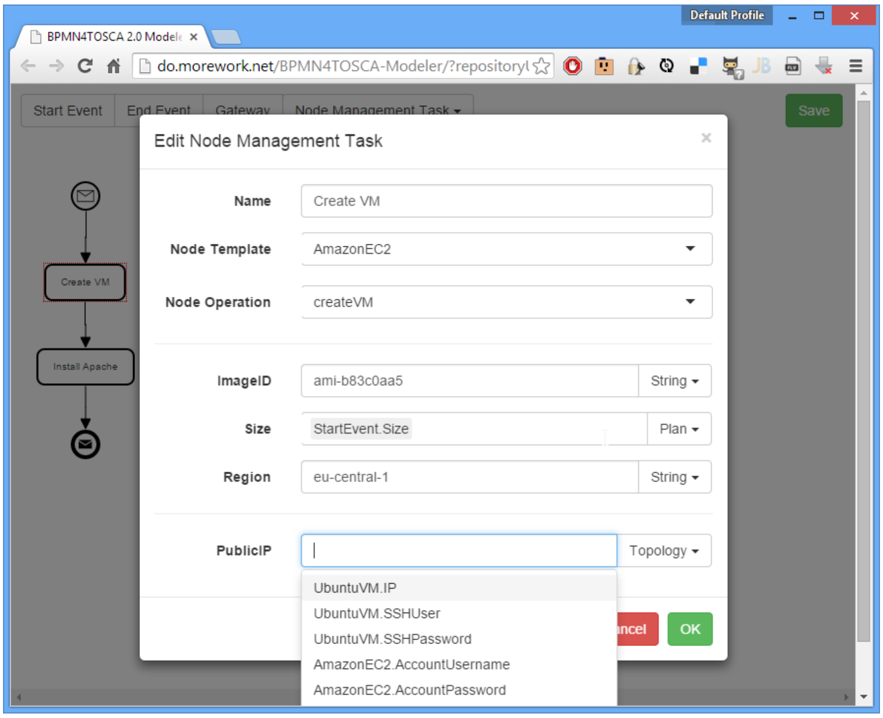


Figure 1. Editing the properties of a node management task

(01MD11023) and NEMAR (03ET401 8B), the BMBF project ECHO (01XZ13023G), and the DFG project SitOPT (610872).

References

1. Binz, T., et al.: OpenTOSCA – a runtime for TOSCA-based cloud applications. In: ICSOC. Springer (2013)
2. Breitenbücher, U., et al.: Vinothek – a self-service portal for TOSCA. In: ZEUS. CEUR (2014)
3. Kopp, O., et al.: BPMN4TOSCA: A domain-specific language to model management plans for composite applications. In: BPMN (2012)
4. Kopp, O., et al.: Winery – modeling tool for TOSCA-based cloud applications. In: ICSOC. Springer (2013)
5. Michelbach, T.: Ein Modellierungswerkzeug für BPMN4TOSCA. Diploma thesis, University of Stuttgart (2015)
6. OASIS: OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) Version 1.0 (2013)
7. Rücker, B.: Data in processes (2013), `https://app.camunda.com/confluence/display/BestPractices/Data+in+Processes`
8. Wettinger, J., et al.: Unified invocation of scripts and services for provisioning, deployment, and management of cloud applications based on TOSCA. In: CLOSER. SciTePress (2014)

Efficient Attribute Based Access Control for RESTful Services

Marc Hüffmeyer and Ulf Schreier

Furtwangen University of Applied Sciences, Germany

Abstract. The popularity of REST grows more and more and so does the need for fine-grained access control for RESTful services. Attribute Based Access Control (ABAC) is a very generic concept that covers multiple different access control mechanism. XACML is an implementation of ABAC based on XML and is established as a standard mechanism. Its flexibility opens the opportunity to specify detailed security policies. But on the other hand it has some drawbacks regarding maintenance and performance when the complexity of security policies grows. Long processing times for authorization requests are the consequence in environments that require fine-grained access control. We describe how to design a security policy in a resource oriented environment so that its drawbacks are minimized. The results are faster processing times for access requests and an easy to manage concept for security policies for RESTful services.

1 Introduction

Many of today's information systems and applications manage huge amounts of users and data. Often users share their own content (e.g. photos, documents) within these applications. A substantial need to control who may access this content is the consequence. In an environment where a lot of users share a lot of data and specify multiple access rights, a flexible, high-performance access control mechanism is required. Because older access control mechanisms do not fit the need for flexibility, research is required how to implement newer and flexible access control mechanism so that high performance can be guaranteed even in complex environments. Processing times in general should be kept small to provide a excellent user experience and therefore processing times for access requests in particular must be kept small.

