Christoph Hochreiner
Stefan Schulte

ZEUS 2016

8th ZEUS Workshop, ZEUS 2016,
Vienna, Austria, 27-28 January 2016
Proceedings

Volume Editors

Christoph Hochreiner
TU Wien, Distributed Systems Group (DSG)
A-1040 Wien, Argentinierstrasse 8/184-1
c.hochreiner@infosys.tuwien.ac.at

Stefan Schulte
TU Wien, Distributed Systems Group (DSG)
A-1040 Wien, Argentinierstrasse 8/184-1
s.schulte@infosys.tuwien.ac.at

Preface

In January 2016, we had the pleasure to host the 8^{th} edition of the ZEUS Workshop in Vienna. This workshop series offers young researchers an opportunity to present and discuss early ideas and work in progress as well as to establish contacts among young researchers. For this year's edition, we selected 10 submissions by researchers from Austria, Germany, Italy, and the United Kingdom that were presented to 20 participants. Each submission went through a thorough peer-review progress and was assessed by at least three members of the program committee with regard to their relevance and scientific quality. The accepted contributions covered the areas of Business Process Management (BPM), Cloud Manufacturing, Stream Processing as well as Medical Support Systems. The workshop program was enriched by the keynote on *From Scientific Process Management to Process Science: Towards an empirical research agenda for Business Process Management* by Jan Mendling, who presented the current scientific state of research in the BPM community and outlined open research areas.

Vienna, February 2016

Christoph Hochreiner
Stefan Schulte

Organization

Steering Committee

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

Organization, Program Committee Chairs

Christoph Hochreiner	TU Wien
Stefan Schulte	TU Wien

Program Committee

Anne Baumgrass	Synfioo – 360° Transportation Monitoring
Claudio Di Ciccio	Vienna University of Economics and Business
Dirk Fahland	Eindhoven University of Technology
Alessio Gambi	Saarland University
Georg Grossmann	University of South Australia
Nico Herzberg	HPI at the University of Potsdam
Christoph Hochreiner	TU Wien
Meiko Jensen	Independent Centre for Privacy Protection
Oliver Kopp	University of Stuttgart
Agnes Koschmider	Karlsruhe Institute of Technology
Matthias Kunze	Zalando SE
Philipp Leitner	University of Zurich
Joerg Lenhard	University of Bamberg
Henrik Leopold	VU University Amsterdam
Niels Lohmann	Carmeq GmbH
Andre Ludwig	Kühne Logistics University
Jürgen Mangler	University of Vienna
Stephan Reiff-Marganiec	University of Leicester
Andreas Schoenberger	Siemens AG
Jan Sürmerli	Humboldt University of Berlin
Matthias Weidlich	Humboldt University of Berlin

Sponsoring Institutions

SAP
T-Systems Austria

Contents

From Scientific Process Management to Process Science: Towards an empirical research agenda for Business Process Management

Jan Mendling

Vienna University of Economics and Business Administration, Austria
jan.mendling@wu.ac.at

Abstract. Business Process Management (BPM) as a research field integrates different perspectives from the disciplines computer science, management science and information systems research. Its evolution has by been shaped by the corresponding conferences series, the International Conference on Business Process Management (BPM conference). As much as in other academic discipline, there is an ongoing debate that discusses the identity, the quality and maturity of the BPM field. In this paper, we will formulate recommendations to further develop BPM research based on the major findings a larger study by Recker and Mendling, which will be published in the Business & Information Systems Engineering journal in 2016. This recent study of the BPM field provides a good basis for discussing how BPM research can be further developed towards a true process science, which will eventually provide insights for practitioners on how to apply *scientific process management*.

1 Introduction

Van der Aalst and Damiani recently observed that the current discussions on data science needs to be extended towards a *process science* perspective [1]. In this paper, we build on a recent study by Recker and Mendling [2] that examines the state of the BPM field based on the proceedings papers of the BPM conference. It is specifically interesting to focus on the BPM conference, because a recent analysis [3] indicated that papers at the BPM conference are somewhat reductionistic in scope, often pursuing either popular problems (such as process modeling languages) or "exotic or even non-existing problems" [3, p.29]. These observations emphasize the need to discuss how BPM research can be further developed towards a true process science, which will eventually provide insights for practitioners on how to apply *scientific process management*.

We proceed as follows. Section 2 presents findings to which extent the BPM lifecycle is covered in recent research. Section 3 discusses to which extent certain research components are utilized in BPM research. Section 4 presents recommendations for future BPM research.

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

2 Coverage of the BPM Lifecycle by BPM Conference Papers

Business Process Management is often described as a lifecycle in order to clarify how different BPM-related tasks fit together. Figure 1 shows a lifecycle with six phases [4]. It also visualizes the coverage of BPM conference papers of each of the phases with a pink dot. An important observation in [2] is that the phases of the BPM lifecycle are covered to a different extent. Most research of the BPM conference is dedicated to questions that are associated with the process discovery and the process implementation phase. Typical matters that are studied in these two pockets are models and modeling languages together with techniques for verification, formal analysis and process mining. The least covered phases are topics associated with monitoring and with redesign.

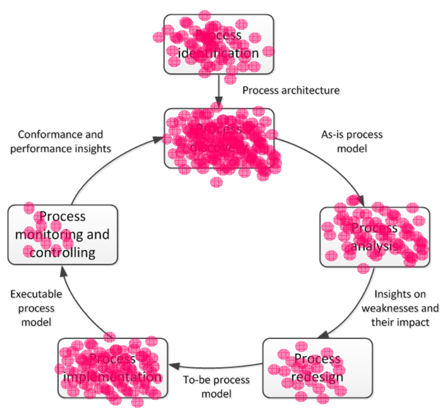


Fig. 1: The BPM Lifecycle and Plotted Conference Papers

3 Research Components

The maturity of the research contributions are arguably linked to the quality of methodological aspects as report in BPM conference papers. Therefore, Recker and Mendling [2] examined whether papers explicitly discuss components of research designs such as variables and hypotheses (for empirical research), or artifact and theory (for engineering and design papers). They observe that, first, the maturity in terms of methodological rigor appears to be a two-sided coin. On the one hand, it appears that engineering papers that report on artifacts and formal concepts are traditionally well-represented at the BPM conference. On the other hand, from the viewpoint of empirical and theoretical work, however, there are only a handful of BPM conference papers explicitly developing hypotheses, and very few stating independent or dependent variables. The share of papers with explicit discussion of theory or hypotheses is also not notably increasing over time. This is a concern, because one would expect that with increasing maturity of research that is presented at a conference, studies would increasingly evaluate and falsify theoretical predictions rather than explore empirical evidence without a priori expectations. This also indicates concerns about the possibility of retroduction as a means of scientific appraisal.

4 Recommendations

Developing BPM towards a true process science requires strengthening the empirical side of BPM including research methods from behavioural science and design science. Based on their analysis, Recker and Mendling formulate the following recommendations [2].

Progressing BPM as Formal Science: It appears that BPM as a formal science is well-represented in the BPM conference series and that it is well-understood by its key contributors. This is, for instance, reflected in the extensive reference to formal Petri net concepts, algebraic definitions and utilization of formal logics in many papers.

Progessing BPM as Behavioural Science: BPM as a behavioural science is concerned with human and organizational behaviours in the context of managing business processes. Such aspects are important for studying, among others, how process knowledge can be effectively documented, which redesign suggestions provide better efficiency, or how processes can be effectively monitored. It appears that there is a need to further strengthen BPM as a behavioural science. Methodological guidelines is available in neighboring fields. The software engineering community has turned to empirical research methods already in the 1980s, most strongly inspired by works of Victor Basili [5]. There has been a growing uptake of experimental research and corresponding methodological guidelines as, for example, summarized in the book by Wohlin et al. [6]. Behavioural research on BPM can benefit from adopting such guidelines from software engineering research.

Progressing BPM as Design Science: BPM as a design science can be considered a third line of inquiry. It perceives BPM as an engineering discipline with the research objective of designing artifacts that provide superior utility in the context of managing business processes. It requires the capability of a researchers to design algorithms and systems, but it also requires empirical research methods [7] for artefact design and evaluation. There appears to be a need for taxonomies to structure the field and the relevant artifacts. This would start with a definition of types of processes [8, p.11] but could expand to a typology of improvement approaches, management techniques or BPM systems.

5 Conclusion

In this paper, we reflected upon BPM research as published in the BPM conference proceedings between 2003 and 2014. Our review of this study focused on the retrospective analysis of research approach, methodological maturity and impact of BPM papers, and we generated a set of varied recommendations for progressing research published at the BPM conference. More details of the study including analyses and recommendations are published in [2].

References

1. van der Aalst, W.M.P., Damiani, E.: Processes meet big data: Connecting data science with process science. IEEE T. Services Computing **8** (2015) 810–819
2. Recker, J., Mendling, J.: The state of the art of business process management research as published in the bpm conference. Business & Information Systems Engineering (2016)
3. van der Aalst, W.: Business process management: A comprehensive survey. ISRN Software Engineering **2013** (2013)
4. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer (2013)
5. Basili, V.R., Weiss, D.M.: A methodology for collecting valid software engineering data. IEEE Transactions on Software Engineering (1984) 728–738
6. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering. Springer Science & Business Media (2012)
7. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly **28** (2004) 75–105
8. Recker, J.C.: Suggestions for the next wave of bpm research: strengthening the theoretical core and exploring the protective belt. Journal of Information Technology Theory and Application **15** (2014) 5–20

A Novel Framework for Visualizing Declarative Process Models

Michael Hanser, Claudio Di Ciccio and Jan Mendling

Vienna University of Economics and Business
michaelhanser@gmx.net, {claudio.di.ciccio, jan.mendling}@wu.ac.at

Abstract The declarative approach to business process modeling has been introduced to deal with the issue of managing flexible processes. Instead of explicitly representing all the allowed enactments of a process, the approach describes the constraints that limit its behavior. However, current graphical notations for declarative processes are prone to be difficult to understand, thus hampering a widespread usage of the approach. To overcome this issue, we present a novel notation framework for visualizing declarative processes, which is devised in compliance with well-known notation design principles.

1 Introduction

Caused by an ever increasing demand for business processes to remain flexible, a declarative process modeling approach seeks to address the issue of current modeling languages lacking support for highly flexible scenarios [13]. Given the fact that a declarative approach is considered less intuitive and tougher to understand [6], a declarative modeling language and notation capable of conveying concepts in a quick and straight-forward manner is necessary. Current state of the art solutions struggle with effectively communicating explicit principles of how to interpret a declarative process model. Owing to the results in existing literature [6,7], a new notation, facilitating understandability and maintainability, is needed. The novel notation outlined in this paper is designed to ease the process of understanding declarative process models. Being developed in compliance with respected notation design principles [12], it offers a set of consistent and explicit mechanisms to effectively communicate semantic constructs. Our framework contributes to existing literature as it builds upon, refines and extends the notation approaches presented in [13,3,4,5].

This paper is structured as follows. Section 2 summarizes Declare and notational design. Section 3 describes the proposed notation and Section 4 discusses its notational quality. Section 5 concludes the paper.

2 Background

In contrast to the widely-used *imperative* paradigm of process modeling, a *declarative* modeling approach does not impose a strict order on activities, but

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

limits their behavior by using constraints. In fact, a declarative model allows any order, repetition or absence of activities, as long as it does not violate the constraints. As each constraint can either evaluate to *true* or *false* during the run time, the state of a process instance is *accepting*, and consequently considered *complete*, if and only if all constraints in the model evaluate to *true*. Declare offers a predefined set of *constraint templates*, each of them consisting of a unique name, a graphical representation and a formal semantic specification in terms of Linear Temporal Logic (LTL) [14,1,2].

Constraints are divided into *(i) Existence constraints*, specifying the cardinality of a task or the first and last activity in a trace; *(ii) Relation constraints*, making an activity's behavior depend on the one of another task; *(iii) Mutual Relation constraints*, which build upon Relation constraints but further cover the converse behavior, i.e. both activities depend on their respective others; and *(iv) Negation constraints*, representing negated versions of Relation or Mutual Relation constraints. *Participation*(a), for instance, is an Existence constraint specifying that activity a must be performed at least *once*. Similarly, *AtMostOne*(a) prescribes that this activity can only be performed either *zero* times or once. Existence constraints are also used to mark the first and last activities in a process instance. *Init*(a) states that task a must be the first activity to be executed in a process instance. Likewise, the constraint *End*(a) indicates a as the very last activity to be performed. *Response*(a,b) is a Relation constraint, which prescribes that activity a must eventually be followed by activity b. Dually, *Precedence*(a,b) imposes that b must be preceded by a. *Succession*(a,b) depicts a combination of the former and the latter, i.e. every activity a must be succeeded by b and every activity b must be preceded by a, thus being a Mutual Relation constraint. These three constraints can be further strengthened by using the *Alternation* and *Chain* limitation. The concept of *Alternate* constraints indicates that the activating task can not reoccur without having the other task executed in between. For instance, *AlternatePrecedence*(a,b) forces activity b to be preceded by a, whilst allowing no further executions of b until a is performed again. Similarly, *Chain* constraints represent an even stricter limitation as they prohibit the execution of *any* other activity in between. E.g., *ChainSuccession*(a,b) forces activity a to be *directly* preceded by activity b and vice versa. Furthermore, certain Relation constraints signify the correlated execution of activities, with no restriction on their temporal order. *RespondedExistence*(a,b), for example, specifies that the execution of activity a also requires activity b to happen at some point in the process. Yet it does not matter whether this is before or after a occurs. Building upon the latter, *CoExistence*(a,b) also includes the converse behavior, thereby implying that the occurrence of a or b always implies the occurrence of one another. Ultimately, Negation constraints are based on existing Mutual Relation constraints, depicting their respective negated form. The *NotSuccession*(a,b) constraint, e.g., states that activity a must *never* be succeeded by b and b must *never* be preceded by a – hence stating the opposite of *Succession*(a,b). Likewise, *NotChainSuccession*(a,b) states that a and b *cannot* occur one after the other, as opposed to *ChainSuccession*(a,b). *NotCoExistence*(a,b) imposes that a and b are *not* allowed to occur in the same trace.

Visual notations such as the one of Declare can be evaluated using Moody's principles of *cognitive effectiveness*, which relate to the speed, ease and accuracy by which the human mind can process a visual notation [9]. Cognitive effectiveness is established as the primary design goal or *dependent variable* for comparing and evaluating visual notations and is thus suitable for making judgements on the goodness of notations. In order to facilitate designing cognitively effective notations, a set of principles is defined relating to the way the *visual vocabulary*, *grammar* and *semantics* should be combined to achieve a good visual notation [12]. In fact, Moody's principles have been demonstrated to positively influence a notation's perceived usefulness [8]. These principles are:

1. **Semiotic Clarity**: semantic constructs have a 1:1 correspondence with respective graphical symbols.
2. **Perceptual Discriminability**: symbols can be clearly distinguished.
3. **Semantic Transparency**: graphical representations suggest their meaning.
4. **Complexity Management**: explicit mechanisms for dealing with complexity exist.
5. **Cognitive Integration**: the integration of information from different diagrams is supported.
6. **Visual Expressiveness**: full range and capacities of visual variables is used.
7. **Dual Coding**: text complements graphical symbols.
8. **Graphic Economy**: the number of symbols is cognitively manageable.
9. **Cognitive Fit**: different visual dialects exist for different purposes.