Attribute Based Access Control (ABAC) may be the next important concept in access control [11]. The main idea behind it is that any property of an entity can be used to determine authorization decisions. The eXtensible Access Control Markup Language (XACML) is a standard that describes how to implement attribute based access control [9]. It consists of three parts: an **architecture** describes multiple components and their responsibilities in the authorization context, a **declaration language** can be used to specify access control policies based on XML and a **request language** to formulate access requests and responses. This work focuses on the declaration language. There are three core elements in the structure of a XACML document: **Rules** describe if an access

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at http://ceur-ws.org

request is permitted or denied. **Policies** group different rules together and **policy sets** group different policies together. Policy sets may contain also other policy sets enforcing a hierarchical composition. Each of these elements has a **target** that describes if the element can be applied to a request by defining a condition. A single access request may be applied to multiple policy sets, policies and rules. In that case those rules may have different **effects** (*Permit* or *Deny*) and a winning rule must be found (based on the structure of the policy). XACML uses combining algorithms for that purpose. An example for a combining algorithm is *PermitOverrides*. It states that an applicable rule with the effect *Permit* will always win against a rule with the effect *Deny*. A full list of algorithms can be found in [9].

Listing 1 shows a simplified version of a XACML policy. For simplification some required information like data types or matching methods (e.g. equals, greater than) are removed. The policy contains two rules and is applicable to a HTTP *GET* request on a resource */users/1/photos*. The first rule grants access to a user identified by an URI */users/2* and the second rule prohibits access for any subject. As one can see fine-grained policies may easily become very complex and performance and maintenance need to be optimized.

```
<Policy CombAlg="FirstApplicable">
  <Target>
    <Resource designator="URI" value="/users/1/photos" />
    <Action designator="HTTP-method" value="GET" />
  </Target>
  <Rule Effect="Permit">
    <Target>
      <Subject designator="URI" value="/users/2" />
    </Target>
  </Rule>
  <Rule Effect="Deny"/>
</Policy>
```

Listing 1. Simplified XACML granting GET access for one user to the photos of another user

2 Related work

XACML computes access decisions at runtime and must evaluate multiple attributes of different categories to find a decision. Therefore the average computation time for an access request increases with growing policy complexity. The problem of computation at runtime is related to the architecture resp. the general concept of XACML. A graph based approach described in [10] tries to address performance issues by changing the processing algorithms. Two different trees are used to evaluate an access request. The first tree identifies applicable rules. The second tree holds the original structure of the security policy and identifies the winning rule. Another approach uses numericalization and normalization to optimize performance [4,5]. Numericalization converts every attribute to an integer value. Normalization converts every combining algorithm into FirstApplicable. In [7] processing time is optimized by reordering policy sets and policies based on statistics of past results. A similar approach to ours also reorders policies based on cost functions but focuses on categories rather than attributes [8]. Also they assume that a rule always is a 4-tuple of a subject, an action, a resource and an effect. Other categories and combinations are not allowed. Declarative authorization for RESTful services is handled in [3]. Attributes are not considered in this approach. In [13] an architecture is described to secure web services (SOAP) based on attributes.

A second major problem of XACML is the modification of policies. XACML does not define how to handle changes to a security policy. The most common way is manually inserting new policy sets, policies and rules supported by a graphical user interface like in [6]. But manually modifying complex policies is very error prone because multiple changes in different branches of the structure may be required. A lot of works exists that addresses the manipulation of XML documents [1,12]. But in this work the security context of XACML is not taken into account. Therefore changes of the security policy are hard to handle.

3 Efficient policy design

An efficient security policy design should enable fast request processing and should be easy to maintain. The security policy described in XACML is a unidirectional graph without cycles. To enable fast request processing we need to consider the costs of processing an access request in a single node of that graph. We define the cost function as:

$$c : N \times Q \to \mathbb{Q}$$

where N is the set of nodes in the security graph, Q is the set of possible access requests and $\mathbb{Q}$ is the set of rational numbers. The set of child nodes of a node k can be expressed as:

$$S_k := \{s \in N \mid \exists p \in path(k, s) \ \& \ length(p) = 1\}$$

Let α_k be the combining algorithm of a node k and let A be the set of combining algorithms. Let ϵ be an effect within the set of effects E. Then one has:

$$\forall \alpha_k \in A \ \exists \epsilon \in E : compute(s_i) = \epsilon \Rightarrow \alpha_k \ stops; \ s_i \in S_k$$

That means that for any given combining algorithm there are one or more effects that cause the algorithm to stop if one of its child nodes computes to one of these effects. For example a node may half two child nodes and the combining algorithm *PermitOverrides*. If the first child node computes to *Permit* the effect of the node will also be *Permit* no matter what the result of the second child node may be. Therefore the combining algorithm stops and should not process the second child node. We define a function γ that describes this behavior:

$$\gamma_k(q, s_i) = \begin{cases} 1 & \text{if } \forall s \in \{s_1, ..., s_{i-1}\} : \alpha_k \ does\ not\ stop \\ 0 & \text{if } \exists s \in \{s_1, ..., s_{i-1}\} : \alpha_k \ does\ stop \end{cases}$$

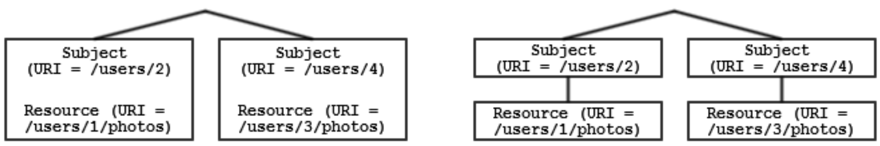
With q ∈ Q and i ∈ $\{1, ..., | S_k |\}$. The cost function c then can be expressed as:

$$c_k(q) = t_k(q) + \sum_{i=1}^{|S_k|} \gamma_k(q, s_i) * c_{s_i}(q)$$

The function $t_k(q)$ describes if a target matches the request q. Therefore it is mainly dependent on how many attributes are specified in the target. Hence, the costs for processing a node depends on the number of attributes in the target, the sum of child nodes and the combining algorithm resp. the order of the children. The following sections describe how to minimize the costs of processing access requests and decrease maintenance efforts for each of the listed factors.

3.1 Target design (minimize t_k)

Attributes should be added carefully to targets to keep the target small and thus reduce the number of comparisons needed to be executed in the worst case. For example a security policy may contain two conditions. Each condition specifies a subject attr. (URI = /users/*userid*) and a resource attr. (URI = /users/*userid*/photos). An intuitive way would be handling each condition in a single target of a rule as indicated in Fig. 1. (a). Processing a request with a subject attr. (URI = /users/2) and a resource attr. (URI = /users/3/photos) requires four attribute comparisons in the worst case because XACML does not specify an order in which attributes must be checked. If a single condition is splitted into multiple targets of rules, policies and policy sets as indicated in Fig. 1. (b) a max of three comparisons is required. This optimization has a benefit for targets that are not applicable to a request while the cost for an applicable rule remains unchanged. Variations in processing times are reduced to a minimum. The maintenance benefit is that it becomes easier to add new conditions that affect a subject with attr. (URI = /users/*userid*) but not a resource with attr. (URI = /users/*userid*/photos).

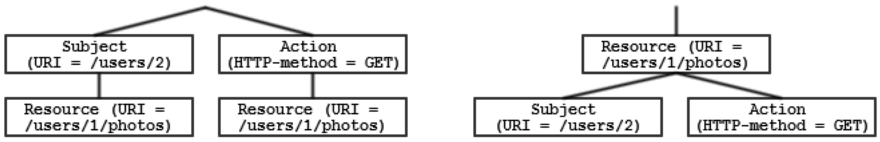


(a) Max. of 4 comparisons (b) Max. of 3 comparisons

Fig. 1. Target design

3.2 Number of sub nodes (minimize $\sum_{i=1}^{|S_k|} c_{s_i}$)

It is obvious that the overall processing time for few nodes is less than the overall processing time for many nodes. Hence, wherever possible targets should be grouped together. That means an efficient policy design must have its branching points at the lowest possible position. Besides the performance gain maintenance efforts for the resource with attr. (URI = /users/1/photos) are reduced because it does not occur twice (cp. Fig. 2.).



(a) Upper branching point (b) Lower branching point

Fig. 2. Number of sub nodes

3.3 Combining algorithm and node order (minimize γ_k)

The selection of the combining algorithm and the child node order also has an effect on performance. Processing those rules first that override the effects of others leads to shorter average processing times. If an overriding rule matches, no other rule needs to be checked. And if no overriding rule matches, the combining algorithm can stop after the first match of the non-overriding rules because they cannot be overridden. This is the basic idea of normalization [5]. Fig. 3 shows the effect of normalization. A given policy with the combining algorithm *DenyOverrides* and two rules as indicated in Fig. 3(a) is transformed so that is has a combing algorithm of *FirstApplicable* and a node order that gives performance improvements. In Fig. 3 (a) both Rule A and Rule B must be processed to find a decision. But for the policy in Fig. 3 (b) it might be enough to process Rule B.