Various declarative notations have been defined up until now. Van der Aalst et al. propose to visualize declarative models by means of static diagrams that represent the entire process scheme at once [13,3]. Their notation, based on representing Declare constraint templates, has become the de-facto standard for visualizing declarative process models. Even though the notation's visual syntax facilitates a compact illustration of a declarative process model, its semantics tend to be difficult to understand at first sight. Especially when process models increase in size and complexity, as is common in the process mining field, the Declare notation discloses lack of providing a clean and comprehensible overview. Consequently, this increases the mental effort necessary for a user to process and interpret such a model. Given the shortcomings of the original Declare notation [6] in terms of understandability, alternative notations are needed.

3 Notation

Di Ciccio et al. [4] propose a visualization of declarative process models on the basis of Declare constraint templates [13,3] by means of three complementary views: *(a)* the *global view*, depicting a static bird-eye sketch of a process scheme; *(b)* the *local view*, focussing on one activity at a time; and *(c)* the *dynamic view*, visualizing the current state of a running instance. With the notation primarily being devised for representing mined processes of e-mail collections, it is designed to handle larger and more complex process models. However, since the visual elements between these views do not remain consistent, understanding the connection between them can be a tough task.

Building upon the work in [4,5], the new notation employs two corresponding views on a process in a similar manner: *(i)* a static, multi-level *global view*, illustrating the entire process at once and *(ii)* a *local view*, focussing on one activity and its directly related constraints and implications at a time.

The static multi-level *global view* serves as a way of regarding an entire process scheme at once. Within this view the notation provides for different levels of granularity, i.e. we abstract away from various types constraints and merely indicate *positive* or *negative* relations between activities, thus increasing readability at first sight. For the sake of conciseness, this paper focusses on the more detailed "standard" granularity level of the global view. It bases its rationale on a network topology-like alignment of activities, which are accordingly depicted by means of circular elements and complemented by full text identifiers. Relation constraints are embodied by utilizing solid lines and cursors between activities, whereas Existence constraints are delineated by text annotations within the activity element. The notation illustrates constraints prescribing the cardinality of a task, e.g. *Participation*(a) or *AtMostOne*(a), by adding text to the upper half of the circular element. If the constraint specifies the first or last activity of a process, *Init*(a) or *End*(b) respectively, it is indicated by an annotation in the lower left or right part of the element.

A visualized constraint involving a dashed line always implies its belonging to the group of Negation constraints. The notation illustrates Relation constraints between activities by using *solid* cursors for positive constraints and *empty* cursors for Negation constraints, each of them connecting two activities per constraint. Relation constraints are perceived as "if-then" statements: The "if-part" or activation part is complemented by a cursor, being placed pointing either *inwards* or *outwards* of the activating task circle, depending on the sequence-verse of the constraint. This suggests that, if the cursor points inwards, the respective target activity ("then-part") must have been executed before the activation task can be performed. Conversely, if the cursor points outwards, the target activity must happen after the activation task is completed. Applying this rationale to the *Response*(a,b) constraint consequently implies that, since a is the activation task of the constraint, the cursor is placed at this very activity. Moreover, as it specifies that the respective target activity b must eventually be performed afterwards, the cursor is placed *outwards* on the activity border. Contrarily, in order to illustrate the *Precedence*(a,b) constraint, the cursor is now located at activity b, pointing *inwards*. The combination of both constraints, i.e. *Succession*(a,b), is depicted by joining the distinctive elements of their respective graphical representations. Figure 1 illustrates the graphical notation of Declare constraints.

In case a constraint allows executions of further tasks in between, these optional activities are visualized by means of smaller circles and complemented by a Kleene star ($*$), referring to "any other activity". If the constraint does

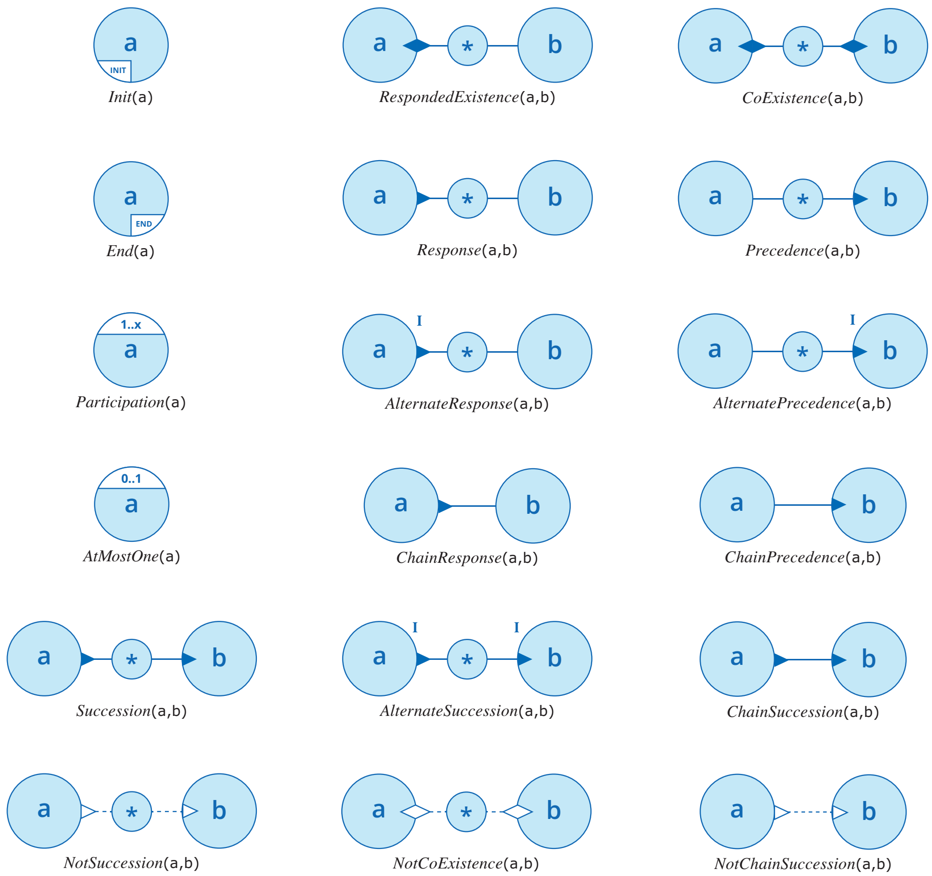


Figure 1. Constraint templates in Declare and their corresponding visual notation.

not prescribe a particular sequence, as in the case of *RespondedExistence*(a,b) or *CoExistence*(a,b), the notation employs two connected cursors, thus forming a diamond, which is placed at the activation part of the constraint. In order to indicate an *Alternate* limitation, the Roman symbol for 1 ("I") is added to the activation part of the constraint. This acts as a counter, stating that this very activity is allowed to only happen once until the other one is performed. Finally, *Chain* variations are depicted by leaving out *optional* activity circles, thereby specifying that no further activity must be performed in between. The process model in Figure 2 depicts an example of the global view.

In contrast to the panoramic global view, the *local view* only focuses on one activity and its *directly* related constraints at a time. As shown in Figure 3, it aims at providing a clear picture of what can, must or must not happen before and after the execution of the examined activity. As the local view's objective is to suggest a possible order of activities, two parameters are taken into account:

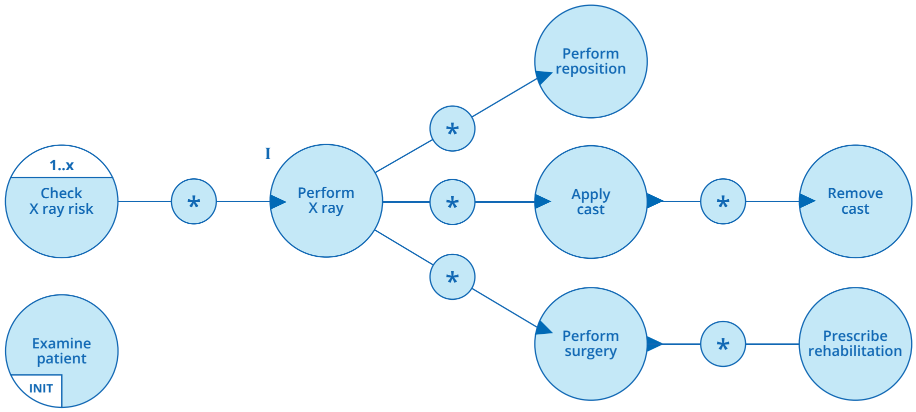


Figure 2. The enhanced global view level of a fracture treatment process.

time and *implication*. Based on the approach in [4,5], its rationale is inspired by the two-dimensional Cartesian coordinate system. With *time* being put on the x-axis, a timeline intuitively leads from left (past) to right (future), while the activity to be analyzed is put at the origin. Pointing from top to bottom, the upper part of the y-axis contains all activities that *imply* the activity located at the center. Conversely, the lower part of the y-axis encompasses all activities that *are implied by* the activity located at the origin.

4 Discussion

This section briefly discusses the implications of our findings with respect to Moody's nine principles [12] of designing cognitively effective visual notations.

The principle of *graphic economy* [12] is applied, i.e., the number of different symbols is being kept as low as possible in order to stay cognitively manageable. This principle explains, e.g., why optional unspecified activities (labeled with $*$) are being illustrated by means of the same geometrical shape as regular activities. The principle of *cognitive integration* [12] motivates the usage of the same set of graphical elements both in the global and the local view. This mechanism supports the integration and enhancement of information from the former to the latter. By employing a rationale with corresponding arrowheads for visualizing "if-then" statements, the notation builds upon using an explicit mechanism for dealing with complexity, as described in the principle of *complexity management* [12]. Moreover, employing this rationale applies to the principle of *semiotic clarity* [12], since a user can easily trace back how each graphic representation is constructed on the basis of its respective semantic construct. Finally, the same principle is considered in delineating *Alternate* constructs as they are indicated by adding a counter of 1, thereby specifying that an activity can be involved once in every alternation. Note that, by contrast, the standard notation of Declare [13] does not exploit explicit principles to increase comprehensiveness.

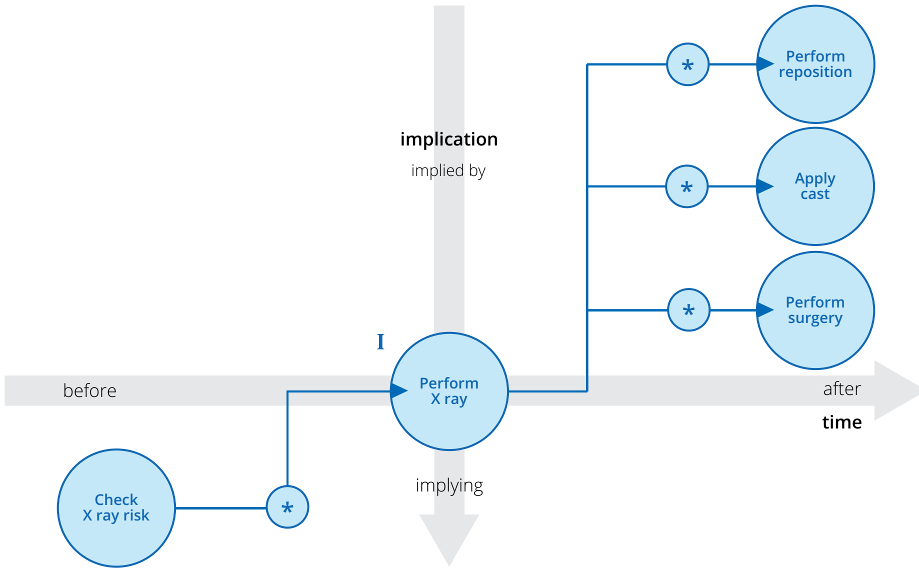


Figure 3. Design rationale of the local view examining the activity perform X ray.

Utilizing circular elements for depicting activities supports the alignment of activities in a more space saving and tidy way, hence enhancing readability and understandability. As cursors can easily be moved alongside the circular border, their connecting lines' bending points can be reduced to a minimum.

5 Conclusion

In this paper, we presented a novel conceptual framework for representing declarative process models on the basis of Declare constraint templates [3]. As this work is only concerned with the design of the notation, future research investigating and evaluating the framework is needed. In the context of process mining, the possibility of scaling the size of activity circles could be used to emphasize reoccurring activities and constraints in a model, as first addressed in [10]. Studies on the guidelines of declarative process modeling could be established, as already existing for imperative languages [11].

References

1. De Giacomo, G., De Masellis, R., Montali, M.: Reasoning on ltl on finite traces: Insensitivity to infiniteness. In: Proceedings of the 28th AAAI Conference on AI. pp. 1027–1033 (2014)
2. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Proceedings of the 23rd international joint conference on AI. pp. 854–860. AAAI Press (2013)

3. van Der Aalst, W.M., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science-Research and Development 23(2), 99–113 (2009)
4. Di Ciccio, C., Mecella, M., Catarci, T.: Representing and visualizing mined artful processes in mailofmine. In: USAB. pp. 83–94. Springer (2011)
5. Di Ciccio, C., Mecella, M., Scannapieco, M., Zardetto, D., Catarci, T.: Mailofmine – analyzing mail messages for mining artful collaborative processes. In: SIMPDA. pp. 55–81. Springer (2011)
6. Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., Zugal, S.: Declarative versus imperative process modeling languages: The issue of understandability. In: Enterprise, BP and IS Modeling, pp. 353–366. Springer (2009)
7. Fahland, D., Mendling, J., Reijers, H.A., Weber, B., Weidlich, M., Zugal, S.: Declarative versus imperative process modeling languages: The issue of maintainability. In: BPM Workshops. vol. 43, pp. 477–488. Springer (2009)
8. Figl, K., Derntl, M.: The impact of perceived cognitive effectiveness on perceived usefulness of visual conceptual modeling languages. In: Conceptual Modeling–ER 2011, pp. 78–91. Springer (2011)
9. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11(1), 65–100 (1987)
10. Maggi, F.M., Bose, R.J.C., van der Aalst, W.M.: Efficient discovery of understandable declarative process models from event logs. In: Advanced IS Engineering. pp. 270–285. Springer (2012)
11. Mendling, J., Reijers, H.A., van der Aalst, W.M.: Seven process modeling guidelines (7PMG). Information and Software Technology 52(2), 127–136 (2010)
12. Moody, D.L.: The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. Software Engineering, IEEE Transactions on 35(6), 756–779 (2009)
13. Pesic, M., Van der Aalst, W.M.: A declarative approach for flexible business processes management. In: Business Process Management Workshops. pp. 169–180. Springer (2006)
14. Pnueli, A.: The temporal logic of programs. In: Foundations of Computer Science, 18^{th} Annual Symposium on. pp. 46–57. IEEE (1977)

SemFrameX – Towards a Framework for the Semantic Justification of BPMN Adaptations

Richard Braun

Technische Universität Dresden,
Chair of Wirtschaftsinformatik, esp. Systems Development, 01062 Dresden, Germany
richard.braun@tu-dresden.de

Abstract. In recent years numerous extensions and adaptations of the BPMN evolved, since model users aim to both exploit the benefits of the modeling standard and adapt BPMN to particular domain peculiarities or project requirements. Methodical support for conducting such adaptations is generally rare and very focused on the abstract syntax, which is actually insufficient, since particular semantics are more relevant. Consequently, it seems to be reasonable to explicitly conduct semantical analysis and comparison checks before extending or adapting BPMN. However, appropriate semantic specifications of BPMN are missing. After introducing and motivating the entire issue, we therefore outline the *SemFrameX* framework that aims to specify the BPMN meta model semantics with a special consideration of ontic, epistemological, conceptual, linguistic and pragmatics aspects.