(a) Not normalized (b) Normalized

Fig. 3. Normalization

4 XACML for RESTful Services

One core concept of REST is resource orientation [2]. Therefore we also want to build the security policy based on resources. This is a reasonable technique in a resource oriented architecture and offers the benefit of very fast identification of authorization rules that must be applied during the evaluation process. Therefore targets of policy sets must only contain one resource attribute: the URI. With this constraint it is not necessary to consider combining algorithms since multiple matches of different policy sets or policies are not possible. That means that *FirstApplicable* can be used in every policy set to improve performance. In consequence the processing time for access requests is nearly constant even if new resource paths are added or the security policy is extended.

Another important concept of REST is an uniform interface. Therefore we consider that the set of allowed methods is limited to the HTTP methods. We use these methods as possible actions in a security policy. For each action a new policy should be used right under the policy set for the requested resource. Within these policies rules may be specified that describe under which circumstances the resource may be accessed.

Figure 4 shows an efficient security policy for a RESTful application that follows the optimizations described in the previous sections.

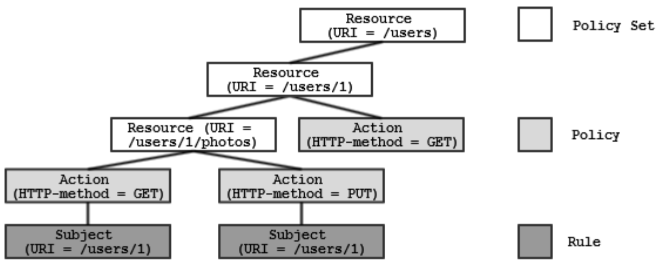


Fig. 4. An example of efficient XACML for RESTful services

5 Results

We performed some first, simple and fragmentary tests on different security policies designed to protect a RESTful service. A first set of policies contains 10 conditions on the URI attribute of 10 resources. For each of the main HTTP methods (GET, POST, PUT, DELETE) we assigned a single policy with one rule, resulting in 40 rules per policy. A second and third set have 100 resp. 1000 more resources resulting in 440 resp. 4440 rules per policy. Each set contains 4 policies: a non-optimized policy (flat structure with a lot of rules in the same policy that follows the pattern of Fig. 1 (a)), a normalized policy with the optimizations described in section 3.3, a structure optimized policy containing the optimizations described in section 3.1 and section 3.2 and finally a policy with all of the optimizations described in section 3 that follows the guideline described in section 4. All policies within a single set are functionally equivalent. We used Balana as XACML engine (https://github.com/wso2/balana). The measurement was executed on a dual core system (Intel i7-3250M, 2,90GHz) with 8GB working memory reserved for the tests. Each test was executed 20 times.

Figure 5 shows the average processing time for an access request. As one can see the processing times for the set with the smallest policies only differ insignificantly. But with growing policy complexity the difference becomes considerably. While the average processing time for the optimized policy remains approximately constant at about 15ms, the average processing time for the non-optimized policies increases up to 304ms. The main contribution to the performance benefit of the optimized policy is delivered by the structure changes as indicated by Fig. 5. Normalization only has a significant impact for larger policies with many rules and without an optimized structure and causes great variations in processing time of up to 200% of the average processing time. The optimized, structure optimized and non-optimized policies have a variation in processing time of about 25%.