Keywords: Semantics, Meta Modeling, Extensibility, Process Modeling, Enterprise Modeling Languages, Semantics Framework

1 Extension and Adaptation of BPMN

The Business Process Model and Notation (BPMN) constitutes a prevalent standard for modeling business processes and workflows, which are pivotal parts of enterprises [15]. The level of standardization and application in various domains and projects both in industry and academia leads inevitably to the need for situational extension or adaptation of BPMN [9] in order to enhance, augment [2, p. 51] or specify the language [8]. This implies a particular customization of the BPMN [8, p. 400], which may constitute as dialect [4,10], punctual extension [11] or even as reduced BPMN version [18]. This need for language adaptation is especially caused by the immanent diversity of single domains and enterprises, which factually precludes any *one fits all* approach [24,5]. In contrast to nearly all Enterprise Modeling Languages (EMLs [8, p. 399]) BPMN therefore explicitly provides an extension mechanism aiming to integrate additional meta model elements systematically [31, p. 43]. Further, BPMN as Purpose-Specific Modeling Language (PSML) provides concepts that are explicitly under-specified (cf. [19, p. 136]) in order to enable their specification for respective domains or problems (e.g., *Data Objects* [31, p. 203] or *Pools* [31, p. 306]). While the syntax of BPMN

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

is (widely [30,7]) well-defined, the issue of language adaptation is only supported to a very limited extent implicating both a lack of procedural assistance and semantics [7]. Research on BPMN adaptations merely focuses the syntax and semantic issues are only discussed occasionally [13].

We assume that this is strongly amplified by the general syntax focus of the BPMN specification, which provides only very short and limited semantic references in natural language statements [31]. This might be caused by the rather technical origin of BPMN. Further, parts of BPMN are intended to be executable [31, p. 435] implicating formal behavioral semantics [6, p. 3402], while a range of concepts have material semantics (e.g., *Manual Tasks* or *Pools*). Both types of semantics actually require different kinds of semantic specification (cf. [33]). Due to the stated issues it is difficult to conduct well-justified adaptations of BPMN, since the BPMN specification itself does not provide a solid and detailed semantic base. To the best of our knowledge there is no complete semantic analysis or description of BPMN, which provides respective semantic domain concepts and mappings. Existing research works are either not very mature [13], focus syntactical aspects [30,36] or address only the model layer but not the meta model layer [29]. However, imprecise or even missing semantic specification of EMLs is a general issue that is under-investigated [21, p. 485], [1, p. 108], although semantics seem to be an extremely promising language driver (cf. [4]).

This paper therefore aims to bring light into the dark of semantic specifications in BPMN. Therefore, the semantic issue of BPMN is stated in Section 2. Section 3 then motivates the *semantics first* approach and outlines the *SemFrameX* framework by introducing its dimensions. The paper ends with a short outlook in Section 4.

2 Issues with Semantics

Several authors criticize the lack of semantic specifications in EMLs and emphasize their importance [20, pp. 67-69], [37, pp. 690, 706]. But despite several approaches (e.g., [22,32,26]) no accepted and prevalent standard evolved so far and also the explication of required modeling concepts is still rudimental [19,16]. Instead, both the design of EMLs and EML extensions strongly focus the syntax perspective, while semantics and pragmatics are more or less ignored [4]. In the context of BPMN, we assume that the following reasons may cause the unsatisfying struggles with semantics.

Awareness of relevant parts: There is a lack of consensus about those parts, which determine semantics. The specification of semantics is rather diffuse and remains mostly informal. It is therefore advised to take an integrated view on semantics in regard of the meta model constructs itself as well as the used textual elements.

Formal and material semantics: In contrast to formal domains in the field of Computer Science (CS) [17], the domain of enterprises and business process cannot be completely formalized (material semantics [33]). However, some tasks or purposes require the definition of formal specifications (formal semantics

[17,6]). Respective differences and also integration points should be investigated (hybrid semantics [12]).

Ambiguity: Enterprises are complex socio-technical information systems affecting several aspects – both real-world things and artificial things [26]. This underlines the importance of subjective interpretation depending on personal experiences, cognition and mental conceptualization [27], which is especially relevant within collaborative process modeling in order to avoid misunderstandings, for instance. It is hence necessary to take ontic and epistemological issues into account in order to become aware of its impact and respective consequences for language design and language application.

Multiple research fields: The investigation of semantics is an essential topic in philosophy and cognition research that addresses fundamental epistemological questions. In the CS community, semantics is relevant in the field of Information Retrieval or Semantic Web, for instance. Both areas seem to be relevant in the Information Systems (IS) discipline. However, integration is not trivial due to differing semantic understandings (cf. [38,20]).

3 SemFrameX - Integrated Framework Approach

3.1 Semantically Driven Justification of BPMN Adaptations

It becomes obvious that semantics are crucial in the context of BPMN and also indispensable for BPMN adaptation. We therefore argue that all adaptations should follow a *semantics first* approach in the sense of the following two parts:

Explication of the expected semantics: First, the expected semantics in the sense of required domain concepts should be explicated in order to express objectives and requirements [35,13,16]. The expected semantics are closely coupled with the underlying pragmatic intention [4]. For instance, if the user just aims to document particular real-world aspects then material semantics are applicable (e.g., [2]). If the user intends to enhance BPMN for some automation tasks then formal semantics might become relevant (e.g., [6]). Also *hybrid semantics* as combination of both aspects is imaginable (e.g., in case of clinical decision systems, where supporting processes need to be automated [12]).

Justification of EML adaptation: Afterwards it is necessary to compare the expressiveness of BPMN with the required expressiveness of a particular situation in order to justify and elaborate the need for adaptation based on semantic correspondence checks [1, p. 100]. The comparison finally leads to respective syntactical constructs [13] and might determine the type of extension (e.g., a profile-based BPMN dialect [10]).

3.2 Framework Architecture

As stated in Section 1, semantics in the Enterprise Modeling (EM) context are multi-faceted, not trivial and little considered in literature so far. We therefore aim to tackle the issue by analyzing several dimensions in regard of semantics, which constitutes as the *SemFrameX*[1] framework that is presented in Figure 1. With respect to the limited space of this paper, the architecture is introduced by a brief presentation of the single dimensions.

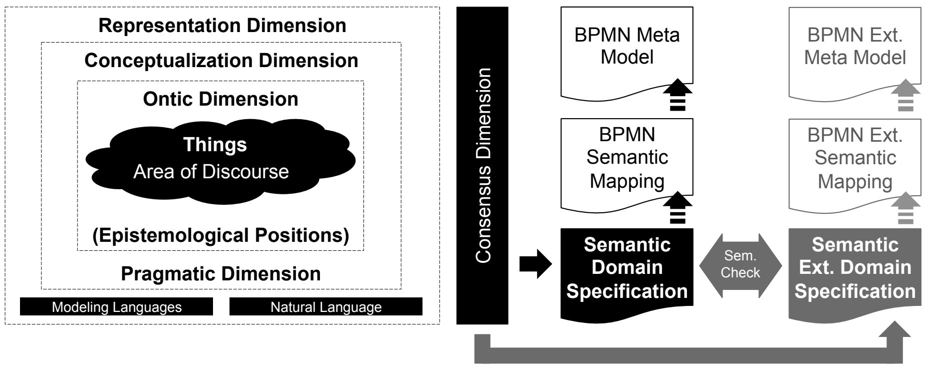


Fig. 1. The *SemFrameX* framework for the multi-faceted specification of meta model semantics and its relevance for language extension.

3.3 Ontic Dimension and Epistemological Position

Fundamentally, any semantic consideration finally leads to an analysis of those *things* that are somehow referred by symbols of a language [39]. It is therefore necessary to analyze and categorize different types of things, e.g. material things or artificial things [26]. As we consider the meta model layer, a thing itself is understood as an already abstracted *class of things* with common features within a particular area of discourse. Hence we consider an implicit abstraction step of modelers, which have to be inferred from single, detectable entities to a class of those entities. More precisely, the fundamental type of respective classes of things in regard of their actual existence in reality (realism vs. idealism) as well as their perception (objectively perceptible vs. subjectively perceptible) have to be considered (adapted from [14,3]). This categorization is important for contingent epistemological positions like Positivism, Critical Rationalism, Radical Constructivism or Methodic Constructivism. Those positions largely determine respective theories of truth, which are especially relevant for the differentiation between formal and material semantics [33] as well as for the identification of differences between conceptualizations of things [25].

[1] *SemFrame* stands for *semantic framework*, while the suffix *X* emphasizes its relevance for extensions and adaptations.

3.4 Conceptualization Dimension

Conceptualization is understood as the individual understanding of the stated class of things. This conceptualization is a central point of analysis, as it depends on the personal understanding of a particular meta model and finally refers to some things s/he has in mind. The only exception is represented by a class of things that is perceived as real and objectively perceptible (Positivism). In each other constellation, the conceptualization is strongly subject-dependent and can cause variant interpretations of meta model constructs by interpreting them differently, for instance [27].

3.5 Pragmatic Dimension

Generally, the application context determines the expected capabilities of a modeling language [38, p. 5] and the concrete modeling purpose plays an immanent role within conceptualization [4, p. 436]. Some authors state that the real meaning of a language finally results from its factual usage [4, p. 438], serving a particular utility [35]. Hence, the underlying or intended pragmatics also influence the aimed semantics. If BPMN is intended to be used for pure documentation then material semantics are relevant, causing a descriptive mode of the semantics (cf. [1]). In contrast, BPMN can be also used to describe (at least partly) automatable processes or message exchange services, which rather cause formal semantics and represent a normative or prescriptive character.

3.6 Representation Dimension

Conceptualizations and things (in case of Positivism) need to be explicated in any form. Ontologies are often proclaimed as means for semantic annotations [22,23]. Basically, even those rather minimalistic languages have a certain syntax and semantics, which have to be taken into account. Further, also the semantics of the meta modeling language used to design the BPMN meta model has to be considered, as it refers indirectly to some artificial things of constructs (e.g., *Generalizations*). These aspects are covered by the framework element *modeling language*. In addition, *natural language* emphasizes the importance of single words (sememe) as basic source of ambiguity. While structural issues are covered by the above mentioned dimensions, natural language based ambiguity is actually the most important issue, since all the stated problems finally lead to lexical topics like synonym and homonym conflicts [34]. For instance, further research on other lexical types like hypernyms, hyponyms, meronyms, holonyms, antonyms and troponyms is needed [22, p. 1628], [38, p. 8], [28, p. 89].

3.7 Consensus Dimension

Despite divergence and ambiguity, it is important to find a particular consensus on semantics in the sense of an agreement of different personal conceptualizations in order to provide an applicable language within a language community. Although

Consensus Theory of Truth is usually applied in Constructivism and Critical Rationalism, we suggest its consequent application, since process modeling usually covers at least some things that are not invariant interpretable. After finding a particular consensus on the semantic specification of either the entire BPMN or prospective BPMN extensions, semantic comparison techniques can be applied in order to justify extension or adaptation need (right side in Figure 1). Currently, we intend to conduct specific ontological comparisons based on generic enterprise ontologies and domain-specific ontologies for this task.

4 Conclusion

This paper presents an overview of a research-in-progress project aiming to elaborate an integrated and multi-faceted semantic description technique for BPMN based on an analysis of ontic and epistemological aspects. The integrated semantic description of BPMN elements should facilitate the justification of potential BPMN extensions or adaptations. Further, the aimed semantic description technique should support the specification of respective requirements profiles, which are the base for comparison with semantics of BPMN. The initial architecture of the *SemFrameX* framework is outlined and its dimensions are briefly introduced.

Further research is manifold, as each dimension has to be investigated in detail. We therefore aim to start with the core of the framework by characterizing and classifying different types of things and respective consequences for their interpretation (formal or material semantics). This is closely related to current investigations in the field of hybrid semantics aiming to specify and integrate both types in BPMN and support the derivation of BPMN extensions and dialects. Furthermore, an inductive application of the proposed architecture to other EMLs seems to be promising.

Acknowledgement: This paper presents results from the research project *SFB Transregio 96*, which is funded by the German Research Foundation (DFG). The author further appreciates the helpful reviewer comments on this paper.

References

1. Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevičius, R., Opdahl, A.L., Panetto, H., Verdecho, M.J.: The unified enterprise modelling language – overview and further work. Computers in Industry 61(2), 99–111 (2010)
2. Atkinson, C., Gerbig, R., Fritzsche, M.: Modeling language extension in the enterprise systems domain. In: Proceedings of the 17th IEEE EDOC. pp. 49–58 (2013)
3. Becker, J., Holten, R., Knackstedt, R., Niehaves, B.: Wissenschaftstheoretische grundlagen und ihre rolle für eine konsensorientierte informationsmodellierung pp. 307–334 (2003)
4. Bjekovic, M., Proper, H.A., Sottet, J.S.: Embracing pragmatics. In: Conceptual Modeling. LNCS, vol. 8824, pp. 431–444 (2014)

5. Bjekovic, M., Proper, H.A., Sottet, J.S.: Enterprise modelling languages – just enough standardisation? In: Business Modeling and Software Design. LNBIP, vol. 173, pp. 1–23 (2013)
6. Bork, D., Fill, H.G.: Formal aspects of enterprise modeling methods: A comparison framework. In: Proceedings of the 47th HICSS. pp. 3400–3409 (2014)
7. Braun, R.: Behind the scenes of the bpmn extension mechanism - principles, problems and options for improvement. In: Proc. of the 3rd MODELSWARD. pp. 403–410 (2015)
8. Braun, R.: Towards the state of the art of extending enterprise modeling languages. In: Proc. of the 3rd MODELSWARD. pp. 394–402 (2015)
9. Braun, R., Esswein, W.: Classification of domain-specific bpmn extensions. In: The Practice of Enterprise Modeling. LNBIP, vol. 147, pp. 42–57 (2014)
10. Braun, R., Esswein, W.: Designing dialects of enterprise modeling languages with the profiling technique. In: Proceedings of the 19th IEEE EDOC. pp. 60–67 (2015)
11. Braun, R., Esswein, W.: Extending the mof for the adaptation of hooks, aspects, plug-ins and add-ons. In: Model and Data Engineering. pp. 28–38. LNCS (2015)
12. Braun, R., Esswein, W.: Towards hybrid semantics of enterprise modeling languages. In: Proc. of the 4th MODELSWARD (2016)
13. Braun, R., Schlieter, H.: Requirements-based development of bpmn extensions: The case of clinical pathways. In: IEEE 1st RE-BPMN. pp. 39–44 (2014)
14. Braun, R., Esswein, W.: Eine methode zur konzeption von forschungsdesigns in der konzeptuellen modellierungsforschung. In: Data Warehousing. pp. 143–172 (2006)
15. Chinosi, M., Trombetta, A.: Bpmn: An introduction to the standard. Computer Standards & Interfaces 34(1), 124–134 (2012)
16. De Kinderen, S., Ma, Q.: Requirements engineering for the design of conceptual modeling languages - a goal-and value-oriented approach. Applied Ontology (2015)
17. Engels, G., Hausmann, J.H., Heckel, R., Sauer, S.: Dynamic meta modeling: A graphical approach to the operational semantics of behavioral diagrams in uml. In: UML 2000 – The Unified Modeling Language, pp. 323–337. Springer (2000)
18. Fernández, H.F., Palacios-González, E., García-Díaz, V., G-Bustelo, B.C.P., Martínez, O.S., Lovelle, J.M.C.: Sbpmn – an easier business process modeling notation for business users. Computer Standards & Interfaces 32(1), 18–28 (2010)
19. Frank, U.: Domain-specific modeling languages: requirements analysis and design guidelines. In: Domain Engineering, pp. 133–157. Springer (2013)
20. Harel, D., Rumpe, B.: Meaningful modeling: what's the semantics of semantics? Computer 37(10), 64–72 (2004)
21. Harzallah, M., Berio, G., Opdahl, A.L.: New perspectives in ontological analysis: Guidelines and rules for incorporating modelling languages into ueml. Information Systems 37(5), 484–507 (2012)
22. Höfferer, P.: Achieving business process model interoperability using metamodels and ontologies. In: Proceedings of the 15th ECIS. pp. 1620–1631 (2007)
23. Kaczmarek, M.: Categories of ontologies' applications in the realm of enterprise modeling. In: Proceedings of IEEE 17th CBI. vol. 1, pp. 98–107 (2015)
24. de Kinderen, S., Ma, Q., Proper, H.A.: Model bundling: Towards a value-based componential approach for language engineering. In: Proceedings of the 8th VMBO (2014)
25. van der Linden, D., Hoppenbrouwers, S.: Challenges of identifying communities with shared semantics in enterprise modeling. In: The Practice of Enterprise Modeling. LNBIP, vol. 134, pp. 160–171 (2012)
26. van der Linden, D., van Zee, M.: On the semantic feature structure of modeling concepts: an empirical study. In: Proc. IEEE 16th CBI. pp. 158–165. IEEE (2014)

27. van der Linden, D.: Personal semantics of meta-concepts in conceptual modeling languages. Ph.D. thesis, Radboud University Nijmegen (2015)
28. Mendling, J., Leopold, H., Pittke, F.: 25 challenges of semantic process modeling. IJISEBC 1(1), 78–94 (2014)
29. Mickevičiūtė, E., Butleris, R.: Towards the combination of bpmn process models with sbvr business vocabularies and rules. In: Information and Software Technologies, pp. 114–121. Springer (2013)
30. Natschläger, C.: Towards a bpmn 2.0 ontology. In: Business Process Model and Notation, pp. 1–15. Springer (2011)
31. OMG: Business Process Model and Notation (BPMN) - Version 2.0. Object Management Group (OMG) (2011)
32. Opdahl, A.L.: Anatomy of the unified enterprise modelling ontology. In: Enterprise Interoperability, pp. 163–176. Springer (2011)
33. Pfeiffer, D., Gehlert, A.: A framework for comparing conceptual models. In: Proceedings EMISA. pp. 108–122 (2005)
34. Pittke, F., Leopold, H., Mendling, J.: Automatic detection and resolution of lexical ambiguity in process models. IEEE Trans. on Soft. Eng. 41(6), 526–544 (2015)
35. Proper, H.A., Verrijn-Stuart, A.A., Hoppenbrouwers, S.: On utility-based selection of architecture-modelling concepts. In: Proc. of the 2nd APCCM. pp. 25–34 (2005)
36. Rospocher, M., Ghidini, C., Serafini, L.: An ontology for the business process modelling notation. In: Formal Ontology in Information Systems - Proceedings of the 8th FOIS. vol. 267, pp. 133–146. IOS Press (2014)
37. Santos, P.S., Almeida, J.P.A., Guizzardi, G.: An ontology-based analysis and semantics for organizational structure modeling in the aris method. Information Systems 38(5), 690–708 (2013)
38. Thalheim, B.: Syntax, semantics and pragmatics of conceptual modelling. In: Natural Language Processing and Information Systems, pp. 1–10. Springer (2012)
39. Ullmann, S.: Semantics: An introduction to the science of meaning (1979)

Maintaining Goals of Business Processes during Runtime Reconfigurations

Budoor Allehyani and Stephan Reiff-Marganiec

University of Leicester
University Road, Leicester, LE1 7RH - UK
{baaa2,srm13}@le.ac.uk

Abstract Business processes have been used extensively to describe how a business achieves its goals; more recently they have also been used embedded into workflow or business process engines to drive the processes. However, business procedures and demands change and consequently the processes need to be adapted. We are building on mechanisms to change processes at runtime and investigate the important aspect of ensuring that the process remains true to its original goal. In this paper we outline our framework and focus on the formal assurance of the goal.

Keywords: Workflows, Business processes, Policies, Goal Consistency, Refinement

1 Introduction

Business processes are used to describe how a business achieves a goal and also to drive processes through business process engines. Each business process is described in terms of a workflow, essentially a set of tasks that are conducted in a specific order and the interaction of those tasks with the environment ultimately capturing how a business goal is achieved. Anton [3] defines goals as "high-level objectives of the business, organization or system"; they capture the reasons why a system is needed and guide decisions at various levels within the enterprise. Business process goals can be defined as rule(s) describing the business outcome and are considered at the development phase to ensure they are met at all levels.

Scientific domains have used workflows to structure and execute processes. While in this work the focus is on business processes, we believe that it has merit in the scientific workflow domain, too. For our convenience we use business process interchangeably with workflow in this work.

Today's businesses operate in a very dynamic environment, where change is almost constantly required due to customer demands, legislation and changes to the business' nature (e.g. mergers) as well as the desire to work more efficiently. These changes have implications on how the business operates and hence on the processes describing how the business goals are achieved. Typically making such changes is a matter of redesigning the processes,thus involving business analysts and then updating the software executing the processes. Gorton et al. [8]

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

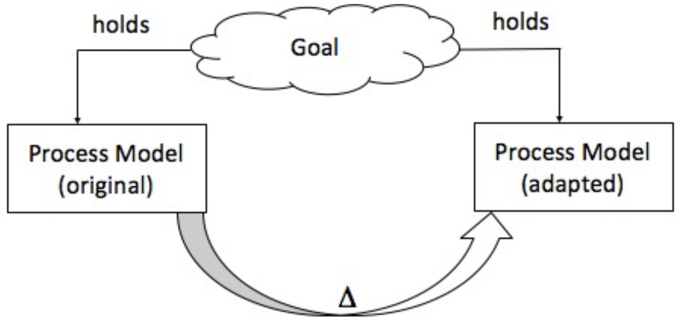


Figure 1. Motivation

presented StPowla which envisioned changes to be made dynamically during process runtime driven by policies describing the rules that one wishes to apply to the process instance. While the previous work captures how the processes are executed and changes are applied it essentially does only provide syntactic guarantees on correctness. However, there is an obvious desire to ensure that the adapted workflow still satisfies its original goal within some sensible range of expectations: no one would want their travel booking process to become one that orders home appliances. More generally, enabling flexibility on workflow systems is a critical challenge in the field of software engineering. Typical issues related to workflow reconfigurations include both syntactical and semantical correctness have been investigated in [5,18,19].

In this work we focus our attention on ensuring consistency in terms of compliance to business goals (note that we only consider functional (hard) goal) . It is of utmost important to ensure that the dynamically reconfigured workflow does what is expected, see Figure 1 where Δ represents the change as defined by the applied policy.

This paper presents our novel contributions (1) the overall framework for goal preserving runtime reconfiguration and (2) the approach to preserving goals.

As an overview, we can consider three different levels for our work, all of which are used at runtime on process instances; note however that obviously the policies and initial processes are specified in a design phase before we execute the process and apply adaptations. Table 1 presents an overview of the levels and their respective inputs and outputs. Specification refers to the original workflow specification including goal and process speciifications, Reconfiguration refers to changing the workflow specification and Verification refers to ensuring that the changed specification does not violate the original goal specification.

The remainder of this paper is structured as follows: Section 2 introduces background work for the formal specification of business processes and their goals as well as StPowla reconfiguration policies. Section 3 presents an overview of our framework and Section 4 presents our example, focuses on the approach ensuring goal compliance. Section 5 presents related work and finally Section 6 concludes on the work and outlines future directions.

Table 1. Overview of runtime levels and languages

	Techniques	Inputs	Outputs
Specification	Wong's prototype tool [20]	BPMN diagram	CSP script
Reconfiguration	Java implementation	CSP + Reconfiguration policies [8]	CSP' script
Verification	Model checking FDR [2]	CSP + CSP' + Semantic properties	Result: Passed/Failed

2 Background

In general, we can look at business processes at several stages of their life cycle; here we assume automatically executed processes only. Typically, we have specifications capturing the design, which are then converted into an executoteable format, which in turn is instantiated and run as and when demanded. If a change is needed this is managed manually at runtime for emergent issues or for more permanent updates by manually redesigning the process. Specifications are supported by a number of languages such as BPMN [1] and there are formal instantiations of such languages in e.g. Petri nets [6] or process algebras such as CSP (Communicating Sequential Process) [20] which allow verification of semantic correctness. We use CSP and its associated tools. Wong [20] provides a formal semantics to (a subset of) BPMN. This work is supported with a prototype tool that converts a BPMN specification into a semantically equivalent CSP specification which then can be further investigated. In their work they focus on deadlock freedom and other generic properties of the process.

Goals of BPMN need to be modelled and formally defined in order to allow for measuring goals achievement [7]. There are plethora of researches in the field of requirement engineering about goal modelling and measuring [10]. KAOS (Keep All Objectives Satisfied) [12] is a requirement specification methodology aimed at supporting requirement elaboration. In KAOS, goal model consists of the strategic goal and its refinement subgoals each mapping to one or several tasks in the process model; these tasks contribute to achieve these subgoals/objectives. Goal modelling in KAOS is declared in two different ways: semi-formal goal structuring model and formal definition in LTL (Linear Temporal Logic). The formal declaration of goals allows for specification in LTL with variant patterns: (1) Achieve goals; the target must eventually occur (desire achievement). (2) Cease goals; there must be a state in the future where the target does not occur (disallow achievement). (3) Maintain goals; the target must hold at all time in the future. (4) Avoid goals; the target must not hold at all time in the future.

In our work we are looking at integrating the change management into the runtime phase by using runtime reconfigurations, which have been proposed in the StPowla [8] framework. Crucially the changes are applied to running instances of a process, so they can make use of data in the instance. As StPowla is meant to operate on running instances of processes the proposed validation needs to fit into the runtime environment. The runtime framework assumes an adapted process execution engine. For simplicity we assume here that we have an engine that can execute BPMN processes directly (this allows us to focus on the main aspects rather than worrying about converting these into some executable formats). The engine is able to pause a process instance and also to make changes to instances.

As the process instance executes it will raise triggers – e.g. at the start of a task which are passed to the policy server (a policy enforcement point) which either returns "no change" allowing the instance to be processed as it is or a specific change action, e.g. the need to insert a task which will lead to updating the process structure of the instance. The action that the policy server demands depends on the policies in the repositories and of course the instance data in the process. The policy server retrieves policies from the policy store, checks for the applicability and then considers the actions to be applied. Once it has determined which actions should be applied the process instance is updated accordingly and would continue executing in its new shape. Through the work presented here an extra phase is added, namely that of checking that the change is appropriate in the sense that it maintains the goal of the original process.

Reichert and Weber [18] analysed typical correctness demands 1) Control flow as well as behavioural correctness, 2) Data flow correctness, 3) Compliance to business rules 4) Instance status compliance and 5) Concurrent change management. Current approaches studied these issues. Semantic correctness in terms of compliance to business rules is addressed in [4] which deals with validating business laws and regulations (compliance rules) as these rules should be met at design as well as reconfiguration levels.

3 Framework for Runtime Goal Assurance – an Overview

As menthioned above, we consider workflow reconfiguration as three level process. The specification and reconfiguration levels was found in [20] and [8] respictively. We integrate these levels into our Java framework and add the verification process. In other words, we implement the reconfiguration policies defined in [8] as Java functions including: i) the insertion of atomic/composite task in parallel/sequence with existing atomic/composite task , ii) the insertion of new branch(s) to existing decision operators and iii) the deletion of atomic/composite task in parallel/sequence with another atomic/composite task. We consider structural correctness in our implementation, e.g. the reconnection of flows when removing atomic existing task as well as inserting new parallel operator when inserting new atomic task in parallel with existing atomic task. For the verification process, we invoke FDR in our framework to check for certain properties (see next section). If the properties hold we allow the change, otherwise we reject it.

4 Approach for Ensuring Goal Compliance

In this work we use the goal concept of KAOS and link it to the process model in order to be able to analyse and reason about the reconfiguration effect. Our verification process benefits from goal modelling patterns (mentioned in section 2) as follows: 1) patterns 1 and 3 allow to ensure the availability of the activities related to the goal when deleting activities from the process and 2) patterns 2 and 4 help to identify the undesired states when inserting new activities.

In KAOS, the strategic goal of the enterprise is refined to different subgoals, see Figure 3, this means that these subgoals are contributing to achieve the main goal. Hence, their representative activities in process model must be available after runtime reconfiguration (as with policy reconfigurations we have the ability to delete activities). Furthermore, the availability of representative activitie(s) for one of these subgoals is not sufficient to ensure the fulfilment of the strategic goal. Note that in the KAOS goal model, these refined goals are connected with AND/OR relations (AND meaning all subgoals are contributing to fulfil the supergoal, OR meaning at least on of the subgoals must be achieved). So, we have two types of check when considering delete policies: 1) the availability of activities contributing to achieving subgoals and 2) the availability of all activities which together fulfil goals. If we consider insert policies we have to check that the new tasks belong to the domain and relate "somehow" to the goal model.

We can also establish a link that preserves/allows these activities from/to changes at policy level. We could call this link a control or management link.

For operational purposes we use CSP tools, so we define the semantic constraints in CSP as well as rules for satisfaction of these constraints. As we want to preserve semantics in the adaptive system, we have the original workflow (source) and reconfigured workflow (target). Our semantic check is based on activities (tasks) type annotation.

So, taking the source (P) and target (P') workflows which are represented in CSP as processes together with the goal specifications G1,....,Gn we wish to check if the property specification refines the goal specification. Such refinement explores the dependency between processes in the process model and the goals in the goal model. Our properties are (informally) defined below:

1. Let X be a type annotation for the tasks which contribute to achieve the goal. All processes of type X in P must be available in P'.
2. Let T be the type annotation for tasks in the relevant domain. All processes in P' must be annotated with one or more type from T.

Here are the steps towards linking goal model to process model: 1. Declare formally the goals/ subgoals using the goal specification patterns, 2. Convert these specifications into CSP specifications, 3. Establish management link between goals and processes. The first step to achieve that is to annotate the contribution activities, 4. Declare our properties, which specify what we want to avoid when change policy affects process model, 5. Define the refinement relation (satisfaction function): Spec |= G; which indicates that the property specification "Spec" satisfies the goal specification in question "G".