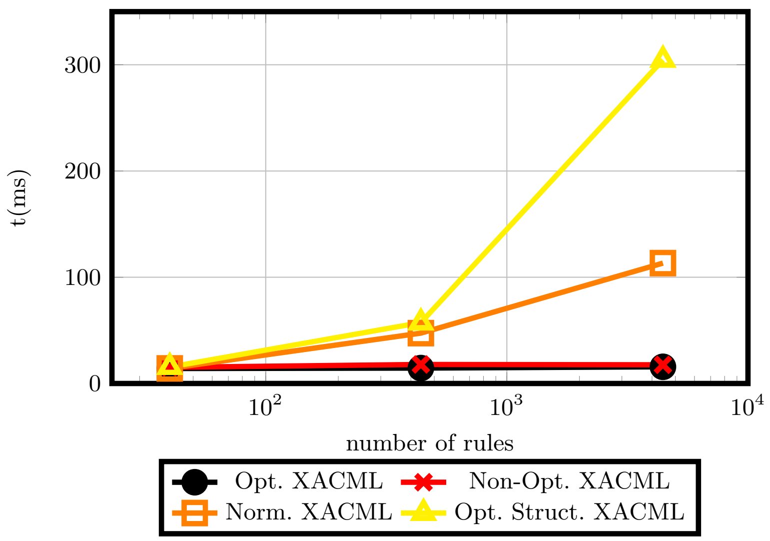


Fig. 5. Average processing time

6 Future work

With every target on a path to a rule access conditions become more restrictive in XACML. This can be a problem for RESTful services. We may have a resource user list *http://example.org/users* and access to this resource is granted only to some administrators but not to single users. But a resource *http://example.org/users/1* may be accessed by administrators and user 1. Since user 1 is a sub resource of the user list, the policy or policy set that handles access to this sub resource should be placed below the policy set that handles access to the user list. In XACML you cannot extend a condition at sub policy level. In consequence the same condition must be repeated multiple times which causes the policy complexity to grow unnecessarily and increases the maintenance efforts.

To handle the maintenance, performance and restriction problems described in the previous sections, we are developing an alternative language similar to XACML. The language targets RESTful services and should guarantuee that only optimized security policies can be written. A draft version already exists and a prototype is implemented. First results show slightly improved performance even to optimized XACML policies. We want to address the maintenance problems with a structured query language that makes it easy to handle changes in a resource oriented context.

References

1. Cubera, D., Epstein, A.: Fast Difference and Update of XML Documents. XTech '99 (1999)
2. Fielding, T.R.: Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine (2000)
3. Graf, S., Zholudev, V., Lewandowski, L., Waldvogel, M.: Hecate, Managing Authorization with RESTful XML. WS-REST '11 (2011)
4. Liu, A., Chen, F., Hwang, J., Xie, T.: Xengine: A Fast and Scalable XACML Policy Evaluation Engines. SIGMETRICS '08 (2008)
5. Liu, A., Chen, F., Hwang, J., Xie, T.: Designing Fast and Scalable XACML Policy Evaluation Engines. IEEE Transactions on Compters, Vol. 60 (2011)
6. Lorch, M., Kafura, D., Shah, S.: An XACML-based Policy Management and Authorization Service for Globus Resources. GRID '03 (2003)
7. Marouf, F., Shehab, M., Squicciarini, A., Sundareswaran, S.: Adaptive Reordering and Clustering-Based Framework for Efficient XACML Policy Evaluation. IEEE Transactions on Services Computing, Vol 4 (2010)
8. Miseldine, P.: Automated XACML Policy Reconfiguration for Evaluation Optimisation. SESS '08 (2008)
9. Organization for the Advancement of Structured Information Standard: eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS Standard (2013)
10. Ros, S., Lischka, M., Marmol, F.: Graph-Based XACML Evaluation. SACMAT '12 (2012)
11. Sandhu, D.: The authorization leap from rights to attributes: maturation or chaos? SACMAT '12 (2012)
12. Wang, Y., DeWitt, D., Cai, J.: X-diff: An Effective Change Detection Algorithm for XML Documents. ICDE '03 (2003)
13. Yuan, E., Tong, J.: Attributed based access control (ABAC) for Web services. ICWS 2005 IEEE International Conference on Web Services (2005)

Privacy-Aware Scheduling for Inter-Organizational Processes

Christoph Hochreiner

Distributed Systems Group, Vienna University of Technology, Austria
c.hochreiner@infosys.tuwien.ac.at

Abstract Due to the increasing specialization of companies in a globalized world, inter-organizational process enactments have become increasingly relevant in recent years. Nevertheless there are hardly any scheduling approaches that meet the requirements of these inter-organizational processes, especially in terms of privacy aspects. In this paper we present a privacy-aware scheduling approach for hybrid clouds, which represents a vital starting point to design a holistic execution environment for inter-organizational process enactments.