Considering our admission example depicted in Figure 2, we suppose that the main (strategic goal) is to assign the right student to the right place. Then, this goal is refined to operational objectives which are related to activities in process model. For example, the activities `Get_GTA`, `Get_Attainment` and `Get_English_Test` are contributing to fulfil the subgoal `RequirementsMet`. Considering this example shows that a block of activities contribute to achieve a

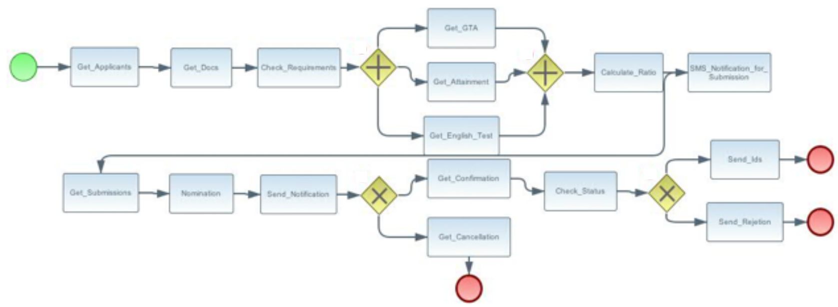


Figure 2. The process model "BPMN" for university admission

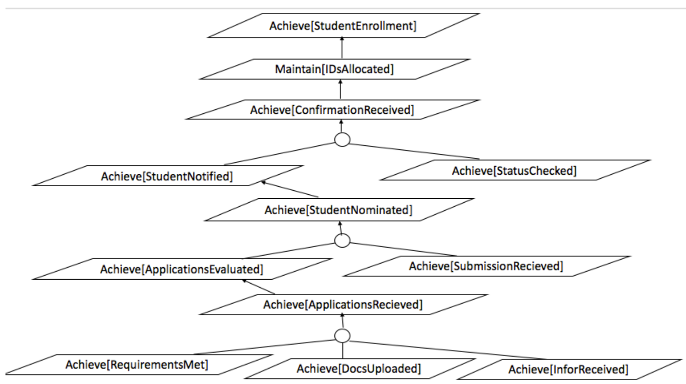


Figure 3. KAOS goal model for university admission

single subgoal.Therefore, we need a CSP assertions that check the fulfilment of each subgoals to ensure the overall achievement of the strategic goal. For verification we consider CSP trace and refusal trace refinement [16] as we are interesting in the availability of events. A process trace is a sequence of all events the process can execute. This helps us to formulate our assertions as we check the trace for certain (annotated) processes.

5 Related Work

In the literature, workflow adaptation is a twofold issue: 1) providing flexible workflows by adding/deleting/skipping tasks, changing the order of tasks and rolling back failed tasks and 2) ensuring correctness and consistency; it is inevitable to ensure the robustness of adaptive workflows. The robustness of workflow systems includes correct control and data flows, consistent behaviour and consistent semantics. Change can take place at different levels as classified in [9]. Most of the current approaches consider process change at design time handling the change either manually [17], or semi-automatically [14], [15]. StPowla [8] provides a promising solution towards flexible workflows as it handles changes (online) on running instances and automatically through reconfiguration policies. StPowla considers two types of policies, refinement and reconfiguration policies. Briefly, refinement policies specify requirements on the service that can be chosen to execute a task while reconfiguration policies make changes to the workflow structure. It is the latter that are the focus of this work. Reconfiguration policies allow at the most fundamental level for insertion or deletion of tasks in the process, which can easily be extended to inserting/deleting sub-processes and changing operators.

Koliadis and Ghose [11] have studied how the process model is affected when adapting the goal model by establishing (traceability and satisfaction) links between process model (in BPMN) and goal model (using KAOS). We study the effect of process reconfigurations but when adapting the process model.

Ly et al. [13] provide a definition for semantic constraints and their satisfaction in adaptive Process Management Systems (PMS). Their approach based on integrating application knowledge into adaptive PMS, which allows to define semantic constraints based on this knowledge. This work differs from our work as we consider consistency in terms of adhering to the main goal at an abstract level, while they address it at data level considering tasks incompatibility.

We believe that the criteria from [18] are insufficient to ensure workflow robustness as they do not guarantee that the workflow will still adhere to its original specification. We add 6) compliance to business goal as an important semantic constraint when adapting workflows as an additional requirement and the work presented here is a step towards allowing assurance of that.

6 Conclusion and Outlook

Much work has been considering adaptation of workflows; typically this is done manually and not at runtime of process instances. This has the disadvantage that it cannot react to data in the instance and also that it usually requires human intervention. We have previously presented an approach [8] that can dynamically adapt running instances – however that leaves a problem of ensuring that the process still works towards the business goal. In this work we have presented our approach for ensuring such goal compliance for runtime reconfigurations based on formal refinement checking. We are considering semantic consistency against high level specification, taking into account structural correctness. Nevertheless, workflows are knowledge intensive systems that carry and exchange data during execution and it is of utmost important to guarantee data flow correctness but it is outside the scope of this paper. Note that there is mapping from BPEL, the BPMN execution language, to CSP [21] and it is therefore possible to model check data properties through FDR. Future work will investigate the formulation of more generic assertions that can be validated automatically.

References

1. Object management group business process model and notation, http://www.bpmn.org
2. Oxford university computing laboratory, http://www.cs.ox.ac.uk/projects/concurrency-tools/
3. Antón, A., et al.: Goal-based requirements analysis. In: Requirements Engineering, 1996., Proceedings of the Second International Conference on. pp. 136–144. IEEE (1996)
4. Awad, A., Decker, G., Weske, M.: Efficient compliance checking using BPMN-Q and temporal logic. In: Proceedings of the 6th International Conference on Business Process Management. pp. 326–341. BPM '08, Springer-Verlag, Berlin, Heidelberg (2008)
5. Casati, F., Ceri, S., Pernici, B., Pozzi, G.: Workflow evolution. Data Knowl. Eng. 24(3), 211–238 (Jan 1998)
6. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in bpmn. Inf. Softw. Technol. 50(12), 1281–1294 (2008)
7. Edirisuriya, A., Zdravkovic, J.: Goal support towards business processes modelling. In: Innovations in Information Technology, 2008. IIT 2008. International Conference on. pp. 208–212. IEEE (2008)
8. Gorton, S.: Policy-driven Reconfiguration of Service-targeted Business Processes. PhD thesis, University of Leicester (2011)
9. Han, Y., A. Sheth, A., Bussler, C.: A taxonomy of adaptive workflow management. Workshop of the 1998 ACM Conference on Computer Supported Cooperative Work (1998)
10. Kavakli, E.: Goal-driven requirements engineering: modelling and guidance. Ph.D. thesis, University of Manchester (1999)
11. Koliadis, G., Ghose, A.: Relating business process models to goal-oriented requirements models in kaos. In: Advances in Knowledge Acquisition and Management, pp. 25–39. Springer (2006)
12. Lapouchnian, A.: Goal-oriented requirements engineering: An overview of the current research. University of Toronto p. 32 (2005)
13. Ly, L.T., Rinderle, S., Dadam, P.: Semantic correctness in adaptive process management systems. Springer (2006)
14. Müller, R., Greiner, U., Rahm, E.: Agent work: A workflow system supporting rule-based workflow adaptation. Data Knowl. Eng. 51(2), 223–256 (Nov 2004)
15. PANG, S., Yin, L., Hua, H., Ch, L.: A model for dynamic business processes and process changes. Chinese Journal of Electronics 20(4) (2011)
16. Phillips, I.: Refusal testing. Theoretical Computer Science 50(3), 241 – 284 (1987)
17. Reichert, M., Dadam, P.: Adeptflex—supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems 10(2), 93–129 (1998)
18. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer Science & Business Media (2012)
19. Rinderle, S., Reichert, M., Dadam, P.: Correctness criteria for dynamic changes in workflow systems: A survey. Data Knowl. Eng. 50(1), 9–34 (Jul 2004)
20. Wong, P.: Formalisations and Applications of Business Process Modelling Notation. PhD thesis, University of Oxford (2011)
21. Yeung, W.: Csp-based verification for web service orchestration and choreography. Simulation 83(1), 65–74 (Jan 2007)

3D Printing Process Pipeline on the Internet

Felix Baumann and Dieter Roller

Institut für Rechnergestützte Ingenieursysteme
(baumann|roller)@informatik.uni-stuttgart.de

Abstract. We develop a small and lightweight cloud based service for the utilization of 3D printer resources within an academic context. This service consists of user, artefact and printer management and utilizes existing business process management systems (BPMS) where possible and extends their functionality. It enables scheduling of printing jobs for artefacts and high utilization of 3D printer resources. This cloud based manufacturing (CBM) system enables 3D printers that are non-native networked to be used remotely by providing easily installable low cost networked computers. It focuses on the interface between the physical resources and their representation in software to form a cyber physical system (CPS). This service requires smart 3D printers and representation of technical capabilities of physical resources. We discuss the design and concept of this work in progress service and the distinctions from similar systems.

Keywords: 3D Printing, Additive Manufacturing, Cloud Based Service, Cloud Based Manufacturing, CBM, CPS

1 Introduction

3D Printing or Additive Manufacturing (AM) is the process of creating physical objects from digital models usually layer upon layer [5]. Technologies for AM include Fused Deposition Modelling (FDM, trademark by Stratasys Inc., also Fused Filament Fabrication FFF), Laser Sintering (LS), Electron Beam Melting (EBM), Laminated Object Manufacturing (LOM), Stereolitography (SLA) and Electron Beam Freeform Fabrication (EBF). Every AM technology brings restrictions on the materials possible to process. We focus our research on FFF where thermoplastics like acrylonitrile butadiene styrene (ABS) or polylactid acid (PLA) are fed from a roll in filament form to a heated extruder that heats the plastic to a semi-molten state and extrudes it through a nozzle mounted on the printing head that is moveable in two dimensions (X-Y plane) by electro motors following a pre-programmed path (Toolpath). With this setup it is possible to trace contours and interiors of an object slice-wise. After completion of every layer the printing bed is moved in Z-direction so the following layer can be added on top. For the generation of the toolpath (slicing) it is necessary to segment the original digital model into slices that can be analysed for tool movement along the contours. Various strategies exist for the generation of the toolpath as models

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

are mostly created hollow with a specific infill pattern for reduction of weight and processing time. The initial focus on FFF technology does not limit this research to just this technology as the 3D printing process is the same with alterations due to technology used and parameters adapted. It is our understanding that the following reasons mandate the use of printing services over stand-alone 3D printers at the user's workplace:(a) High cost of printer (dependent upon manufacturer and technology) [16] (b) Potential health risks (e.g. fumes, metal dust) [15] (c) Low utilization for non-shared resources [12] (d) Process knowledge necessary for high quality results [10]. The 3D printing process consists of five steps (Fig. 1) that start with the design of the product (also see [5]). For this

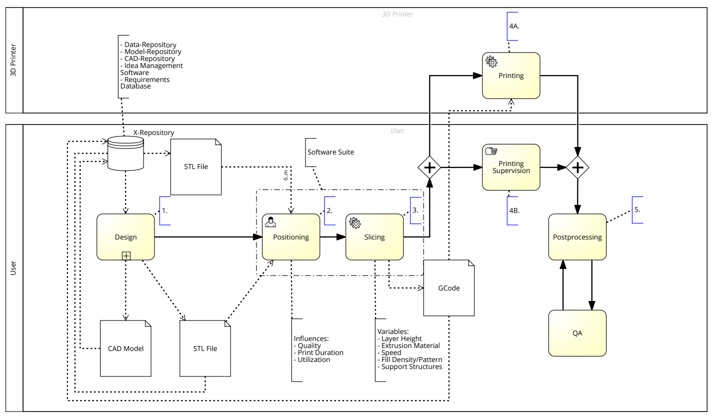


Fig. 1. 3D Printing Process

work we propose two research questions: (a) What requirements are necessary to construct a 3D printing service enabling users utilizing existing 3D printer resources more efficiently (b) How can a 3D printing service be enabled to provide an infrastructure for research.

This article describes current work in progress and outlines the design and implementation considerations and methodology. The design phase can be supported by software using CAD (e.g. Autocad[1] or 3D modelling software[2]). The result of this first step is a CAD model that represents the 3D geometry of the object.

Step two of this process is the positioning of the model in the virtual space that represents the 3D printer and its physical restrictions. Positioning can encompass single objects or multiple objects for increased printer utilization. After the print object is positioned it is sliced using slicer software. A variety of slicing software

[1] http://www.autodesk.com/products/autocad/overview
[2] https://www.rhino3d.com

exists and they differ in aspects like speed, precision, quality and strategies for printing support structures.

The following steps include the upload to the printer if it is a networked device or other means like deployment on memory devices (e.g. SD-Card, USB Stick) and the start of the print which can either require manual interaction or be handled from software. During printing the user is often required to supervise the printing progress as this is error prone especially for consumer grade devices. Post-processing and Quality Assurance (QA) follow when the object has been printed and influence each other. Those steps are not part of our service.

We provide support for all steps but the design, post-processing and QA step within our service. These are omitted for the following reasons 1) The design process is supported by specialized software and integration is not compatible with our lightweight approach 2) Post-processing and QA is not reasonable supportable by soft- or hardware as these steps require intensive human interaction.

The remainder of this article is organized as follows: We display current research in this area in [Sect. 2] and derive implementation requirements from established approaches. Then a introduction of the implementation guidelines [Sect. 3] for the service is given. Following is a summary of requirements [Sect. 3.1] for our research. Then we introduce an example and discuss problems encountered with the implementation [Sect. 3.2]. In [Sect. 4] we discuss our approach, its application and benefits.

2 Related Work

Similar systems or services already exist in form of closed source commercial services where we will name two of: a) 3D Hubs[3]b) 3D Printer OS[4]. As commercial entities their focus is on financial viability. These services allow adding ones own 3D printer and manage it from within the service with a varying degree of granularity. They lack an extension mechanism or plug-in architecture. In contrast to our approach they are not intended as open services. The software octoprint[5] offers remote printing and object management capabilities but does not provide an interface to a BPMS, user-selectable slicing solutions or support for consolidated information on printing information. Further research provides proposals from [18] for CBM systems but our system differs from those approaches as our focus is the tight integration of business process management (BPM) and 3D printing as well as the sensory upgrade of this technology. From Dong et al. [3] we will implement the video supervision approach for the printing process and its remote error detection. Extensions of CBM in the form of Cloud Based Design and Manufacturing [19] provide further insight into the concept of Hardware-as-a-Service (HaaS) and the connection to the broader concept of flexible manufacturing spanning every phase of product development and involvement of different stakeholders. While the availability of affordable

[3] https://www.3dhubs.com
[4] https://www.3dprinteros.com
[5] http://octoprint.org

consumer grade 3D printers certainly has helped the progression of research in and distribution of 3D printers the scenario where every individual will own a digital fabricator [9] is debatable as the general direction is to offer and consume services [2]. Van Moergestel et al. [8] proved the concept of Manufacturing-as-a-Service (MaaS) on cheap, distributed and reconfigurable production machines (equiplets) with a focus on interaction in a multi-agent system. Lan [6] names STL viewers as Java applets or other visualization tools as one of the key issues in his review. We employ JavaScript embeddable visualization into the service as to alleviate the dependency on thick clients. Further key issues e.g. a) Remote control and monitoring b) Job planning and scheduling and c) RP data pre-planning are addressed in our service.

3 Implementation

Our service follows the software framework proposed by Schulte et al. [14] with a focus on the action executioner. It acts as the connector between the printing resources and the printing service in our proposal in contrast to the proposed functionality by Schulte et al. Further foci are the service registry for keeping information on production capabilities and the monitoring data manager that connects the real execution in the 3D printer with the virtual representation. From CloudMan [11] we incorporate the layered service approach but restrict our focus to 3D printers and not manufacturing infrastructure in general. See Fig. 2 for overview of the intended architecture with BPMS supporting the main service controller.