Keywords: Cloud Computing, Business Process Management, Hybrid Clouds

1 Introduction

In the last couple of years Business Process Management (BPM) has become a well-adopted approach for companies to provide value-added services to customers [8]. Business processes are composed of software- as well as human-based services and their design ranges from simple sequences to complex structures involving loops, splits or choices [11]. The process enactment is conducted by a Business Process Management System (BPMS) [12] which is considered as a generic software system that manages operational business processes [2]. The management of operational business processes covers the assignment of the different process steps to the designated services which are required to realize a process enactment and schedules their instantiation. Apart from the process scheduling, a BPMS may also manage the provisioning of computational resources to instantiate the software-based services. Since BPMS are used to execute business processes, the BPMS as well as the services are often deployed on fixed resources within the company's premises, where the companies may combine their computational resources to implement a resource pool, i.e. a private cloud [7]. The most important reason for this internal hosting solution are security and privacy restrictions, since services may deal with sensitive information, e.g., health data or execute algorithms that are considered as trade secrets [10].

This paper proposes a privacy-aware scheduling approach for hybrid cloud environments. To obtain an optimal and privacy-aware scheduling respectively resource provisioning approach, we extend the Service Instance Placement Problem (SIPP) [4] which applies Mixed Integer Linear Programming (MILP).

T. S. Heinze, T. M. Prinz (Eds.): Services and their Composition, 7th Central European Workshop, ZEUS 2015, Jena, Germany, 19-20 February 2015, Proceedings – published at `http://ceur-ws.org`

The remainder of this paper is structured as follows: In Sect. 2 we state the motivation for our work and discuss some preliminaries in Sect. 3. We further present our privacy-aware scheduling approach in Sect. 4 and Sect. 5 concludes the paper with an outlook on our future work.

2 Motivation

Business process enactments are usually triggered by process requests. These process requests are issued by external events, e.g., customer interactions, which lead to alternating amounts of business process requests respectively changing resource requirements. In peak-times, when external events issue an extraordinary amount of process requests, a BPMS may run into an underprovisioning scenario, since there are not enough resources to enact the process requests according to their Service Level Agreements (SLAs) [9]. This leads to a lower Quality of Service (QoS), e.g., longer response times and SLA violations may also trigger penalty cost that increase the overall cost for process enactment. Besides the peak-times, a system with fixed resources is also likely to run into overprovisioning scenarios, since the computational resources will not be used adequately. This leads to economically inefficient cost structures for the companies.

Public clouds, e.g., Amazon EC2, offer a promising solution to the resource usage challenges for varying process requests. A cloud-aware BPMS is able to obtain the required resources *on demand* in an utility like fashion. This enables the BPMS to obtain *resource elasticity* by scaling the computational resources up and down, based on the changing requirements. *Measured services* further allow an exact billing of the computational resources based on the actual resource usage [7]. This elastic resource provisioning strategy avoids underprovisioning scenarios, since the public cloud provides enough resources to cover the peak-requirements. A cloud environment also avoids overprovisioning scenarios, because not required resources can be released as soon as they are not needed any more.

Besides the resource allocation there are also other challenges for BPMS, like privacy issues for service instantiations of inter-organizational processes, i.e., service choreographies. Inter-organizational processes are structured similarly to business processes. The major difference is that their process steps are assigned to software services which are provided by different companies instead of only one. Therefore software services for inter-organizational processes can be executed on a community cloud [7]. Nevertheless this common execution environment is not acceptable for some software services due to privacy restrictions. The most promising approach to tackle these issues is the creation of a hybrid cloud which consists of a community cloud and dedicated private clouds for each company [3]. Although resource scheduling for hybrid clouds already raised some attention in terms of scheduling [1] as well as privacy aware deployments [13], there are surprisingly little efforts towards privacy-aware scheduling approaches for BPMSs [6].

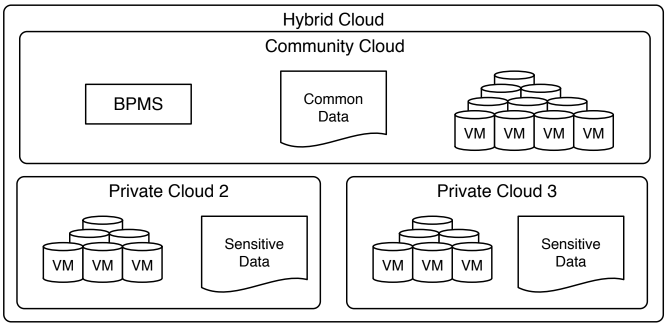


Figure 1: Cloud Landscape

3 Preliminaries

In our previous work we presented the Service Instance Placement Problem (SIPP) [4], which provides a cost-optimized scheduling and resource provisioning plan for multiple parallel process enactments. SIPP represents a multi-objective scheduling strategy which is presented in Sect. 4.1. Up to now the SIPP only considers a single cloud for process enactments. Therefore it does not consider any security nor privacy related aspects which are relevant for process enactments in hybrid cloud environments. Before we describe the privacy related concepts in detail in Sect. 4.2, we define some preliminaries.