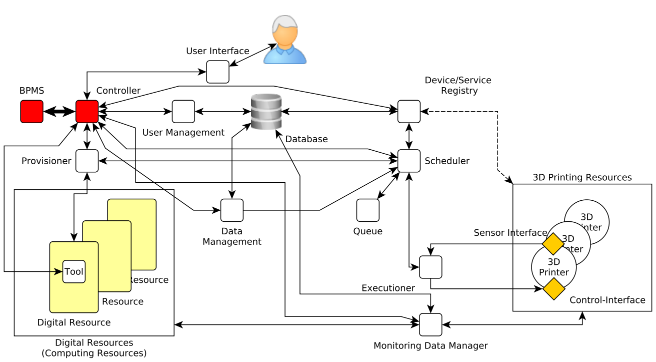


Fig. 2. Abstract Architecture for 3D Printing Service

As per the definition of NIST (SP 800-145) [7] of cloud computing the system is set up to provide a user management system by incorporating available libraries. Besides standard user management information the user is able to store appropriate files[6] in his account. For this we define an interchange format for printing related information consisting of original CAD file(s), resulting STL and GCode [4] files, conversion and printing protocols as well as imagery (for quality assessment). The service is accessible in a standard compliant web browser that supports HTML 5[7] and JavaScript, both are necessary for rendering purposes for the phase of positioning. The resources necessary for slicing and preparation of the models are shared amongst the users based on a scheduling scheme that reflects first-come first-serve. As the 3D printer is the limiting resource at present the pooling of the computing resources not regarded as critical. In anticipation of multiple 3D printers controlled by the system the distribution of computing resources for the preparatory tasks is becoming queue based with data stored in associated cloud service storage (e.g. Amazon S3[8]). Users will be informed if the capacity of the 3D printers is depleted and the projected processing time for an object exceeds a defined threshold. The requirement for “rapid elasticity” is severely impaired by the physical restrictions set by the geometry of the object to be printed and the limitations in the speed vs. quality trade-off of a 3D printer. Basic measurements are intended where the user can track the number and nature of printed objects as well as associated information and a full audit trail for research purposes. Utilization of machines and computing resources is measured and associated with respective user accounts. The systems control layer resides in the cloud and is expandable by utilizing proven technology (e.g. Docker[9]) as means of deployment. The interfacing layer consists of gateway computers that interface directly with 3D printers if they do not support network access natively. These interface solutions depend on rapidly deployable, cost sensitive and reliable computer systems. In the first phase these interfaces will allow direct manipulation of 3D printers via the Internet and limited control information backflow. Further iterations extend this system to a broader sensorial back channel ultimately leading to closed loop printing systems.

3.1 Requirements

The requirements for a CBM provided by Wu et al. [17] “(R1) [...] (R2) Should provide cloud-based distributed file systems that allow users to have ubiquitous access to manufacturing-related data (R3) Should have an open-source programming framework that manufacturing systems can process and analyse big data stored in the cloud (R4) Should provide a multi-tenancy environment where a single software instance can serve multiple tenants (R5) Should be able to collect real-time data from manufacturing resources (e.g., machines, robots, and assembly lines), store these data in the cloud, remotely monitor and control these manufacturing resources (R6) Should provide IaaS, PaaS, HaaS, and SaaS applications to users (R7) [...] (R8) [...]” are considered in the design of our system.

[6] CAD files, STL files, Printing Log files
[7] http://www.w3.org/TR/html5
[8] http://aws.amazon.com/s3
[9] https://www.docker.com

Due to the design goal of developing a lightweight system the requirements R1, R7 and the focus on academic settings requirement R8 are not incorporated in our service. We further define the requirements a) Capability to use BPMN extension for 3D printers (which is proposed separately) and integration of a BPMS b) Modular integration of tools for the 3D printing process and c) Modular and dynamic integration of 3D printing resources.

3.2 Example and Problems

We encounter the problem of defining capabilities of various 3D printers for use in this service. To our knowledge such a description format or language does currently not exist. Resource Description Language (RDL [13]) is a proposition for this issue for the domain of network embedded resources. Capabilities required for interaction with tools includes a) GCode dialect b) Quality settings c) Processing speed and d) Material capabilities. This information is also required for utilization planning and optimization strategies. As a solution for this problem we propose a derivative RDL tailored towards additive manufacturing for subsequent publication. Further problems arise from the firmware of our research printer that limits the transmission speed (ca. 3.5 KiB/s) over the USB serial connection to the device storage resulting in long transmission times. Solutions include flashing a different firmware and utilizing WiFi enabled SD cards.

To clarify the flow of information (see Fig. 2) and data within our proposed service we discuss this by an example of a user printing an object. The first process steps of designing and modelling the object with a CAD or modelling tool are not discussed and we assume the user, which already has an account within the service, logs in and has an AutoCAD DXF[10] file stored on his computer. As a first action the file is uploaded through the web-interface to the controller that instructs the data management service to store the file in the database, then the file is transformed into STL and AMF [1] format for future use and stored in the database. The user then selects a printer for printing. This information is provided by the device/service registry. Future implementations can suggest an appropriate printer to the user. After selecting the printer the user is able to select slicing parameters and position the object in the virtual build environment. Future implementations can suggest appropriate parameters based on analysis of the model file and positioning on optimization criteria to the user. The user is able to add other tool steps to the processing of the object file which are orchestrated by the provisioner and associated virtual computing resources. After the model file is sliced the printing job is instantiated with the scheduler that checks if the requested printer resource is available and if so sends it to the executioner. If not a queue is used to store the job until the resource becomes available. The executioner communicates with the control interface of the 3D printer in order to transfer and start the print. Sensor data is transmitted back to the executioner from the sensor interface. Sensor data is then stored in the database via the scheduler and the controller. BPMS support is intended as to

[10] Drawing Interchange Format

externalise the logic of the controller to a BPMS that orchestrates the other services. During the print the user is informed on the progress and possible failure of the print via web interface. After completion of the print the user is informed through a notification. Data acquired during the print is stored in the database for later analysis.

4 Future Work

To the best of our knowledge no open source 3D printing service is published yet. There are existing solutions that focus on separate parts and provide solutions to different aspects of the 3D printing process. Our approach is characterized and differs from other approaches by: a) Focus on 3D printer b) Focus on communication with manufacturing device c) Interface to BPMS d) Platform for testing BPMN extension e) Smartifying 3D printer f) Platform for testing sensor array and g) Interchange format for print related information. This software service is designed as an open research platform for academic users to embed experiments and utilize distributed resources. Further projects are aimed at 1. providing means of control of 3D printers from within process models as we are writing an BPMN extension, based on the work of [20] tailored for 3D printers, 2. utilize sensors for print status observation and as a means for quality research into 3D printing (see ICRM 2016[11]). Those projects are to be incorporated in the umbrella project described in this work. As a related project we develop a BPMN extension for 3D printer integration into BPMN where the hardware resources and data flows can be modelled using the extension. This extension is out of scope of this work and published separately.

References

1. ASTM ISO: ASTM52915-13, Standard Specification for Additive Manufacturing File Format (AMF) Version 1.1. Tech. rep., West Conshohocken, PA (2013)
2. Banerjee, P., Friedrich, R., Bash, C., Goldsack, P., Huberman, B., Manley, J., Patel, C., Ranganathan, P., Veitch, A.: Everything as a Service: Powering the New Information Economy. Computer 44(3), 36–43 (2011)
3. Dong, B., Qi, G., Gu, X., Wei, X.: Web service-oriented manufacturing resource applications for networked product development. Advanced Engineering Informatics 22(3), 282–295 (2008)
4. ISO: 6983-1:2009 Automation systems and integration - Numerical control of machines - Program format and definitions of address words. Tech. rep. (2009)
5. Kulkarni, P., Marsan, A., Dutta, D.: A review of process planning techniques in layered manufacturing. Rapid Prototyping Journal 6(1), 18–35 (2000)
6. Lan, H.: Web-based rapid prototyping and manufacturing systems: A review. Computers in Industry 60(9), 643–656 (2009)
7. Mell, P., Grance, T.: SP 800-145. The NIST Definition of Cloud Computing. Tech. rep., Gaithersburg, MD, United States (2011)

[11] http://icrm-aachen.com

8. van Moergestel, L., Puik, E., Telgen, D., Meyer, J.J.: Implementing Manufacturing as a Service: A Pull-Driven Agent-Based Manufacturing Grid
9. Mota, C.: The Rise of Personal Fabrication. In: Proceedings of the 8th ACM Conference on Creativity and Cognition. pp. 279–288. C&C '11, ACM, New York, NY, USA (2011)
10. Petrick, I.J., Simpson, T.W.: 3D printing disrupts manufacturing. Research Technology Management 56(6), 12 (2013)
11. Qanbari, S., Zadeh, S.M., Vedaei, S., Dustdar, S.: CloudMan: A platform for portable cloud manufacturing services. In: Big Data (Big Data), 2014 IEEE International Conference on. pp. 1006–1014 (Oct 2014)
12. Ruffo, M., Tuck, C., Hague, R.: Cost estimation for rapid manufacturing - laser sintering production for low to medium volumes. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 220(9), 1417–1427 (2006)
13. Santos, A.C., Pedrosa, L.D., Kuipers, M., Rocha, R.M.: Resource Description Language: A Unified Description Language for Network Embedded Resources. International Journal of Distributed Sensor Networks 2012, 11 (2012)
14. Schulte, S., Hoenisch, P., Hochreiner, C., Dustdar, S., Klusch, M., Schuller, D.: Towards Process Support for Cloud Manufacturing. In: Enterprise Distributed Object Computing Conference (EDOC), 2014 IEEE 18th International. pp. 142–149 (Sept 2014)
15. Short, D.B., Sirinterlikci, A., Badger, P., Artieri, B.: Environmental, health, and safety issues in rapid prototyping. Rapid Prototyping Journal 21(1), 105–110 (2015)
16. Wohlers, T., Gornet, T.: History of Additive Manufacturing. Wohlers Report 2014 24 (2014)
17. Wu, D., Rosen, D.W., Wang, L., Schaefer, D.: Cloud-based Manufacturing: Old Wine in New Bottles? Procedia CIRP 17, 94–99 (2014)
18. Wu, D., Rosen, D.W., Wang, L., Schaefer, D.: Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. Computer-Aided Design 59, 1–14 (2015)
19. Wu, D., Thames, J.L., Rosen, D.W., Schaefer, D.: Towards a Cloud-Based Design and Manufacturing Paradigm: Looking Backward, Looking Forward. In: Computers and Information in Engineering Conference, Parts A and B, pp. 315–328. ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (2012)
20. Zor, S., Schumm, D., Leymann, F.: A Proposal of BPMN Extensions for the Manufacturing Domain. In: Proceedings of the 44th CIRP International Conference on Manufacturing Systems (2011)

Process Engine Benchmarking with Betsy – Current Status and Future Directions

Matthias Geiger, Simon Harrer, and Jörg Lenhard

Distributed Systems Group, University of Bamberg, Germany
{matthias.geiger,simon.harrer,joerg.lenhard}@uni-bamberg.de

Abstract. Business process management and automation has been the focus of intense research for a long time. Today, a plethora of process languages for specifying and implementing process models have evolved. Examples for such languages are established international standards, such as the Web Services Business Process Execution Language 2.0 or, more recently, the Business Process Model and Notation 2.0. Implementations of these standards which are able to execute models, so called process engines, differ in the quality of service they provide, e.g., in performance or usability, but also in the degree to which they actually implement a given standard. Selecting the "best" engine for a particular usage scenario is hard, as none of the existing process standards features an objective certification process to assess the quality of its implementations. To fill this gap, we present our work on process engine benchmarking. We discuss what has been achieved so far and point out future directions that deserve further investigation.

Keywords: business process management, process engine, BPEL, BPMN, benchmarking

1 Introduction

The field of business process management (BPM) forms an umbrella for a variety of research areas, ranging from managerial challenges to application engineering [1]. Among these fields are the modeling and automation of processes using process-aware technologies specifically dedicated to this task. This has led to the development of a multiplicity of process languages and standards [21] that can be used for specifying process models. A subset of these languages allow to specify models that are intended for execution in specific runtime environments, called *process engines*. Typically, multiple alternative engines are available for a given process language. The user of the language can implement process models as defined in the language specification and has to select the best-fitting engine for execution. Naturally, a variety of properties can form the basis for this selection, such as pricing, performance, usability, or actual language support.

The problem in this setting is that it is hard for a potential user to meaningfully judge these properties for a given set of engines, due to the inherent complexity of such software products. In general, this selection problem is not new, and exists

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

in similar fashion for any sufficiently sophisticated software tooling or technology, such as application servers or ERP systems. To make such a decision, there are a plethora of methods available [29], one being the analytic hierarchy process (AHP) [25]. But to apply these methods, the properties of the different alternatives need to be known. One technique to reveal these properties is *benchmarking* [26], which in this case resolves to *process engine benchmarking*. The enabling of the benchmarking of state-of-the-art engines for widely used process standards and for a comprehensive set of quality properties is the long term goal of our work. To this end, we are developing the BPEL/BPMN engine test system (*betsy*), which implements a comprehensive benchmark for process engines[1]. The development of *betsy* is in progress for more than three years already and, by now, more than a dozen engines in a variety of revisions are integrated in a fully automated and reproducible benchmarking process.

In this paper, we briefly discuss related approaches for process engine benchmarking in Sect. 2. Next, we detail the current status of *betsy* and how it has evolved since its first public release in 2012 in Sect. 3 and, in Sect. 4, how we plan to evolve *betsy* even further in the future. The paper is summed up in Sect. 5.

2 Related Work

Benchmarking of IT products is not a new phenomenon and therefore there exists already lots of related work regarding this topic (e.g., [5, 11, 18, 27]). Particularly interesting is [18] which defines general requirements to be fulfilled by benchmarks to be a valid, "good" benchmark: Overall, a benchmark should measure relevant aspects to be able to give substantial answers to the investigated research questions. Benchmarking workflow engines is a relevant topic as there are no certification authorities to check claimed compliance promises. So each vendor can claim that his product is BPMN [19] or BPEL 2.0 [22] conformant without the need to actually prove it. Moreover, also other questions are relevant for users of BPM products such as: ease of installation, portability and conformance to statical analysis rules which also can be compared for different products. However, [18] lists other requirements to benchmarks which might be conflictory to the aspect *relevance* as a benchmark should also be *repeatable*, *fair*, *verifiable* and *economical*. As betsy focuses on standard's conformance testing those four requirements are fulfilled: betsy is Open Source and fully automated which allows for *repeated* test execution. Moreover, (most) tested engines are freely available and directly integrated into our approach, which allows every interested party to execute the tests without *economic* barriers on standard developer hardware. As the standard documents define all relevant aspects to be fulfilled by the implementing engines and we are building upon the same documents, betsy does not give an advantage to some engines but is *fair*. Due to the openness of the standard texts and our implementation the correctness of betsy is open to scrutiny fostering the *verifiability*.

[1] The tool itself is available at `https://github.com/uniba-dsg/betsy`.

Apart from those general works, there are some approaches regarding process engine benchmarking, which are more closely related to our work. In [4] BPEL 2.0 engines are assessed regarding their performance using the SOABench testbed. Another approach dedicated to performance benchmarking of workflow engines is the BenchFlow[2] project which focuses on benchmarking BPMN 2.0 engines [8, 23, 28]. Their latest work [8] evaluates the performance of two anonymized open source BPMN 2.0 engines within a container-based environment. By using container-based environments, the authors follow the recommended approach to achieve reproducible research and benchmarks [5]. Directly reusing the concepts and artifacts generated by those two approaches is not useful for the scope of our tests, as measuring performance needs a far more complex infrastructure apart from the actual engines under test to generate sensible workloads and to ensure the validity of the results [18]. Our tool betsy should be able to reproduce the results without economical and technological barriers, i.e., it should be executable on standard developer machines without any complex installation and configuration steps. However, as both approaches are automatically executing tests on workflow engines at least the usage of virtualization techniques such as virtual machines (e.g., with Oracle VirtualBox[3]) or using containers (e.g., with Docker[4]) to store and restore working engine installations is also relevant for our work.

A third notable approach [7] presents a method to evaluate BPM systems (BPMS) with the aim of selecting the best fitting BPMS for a list of requirements. In a series of case studies, the authors evaluate a large list of open source and proprietary BPMS implementing three different process languages (e.g., the XML Process Definition Language (XPDL) 2.2 [31], BPEL 2.0 [22] and BPMN 2.0 [19]). In contrast to our work, this evaluation is on a more abstract level and the actual engine evaluation is not automated.

3 The Current State of betsy

Betsy 2.1.0, the most recent version, has been published on September, the 29th in 2015[5]. The tool is freely available and licensed under the LGPL v3. Currently, it is capable of benchmarking three BPMN engines in thirteen different versions with 135 tests and seven BPEL engines in 16 different versions, two of them also in an in-memory configuration, with 1110 tests.

The current state of *betsy* can be described according to four dimensions: 1) process languages, 2) process engine capabilities, 3) process engine types and 4) process engine environments. The dimension *process language* is reflected in the *betsy* acronym. Although the acronym never changed, its meaning has evolved from *BPEL engine test system* to *BPEL/BPMN engine test system*, since *betsy* is able to evaluate process engines implementing the process languages BPEL [22] or BPMN [19]. The dimension *process engine capabilities* describes

[2] See `http://www.iaas.uni-stuttgart.de/forschung/projects/benchflow.php`
[3] See `https://www.virtualbox.org/`
[4] See `https://www.docker.com/`.
[5] See `https://github.com/uniba-dsg/betsy/releases` for all releases.

Table 1. Status quo of *betsy* according to its four dimensions

Dimensions	Dimension Characteristic	Publications
Process Languages	BPEL 2.0	[12, 13, 15–17, 20]
	BPMN 2.0	[9]
Process Engine Capabilities	Feature Conformance	[9, 12, 13]
	Static Analysis Conformance	[16]
	Expressiveness	[13]
	Robustness	[15]
	Installability	[20]
Process Engine Types	Open Source Engines	[9, 12, 13, 15–17, 20]
	Proprietary Engines	[13]
Process Engine Environments	Bare Metal Environment	[9, 12, 13, 15, 16, 20]
	Virtual Environment	[17]

the features of the process engine that are tested. At first, *betsy* was used to evaluate feature conformance, but over time it was extended to assess the static analysis conformance, expressiveness, robustness, and installability of process engines. The third dimension *process engine types* investigates which type of process engine is put under scrutiny, being either an open source or a proprietary process engine. The last dimension *process engine environments* refers to the ability to benchmark the process engines in a bare metal environment or a virtual environment, such as in a virtual machine or a container.

4 Future Directions

To support a more meaningful selection of process engines, we aim to extend *betsy* to a process engine benchmarking platform, making it faster, more flexible, powerful, and extensible. Our plans are detailed along the four dimensions.

Dimension process language: The field of process standards is vast [21] and in constant evolution. The relevancy of a process engine benchmarking system depends on the relevancy of the language it supports. Currently, *betsy* supports BPEL 2.0 [22] and BPMN 2.0 [19]. Arguably, these two languages are sufficient at the moment, since there is no competing standard that equally targets process engine execution. XPDL [31] is also a process standard that allows for the specification of executable process models, but it is primarily meant as an interchange format. Although it is used as execution format in some engines, it is expected to be replaced for this purpose by BPMN 2.0 [6]. Therefore, there is no reason to include XPDL in benchmarking directly. Furthermore, academic approaches to process languages, such as Yet Another Workflow Language (YAWL) [2], do exist. However, YAWL is neither standardized, nor do competing implementations of YAWL, apart from the reference implementation, exist. As a result, there is no selection or comparison problem and no reason to consider the language.

Dimension process engine capability: For BPEL 2.0 engines, *betsy* already covers a large variety of engine capabilities [12, 13, 15–17, 20]. With the emergence of BPMN 2.0, we have started to benchmark the feature conformance of

BPMN 2.0 engines as well [9]. Our current goal is to fill in open gaps by benchmarking BPMN 2.0 engines for the same set of capabilities which we benchmarked for BPEL 2.0 engines, including static analysis conformance, expressiveness, installability, and robustness. The challenge here is how can the BPEL 2.0 benchmarks be ported to BPMN 2.0, effectively reusing the benchmarks to some extent. And interesting aspect is the statical analysis conformance, i.e., do perform the engines statical analysis of models as defined in the specification. Whereas the BPEL standard [22] is directly listing relevant statical analysis checks this is not the case for BPMN. As shown in preliminary work [10] this raises issues for BPMN modeling tools which are also to be expected for BPMN engines.

In addition, it would be desirable to increase the set of already covered engine capabilities by also benchmarking performance. Performance has always been an important criterion for software selection and evaluation [30]. In a preliminary work, we evaluated existing benchmarking approaches of BPEL 2.0 engines [24] and revealed that most of them test a very small number of engines, use a limited workload model and only focus on mostly one or two metrics. Moreover, as stated in [24] for BPEL, additional challenges arise as the process engines do not support the same set of features. The same holds true for BPMN engines as well. Hence, either the benchmark's workload can only be executed on a few engines or it must be reduced to using only the features that all engines support. Apart from extending betsy, our current results can be used to improve the related work presented in Sect. 2: The conformance results of betsy can be used to determine a sensible workload leading to a benchmark which produces fair and reproducible results for all or at least the most important engines. What is more, existing test suites, e.g., of the control-flow pattern, can be used as workloads for micro-performance benchmarks. Thus, this area calls for further investigation.

Dimension process engine type: The market of process engines can currently be separated into proprietary and open source engines. In academic research, the usage of open source tooling is much more common, due to a more permissive access that does not involve costs. As a result, most analyses of process engines focus primarily on open source engines, e.g., [9, 12, 15–17, 20]. In contrast, work that explicitly compares these two types of process engines is rare, e.g., [13]. This is problematic, since, to the best of our knowledge, there is no indication that the usage of process engines is dominated by open source solutions. Instead, there are plenty of proprietary engines available, including products by large multi-national enterprises with a huge customer base world-wide. A blind spot regarding the evaluation of proprietary engines in research is problematic, as, potentially, the quality of such engines might be vastly different. An omission of these tools could result in wrong and unfounded conclusions that are not generalizable. This danger is especially valid for practical studies or case studies that depend on particular engines. It is our intention to extend *betsy* to support the benchmarking of more proprietary engines. This is most important for BPMN engines, where no proprietary implementations are supported so far. The biggest obstacle in this endeavor is the licensing strategy of many vendors. Pseudonymization of research results, as used in [13], is a way to relieve restrictions, given academic licenses are available, but this is not always the case. By working together with the vendors, we see a possibility to publish the results nonetheless. What also makes benchmarking proprietary engines complicated is that most proprietary tools are not simple BPMN engines but full-fledged BPM suites. This heavily affects both the installation and startup procedures which are complex and take a long time. We already provide an approach to use virtual machines with snapshots to easily restore a started process engine within a virtual machine [17]. Currently, this is quite cumbersome to use. Therefore, we are aiming to replace this with Docker and its light-weight containers as they are working to include a similar snapshot functionality as well.

Each engine, being it open source or proprietary, has to fulfill certain criteria so that it can be tested by *betsy*. For BPEL 2.0, we already created an API to handle engines uniformly in [14], making it easier to add new engines or new versions of existing engines. In the future, we plan to extend this API to include BPMN 2.0 engines as well. This is especially important for the proprietary engines as they do have more complex APIs, resulting in a higher entry barrier to actually benchmark them.

Dimension process engine environment: For reproducible research and reproducible benchmarks alike, it is paramount that results are correct and their computation is repeatable [5]. Currently, we use a fresh engine installation for every test, ensuring test isolation and an absence of side-effects. Furthermore, *betsy* is fully automated and therefore provides repeatable results. Again, the usage of container technology is promising to achieve an even higher degree of isolation fixing the benchmark environment, which makes it easier to repeat the benchmark.

What is more, we showed in [17], that virtualization helps to circumvent the install and startup times of the engines, reducing the time to compute the benchmark results drastically, thus, leading to a significantly lower turnaround time [3]. This is helps to integrate our benchmark into contemporary continuous integration infrastructures, which can be used by the engine vendors to improve the quality of their implementations. To reduce the execution time even further, we suggest cutting down unnecessary waiting time by calibrating timeouts required during testing to better match the actual system performance. Also parallel and distributed test execution forms a promising area of future work.

5 Conclusion

In this paper, we have presented a roadmap for process engine benchmarking using the *betsy* system. We delineated important dimensions for engine benchmarking and outlined what has been achieved so far in these dimensions with *betsy*. This identifies gaps in current work and outlines potential areas for future work in the area of process engine benchmarking, including a) to put more focus on testing proprietary engines, b) porting benchmarks for BPEL to BPMN engines, and c) speeding up process engine benchmarks through parallelization and virtualization technologies. By filling these gaps in the future, we hope to support process engine users in a meaningful decision when selecting an engine. To help users with such decisions, we are planning to publish all benchmark results as an interactive website. Furthermore, our work could help process engine vendors to enhance the quality of their products, e.g., by integrating the conformance test features of betsy into their continuous integration processes. This should reduce the occurrence of test regressions we were able to reveal in our results. Because of this, we aim to get engine vendors on board, fostering and validating our results.

References

1. van der Aalst, W.M.P.: Business Process Management: A Comprehensive Survey. ISRN Software Engineering pp. 1–37 (2013)
2. van der Aalst, W., ter Hofstede, A.: YAWL: yet another workflow language. Information Systems 30(4), 245–275 (June 2005), http://www.sciencedirect.com/science/article/pii/S0306437904000304
3. Banker, R.D., Datar, S.M., Kemerer, C.F.: Factors affecting software maintenance productivity: An exploratory study. In: Proc. of the 8th Intl. Conf. on Information Systems. pp. 160–175. University of Pittsburgh Pittsburgh, PA (1987)
4. Bianculli, D., Binder, W., Drago, M.L.: Automated Performance Assessment for Service-oriented Middleware: A Case Study on BPEL Engines. In: Proc. of the 19^{th} Int. Conf. on World Wide Web. pp. 141–150. ACM, New York, USA (2010)
5. Boettiger, C.: An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review 49(1), 71–79 (2015)
6. Chinosi, M., Trombetta, A.: BPMN: An introduction ot the standard. Computer Standards & Interfaces 34(1), 124–134 (January 2012)
7. Delgado, A., Calegari, D., Milanese, P., Falcon, R., García, E.: A Systematic Approach for Evaluating BPM Systems: Case Studies on Open Source and Proprietary Tools. In: Open Source Systems: Adoption and Impact, pp. 81–90. Springer (2015)
8. Ferme, V., Ivanchikj, A., Pautasso, C.: A Framework for Benchmarking BPMN 2.0 Workflow Management Systems. In: 13th Intl. Conf. on Business Process Management (BPM 2015). Springer, Innsbruck, Austria (August 2015)
9. Geiger, M., Harrer, S., Lenhard, J., Casar, M., Vorndran, A., Wirtz, G.: BPMN Conformance in Open Source Engines. In: IEEE Intl. Symp. on Service-Oriented System Engineering. IEEE, San Francisco Bay, CA, USA (March 30 – April 3 2015)
10. Geiger, M., Wirtz, G.: BPMN 2.0 Serialization - Standard Compliance Issues and Evaluation of Modeling Tools. In: 5th Int. Workshop on Enterprise Modelling and Information Systems Architectures. St. Gallen, Switzerland (September 2013)
11. Gray, J.: Benchmark Handbook: For Database and Transaction Processing Systems. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1992)
12. Harrer, S., Lenhard, J., Wirtz, G.: BPEL Conformance in Open Source Engines. In: IEEE SOCA, Taipei, Taiwan. pp. 237–244. IEEE (17–19 December 2012)
13. Harrer, S., Lenhard, J., Wirtz, G.: Open Source versus Proprietary Software in Service-Orientation: The Case of BPEL Engines. In: Proc. of the 11^{th} Intl. Conf. on Service-Oriented Computing (ICSOC'13). pp. 99–113. Berlin, Germany (2013)

14. Harrer, S., Lenhard, J., Wirtz, G., van Lessen, T.: Towards Uniform BPEL Engine Management in the Cloud. In: Proceedings des CloudCycle14 Workshops auf der 44. Jahrestagung der Gesellschaft für Informatik e.V. GI (September 2014)
15. Harrer, S., Nizamic, F., Wirtz, G., Lazovik, A.: Towards a Robustness Evaluation Framework for BPEL Engines. In: IEEE Intl. Conf. on Service-Oriented Computing and Applications. pp. 199–206. IEEE, Matsue, Japan (17–19 November 2014)
16. Harrer, S., Preißinger, C., Wirtz, G.: BPEL Conformance in Open Source Engines: The Case of Static Analysis. In: IEEE Intl. Conf. on Service-Oriented Computing and Applications. pp. 33–40. IEEE, Matsue, Japan (17–19 November 2014)
17. Harrer, S., Röck, C., Wirtz, G.: Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines. In: Proc. of the 7^{th} IEEE Intl. Conf. on Software Testing, Verification and Validation Workshops (ICSTW'14). pp. 390–398. Cleveland, Ohio, USA (April 2014), testing Tools Track
18. Huppler, K.: The Art of Building a Good Benchmark. In: Performance Evaluation and Benchmarking. Springer Berlin Heidelberg (2009)
19. ISO/IEC: ISO/IEC 19510:2013 – Information technology - Object Management Group Business Process Model and Notation (November 2013), v2.0.2
20. Lenhard, J., Harrer, S., Wirtz, G.: Measuring the Installability of Service Orchestrations Using the SQuaRE Method. In: IEEE Intl. Conf. on Service-Oriented Computing and Applications (SOCA). IEEE, Kauai, HI, USA (December 16-18 2013)
21. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, E., Elabed, L., Boussaidi, G.E.: Business Process Modeling Languages: Sorting Through the Alphabet Soup. ACM Comput. Surv. 43(1), 4:1–4:56 (December 2010)
22. OASIS: Web Services Business Process Execution Language (April 2007), v2.0
23. Pautasso, C., Ferme, V., Roller, D., Leymann, F., Skouradaki, M.: Towards Workflow Benchmarking: Open Research Challenges. In: Conf. on Database Systems for Business, Technology, and Web. pp. 1–20. Hamburg, Germany (March 2015)
24. Röck, C., Harrer, S., Wirtz, G.: Performance Benchmarking of BPEL Engines: A Comparison Framework, Status Quo Evaluation and Challenges. In: Proc. of the 26^{th} Intl. Conf. on Software Engineering and Knowledge Engineering (SEKE'14). pp. 31–34. Vancouver, Canada (July 2014)
25. Saaty, T.L.: How to make a decision: The Analytic Hierarchy Process. European Journal of Operational Research 48(1), 9–26 (1990)
26. Sim, S.E., Easterbrook, S., Holt, R.C.: Using Benchmarking to Advance Research: A Challange to Software Engineering. In: 25th International Conference on Software Engineering. Portland, Oregon, USA (May 2003)
27. Sim, S.E., Easterbrook, S., Holt, R.C.: Using benchmarking to advance research: A challenge to software engineering. In: Proceedings of the 25th International Conference on Software Engineering. pp. 74–83. IEEE Computer Society (2003)
28. Skouradaki, M., Roller, D.H., Leymann, F., Ferme, V., Pautasso, C.: On the Road to Benchmarking BPMN 2.0 Workflow Engines. In: Proc. of the 6th ACM/SPEC Intl. Conf. on Performance Engineering. pp. 301–304. ACM (2015)
29. Triantaphyllou, E.: Multi-criteria decision making methods: a comparative study, vol. 44. Springer Science & Business Media (2013)
30. Weyuker, E.J., Vokolos, F.I.: Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study. IEEE Trans. Softw. Eng. 26(12), 1147–1156 (December 2000)
31. WfMC: XML Process Definition Language (August 2012), v2.2

Elastic Manufacturing Process Landscapes

Olena Skarlat

Distributed Systems Group, TU Wien

Abstract Because of increasing competition and cost pressure, the manufacturing industry is currently undergoing massive changes that are facilitated by the usage of Information Technologies. Two particular aspects are the usage of Business Process Management (BPM) and Cloud technologies concepts in the manufacturing domain. Rapid elasticity is crucial for the enactment of manufacturing processes in the Cloud. This work in progress paper aims at presenting some basic principles of elastic processes in the manufacturing domain. Henceforth, an approach towards adaptive infrastructure provisioning that allows for predefined Quality of Service (QoS) and Service Level Agreement (SLA) metrics in manufacturing Cloud environments is considered.