The execution environment for inter-organizational processes consists of a a community cloud and dedicated private clouds for privacy sensitive services, as illustrated in Fig. 1. The community cloud hosts all non privacy sensitive services as well as the BPMS. The BPMS schedules process steps, provisions resources for the software-based services on the community cloud and also triggers the deployment of the privacy sensitive services in the dedicated private clouds, based on the privacy restrictions issued for the services. These restrictions are described in detail in Sect. 4.2. In terms of computational resources, we assume that the private clouds offer a limited amount of computational resources, which are only sufficient to run the privacy sensitive services whereas the community cloud offers theoretically unlimited resources.

The inter-organizational processes are composed of multiple process steps that represent the software-based services provided by the participating companies. Fig. 2 represents an exemplary inter-organizational process, which shows the collaboration among 3 companies. Step 3 and 5 are annotated as privacy sensitive and must only be executed in the dedicated private clouds, whereas all other steps can be executed in the community cloud. To execute a process step, the BPMS triggers the deployment of the software-based service on a Virtual Machine (VM) either on the community cloud or on a private cloud. This deployment results in a service instance that can be invoked by the BPMS to execute the service and therefore execute the process step.

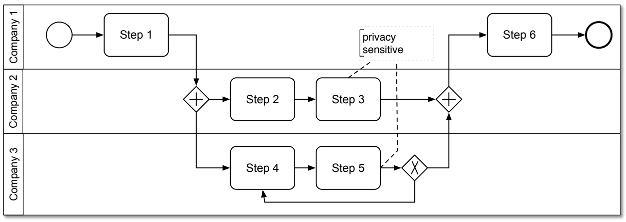


Figure 2: Inter-organizational Process Incorporating 3 Companies

4 Privacy-Aware Scheduling

4.1 Service Instance Placement Problem

The SIPP is represented by a set of different constraints and equations, which are described in detail in [4]. In Eq. 1 the objective for the SIPP optimization model, i.e., the minimization of the overall execution cost, is shown. This objective comprises four terms, where the first term represents the overall leasing cost of the computational resources by summing up the amount of leased VMs $\gamma_{(v,t)}$ multiplied by their cost c_v. The second term shows the penalty cost, which arise, if a process is not finished within the given time. Hereby it sums up all delayed process instances $e^p_{i_p}$ and multiplies them with predefined penalty cost $c^p_{i_p}$. In order to keep the overall cost as low as possible, the optimization model penalizes idle resources (CPU ($f^C_{k_v}$) and RAM ($f^R_{k_v}$)) which are multiplied by the constants ω^C_f and ω^R_f. The last term is designed to prioritize process steps $x_{(j_{i_p},k_v,t)}$, so that steps with a closer deadline DL_{i_p} are executed first.

$$\min \sum_{v\in V} c_v \cdot \gamma_{(v,t)} + \sum_{p\in P} \sum_{i_p\in I_p} c^p_{i_p} \cdot e^p_{i_p} + \sum_{v\in V} \sum_{k_v\in K_v} (\omega^C_f \cdot f^C_{k_v} + \omega^R_f \cdot f^R_{k_v}) - \sum_{p\in P} \sum_{i_p\in I_p} \sum_{j_{i_p}\in J^*_{i_p}} \frac{1}{DL_{i_p} - \tau_t} x_{(j_{i_p},k_v,t)} \tag{1}$$

4.2 Privacy Extensions

Since SIPP is designed for a single cloud, we introduced additional constraints and additional SLA policies to enable the enactment of inter-organizational processes while respecting the privacy constraints of the services.

We extended the set of VM types to distinguish between different deployment locations with their different privacy policies.

$$V = \bigcup_{loc\in Loc} V_{loc} \tag{2}$$

$$K = \bigcup_{loc\in Loc} K_{loc} \tag{3}$$

This differentiation is possible by introducing the identifier loc ($loc \in Loc$), which represents the type of the cloud, e.g., community cloud or 1 of the private clouds. The new set of available VMs is then defined by the union of all VMs, which can be instantiated in the different clouds (Eq. 2). Analogously we also extended the set of all currently instantiated VMs (Eq. 3).

In terms of privacy restrictions, there are 2 specification possibilities to restrict the execution of services regarding the type of the cloud. The first approach is blacklisting: the SLA lists for every process all services, which must not be executed on specific clouds. The major downside of this approach is that the SLA needs to be updated when additional clouds are added.

The alternative approach is whitelisting, i.e., the SLA for each process lists all service instantiation possibilities in the different clouds. Since this SLA pursues a defensive permission approach, there is no need to update the SLA in contrast to the blacklisting approach, when the cloud environment grows by additional private or community clouds. Eq. 4 shows an exemplary SLA for the process presented in Fig. 2.

$$SLA_P = \begin{cases} \text{communityCloud} & \text{(Service 1, Service 2, Service 4)} \\ \text{privateCloud1} & \text{(Service 3)} \\ \text{privateCloud2} & \text{(Service 5)} \end{cases} \tag{4}$$

Based on this SLA for services, a MILP-solver for the optimization problem is able to generate the instantiation possibilities according to Eq. 5. The constraints in Eq. 5 evaluate whether a specific process step j_{i_p} can be instantiated on a specific VM $k_{v_{loc}}$ by querying whether the process step is listed on the whitelist for the given cloud *loc*. If the SLA does not explicitly allow the instantiation of the process step on the specific VM, the constraint rules out the deployment option. Otherwise the MILP-solver decides based on other constraints whether the service is deployed on the specific VM (1) or not (0) as stated in the alternative branch of the constraint.

$$x_{(j_{i_p}, k_{v_{loc}}, t)} = \begin{cases} 0 & \text{, if } j_{i_p} \notin SLA_{P_{loc}}, loc \in Loc \\ \{0, 1\} & \text{, else} \end{cases} \tag{5}$$

5 Outlook

In this paper we focused on the formalization of privacy constraints for the deployment and enactment of inter-organizational processes in hybrid cloud environments. Although these privacy extensions are relevant for process enactment, they only represent a small step towards a holistic process scheduling and resource allocation approach for inter-organizational processes in hybrid cloud environments. Other relevant topics are data transfer aspects among the different clouds, which become increasingly relevant for big data applications or a cost-efficient resource allocation across the cloud environment. In our future work we will evaluate the proposed privacy constraints in a hybrid cloud environment based on the Vienna Platform for Elastic Processes (ViePEP) [5]. Here we plan to evaluate our approach against other privacy ensuring methods, e.g., encryption of privacy-sensitive data in terms of performance and cost-efficiency. Further we will also investigate other areas, like data transfer aspects or pricing policies for hybrid clouds to enable the enactment of inter-organizational processes in an economically efficient manner.

Acknowledgments This paper is partially supported by TU Vienna research funds and supported by the European Union within the SIMPLI-CITY FP7-ICT project (Grant agreement no. 318201).

References

1. van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD). pp. 228–235. IEEE (2010)
2. van Der Aalst, W.M., Ter Hofstede, A.H., Weske, M.: Business process management: A survey. In: Business process management, pp. 1–12. Springer (2003)
3. Goyal, P.: Enterprise usability of cloud computing environments: issues and challenges. In: 2010 19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE). pp. 54–59. IEEE (2010)
4. Hoenisch, P., Schuller, D., Hochreiner, C., Schulte, S., Dustdar, S.: Elastic process optimization - the service instance placement problem. Tech. Rep. TUV-1841-2014-01, Distributed Systems Group, Vienna University of Technology (October 2014)
5. Hoenisch, P., Schulte, S., Dustdar, S., Venugopal, S.: Self-Adaptive Resource Allocation for Elastic Process Execution. In: 6th International Conference on Cloud Computing (CLOUD 2013). pp. 220–227. IEEE (2013)
6. Huang, Z., van der Aalst, W.M.P., Lu, X., Duan, H.: Reinforcement learning based resource allocation in business process management. Data & Knowledge Engineering 70(1), 127–145 (2011)
7. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology (2011)
8. Rosemann, M., vom Brocke, J.: The Six Core Elements of Business Process Management. In: Handbook on Business Process Management 1, pp. 107–122. Springer (2010)
9. Schulte, S., Janiesch, C., Venugopal, S., Weber, I., Hoenisch, P.: Elastic Business Process Management: State of the Art and Open Challenges for BPM in the Cloud. Future Generation Computer Systems 46, 36–50 (2015)
10. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 34(1), 1 – 11 (2011)
11. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14(1), 5–51 (2003)
12. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, 2nd edn. (2012)
13. Zhang, K., Zhou, X., Chen, Y., Wang, X., Ruan, Y.: Sedic: privacy-aware data intensive computing on hybrid clouds. In: Proceedings of the 18th ACM conference on Computer and communications security. pp. 515–526. ACM (2011)

Author Index