Keywords: Elastic Processes, Cloud Manufacturing, Cyber-Physical System, Industry 4.0, Cloud Computing

1 Introduction

The manufacturing industry is now supported by means of a systematic approach of Business Process Management (BPM). Since companies in this industry have to cope with volatile process landscapes, the usage of Cloud resources is a promising approach. However, Cloud support is hardly seen in the BPM area, since most BPM frameworks only support a fixed amount of resources for process execution [6]. In the manufacturing domain there is a need of flexible scaling of manufacturing assets [9] (e.g., sensors, Cyber-Physical System objects) and of instant access to efficient and innovative business technology solutions on a pay-as-you-go basis [8]. Such flexible business processes enacted on the basis of smart resource provisioning in the Cloud are called *elastic processes* [1]. In real-world scenarios, a BPM framework for elastic processes, also known as *elastic BPM System* (eBPMS), needs to be able to solve optimization problems, specifically of scheduling and resource provisioning under a potentially heavy load [3]. Elasticity in manufacturing process landscapes establishes a new infrastructure provisioning approach aiming at the achievement of a minimax effect: minimization of the product life-cycle expenses of the manufacturers and maximization of the production efficiency providing agile accommodation of available manufacturing assets to variable demands of the customers [5,4].

This work in progress paper aims at presenting a methodology for elastic enactment of manufacturing processes in the Cloud. The contribution of this paper

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

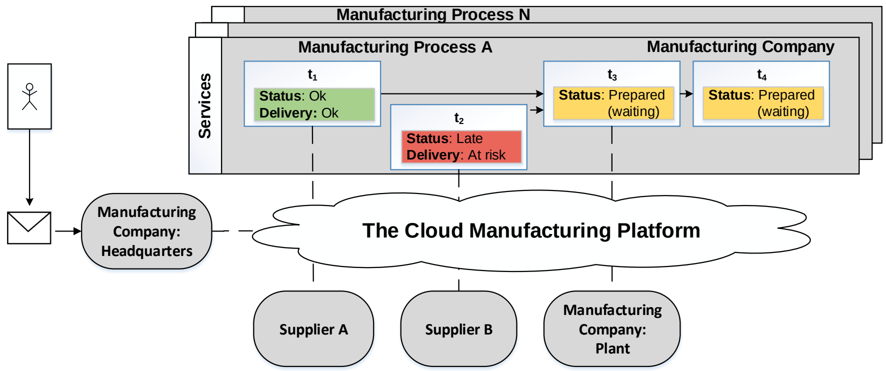


Figure 1. Cloud Manufacturing Scenario

is a motivational scenario of how to apply elasticity principles in manufacturing process landscapes and an overview of instrumentation for its implementation, specifically, of service scheduling and resource provisioning mechanisms.

2 Scenario and Research Questions

To motivate our work we consider a scenario from the manufacturing industry called Cloud Manufacturing. Cloud Manufacturing is a new concept of networked manufacturing that makes use of crowdsourcing and outsourcing models for manufacturing processes. Manufacturing processes here has to be considered as a set of process steps to be performed to create a certain manufacturing product. These process steps are in reality single services, which are responsible for a certain manufacturing asset on the shop floor. Ideally, Cloud Manufacturing offers means to integrate single services of the manufacturing processes from distributed locations as if the complete manufacturing was carried out on the same shop floor (Fig. 1). For this the manufacturers virtualize their single services of manufacturing processes. An integration is possible via a Cloud Manufacturing platform, where these services are presented, advertized, leased, and sold as a part of manufacturing processes maintained in the platform.

The existence and popularization of virtual enterprises sets the challenges to this new concept of Cloud Manufacturing. Virtual enterprises imply plugging together independent virtual factories to manufacture a certain product [7,2]. In contrast, Cloud Manufacturing assumes encapsulation of manufacturing assets into services in the Cloud Manufacturing platform as an inevitable part of its scenario. Cloud Manufacturing provides means to abstract single manufacturing assets, like sensors and Cyber-Physical System objects, and present them as services in the Cloud Manufacturing platform. The aim is to inform about available services on the Marketplace, suggest appropriate services or needed substitutes to the manufacturers, and optimize manufacturing processes.

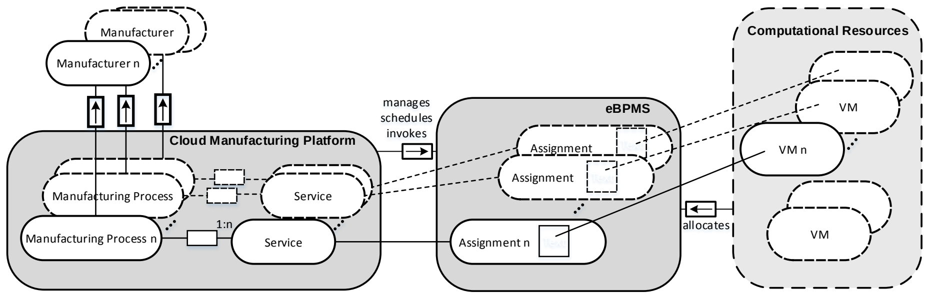


Figure 2. High-Level View on Resource Provisioning in Cloud Manufacturing

The challenge here is that thousands of manufacturers must be simultaneously served. Correspondingly, a large amount of interdependent processes with different QoS and SLA demands may be requested at any point of time. Therefore, elastic processes are a promising approach in Cloud Manufacturing [5]. To adhere to these principles, resource elasticity allowing on-demand scaling of computational resources has to be established inside the Cloud Manufacturing environment. However, computational resource elasticity is not the only dimension to be regarded: cost and quality elasticity bring flexibility in price levels of Cloud services and close the tradeoff between QoS metrics and cost. To enact elastic processes, a BPMS with features to control the Cloud is needed, as depicted in (Fig. 2). Allowing for the demand in computational resources and taking into account QoS demands, such an eBPMS schedules process instances (respectively, the single services used for the enactment of these processes) and allocates Cloud-based resources as necessary. To correlate with elasticity metrics (resources, quality, and cost), it is assumed that the manufacturing processes mentioned in the scenario are composed from single software services instantiated on Virtual Machines (VMs) in the Cloud. With this in mind, achieving elasticity implies leasing VMs when needed, deployment of services onto those VMs, invocation of service instances using a calculated schedule, and releasing resources after services are finished.

In real-world manufacturing processes an eBPMS needs to be able to solve these optimization problems in very short time and under potentially heavy load. Existing exact methods can provide a solution for small-scale scenarios. However, applying exact methods in large-scale manufacturing process landscapes is time-consuming or may not provide any solution in polynomial time, since the underlying decision problem is NP-hard [3]. Therefore an elastic manufacturing process enactment requires reasoning methodologies based on heuristic algorithms. The research question that is tackled in this work is the following: "What is an appropriate reasoning model for smart resource provisioning in elastic manufacturing process landscapes, and what are the methodologies and instrumentation to support an application of this model in the manufacturing domain?" Specifically, this work at progress is focused on presenting an underlying optimization model, and on implementing a heuristic reasoning mechanism for an eBPMS to adaptively select services for service orchestration in manufacturing processes, optimizing service scheduling and resource allocation.

To conclude, the use of elastic processes within the Cloud Manufacturing domain facilitates machine- and human-collaboration in the manufacturing industry. The aim of Cloud Manufacturing is to perform a transformation from production-oriented manufacturing processes to service-oriented manufacturing process networks by virtualizing manufacturing assets as services similarly as Software-as-a-Service or Platform-as-a-Service solutions are already provided by the Cloud providers. Clouds bring benefits to the manufacturing process management, and adaptive elastic enactment of manufacturing processes is intended to take into account QoS and SLA metrics and to perform an optimization and a runtime adjustment of infrastructural components.

Acknowledgments This paper is supported by TU Wien research funds. This work is partially supported by the Commission of the European Union within the CREMA H2020-RIA project (Grant agreement no. 637066).

References

1. Dustdar, S., Guo, Y., Satzger, B., Truong, H.L.: Principles of elastic processes. IEEE Internet Computing 15(2), 66–71 (2011)
2. Hao, Y., Shamsuzzo, A., Helo, P.: Designing of virtual factory information system by enterprise portal. In: Proc. 17th IEEE International Enterprise Distributed Object Computing Conference Workshops. pp. 258–266. IEEE (9–13 Sep 2013)
3. Hoenisch, P., Schuller, D., Schulte, S., Hochreiner, C., Dustdar, S.: Optimization of Complex Elastic Processes (accepted for publication). IEEE Transactions on Services Computing (2015)
4. Leitner, P., Hummer, W., Dustdar, S.: Cost-based optimization of service compositions. IEEE Transactions on Services Computing 6(2), 239–251 (2013)
5. Schulte, S., Hoenisch, P., Hochreiner, C., Dustdar, S., Klusch, M., Schuller, D.: Towards process support for cloud manufacturing. In: Proc. 18th IEEE International Enterprise Distributed Object Computing Conference. pp. 142–149. IEEE Computer Society (1–4 Sep 2014)
6. 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)
7. Schulte, S., Schuller, D., Steinmetz, R., Abels, S.: Plug-and-play virtual factories. IEEE Internet Computing 16, 78–82 (2012)
8. Wen, X., Zhou, X.: Servitization of manufacturing industries based on cloud-based business model and the down-to-earth implementary path. International Journal of Advanced Manufacturing Technology pp. 1–18 (2014)
9. Wu, D., Greer, M.J., Rosen, D.W., Schaefer, D.: Cloud manufacturing: drivers, current status, and future trends. In: Proc. ASME International Manufacturing Science and Engineering Conference. pp. 1–10. ASME (10–14 Jun 2013)

A Tool for Natural Language Oriented Business Process Modeling

Timm Caporale

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

Abstract Process modeling techniques play an important role to capture information about business procedures. This paper suggests two novel methods for business process modeling. The first method allows generating process models from process descriptions created with controlled natural language. This method is based on a parser for natural language and is supported by sentence templates and an autocomplete function. The second method suggests a collaborative setting, which allows discovering process models through user interactions. Both methods have been implemented in a prototype. The aim of this paper is to show new possibilities for process modeling through the combination of the two methods.

Keywords: BPM, business process modeling, natural language processing, bottom up approach, collaborative modeling

1 Introduction

Today organizations have to adapt and improve their business processes more often and on short notice. Thus, there is a need for effective and efficient methods for business process management. The documentation and modelling of existing processes is an important part in this context; but in many cases the results of process modeling projects (with current methods) do not fully comply with the expectations of the parties involved.

2 Approach

To demonstrate new process modeling possibilities, two methods have been developed and implemented within a protoype. The presented tool is an ASP.NET MVC web application executed in a web browser. In addition, there is a Ms Office-App available to have an integration into Office products, which are often used to present or describe process models. The prototype will be available at http://bpm.caporale.eu soon.

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

2.1 Generating Process Models from Natural Language

The first method presented in this paper, is an approach of generating process models from natural language text. Instead of analyzing descriptions of business tasks and then generating the process model (related approaches are presented in e.g. [2,5]), process modelers should use pre-defined templates to describe their business tasks in controlled natural language.

The method is based on so-called sentence templates. Sentence templates are often used to describe requirements for software development projects. They can be considered as a support technique, which helps the user formulating understandable sentences. With the help of these templates, the modelers are able to describe their business tasks in controlled natural language, which will be automatically transformed into a process model. For each basic workflow control-flow pattern (Sequence, Alternative, Parallel Split, Synchronization and Simple Merge), a sentence template has been defined for the German and English language. An example can be found in e.g. [4].

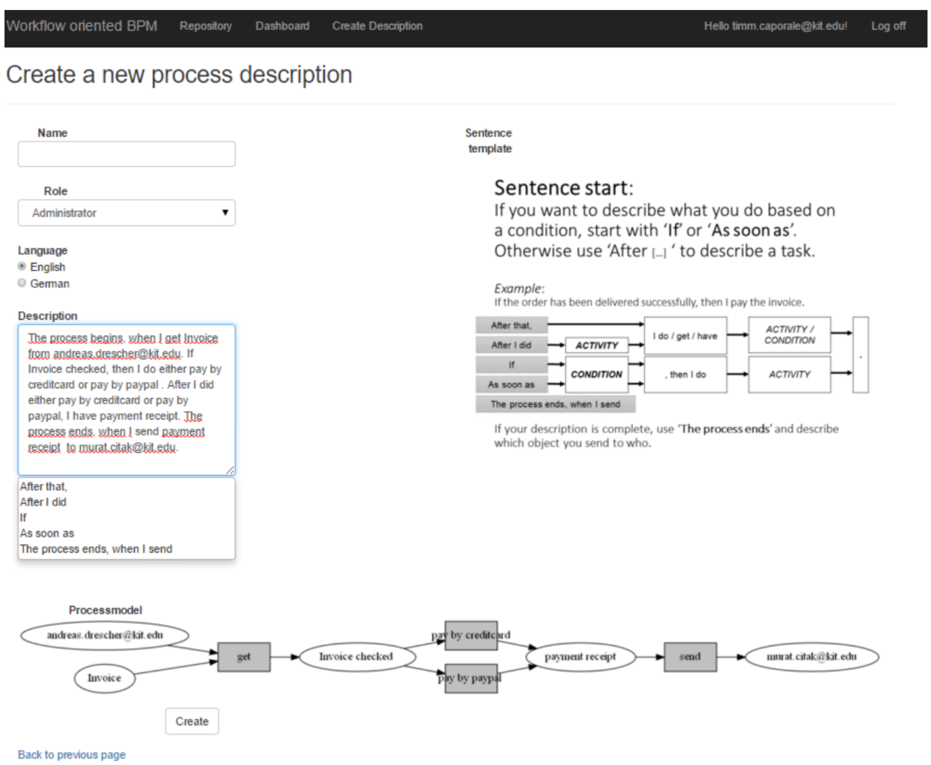


Figure 1. Screenshot of the web application

Descriptions that have been constructed using the sentence templates are automatically analyzed by the tool. For this purpose, the approach of [3] has been applied, which uses the ANTLR parser generator (http://www.antlr.org) to create a text-parser for the controlled natural language. ANTLR needs a grammar in customized extended Backus-Naur-Form. An excerpt of the grammar was published in [4] section 3.2. The text parser constructs an Abstract Syntax Tree (AST), where places, transitions and the control flow of a Petri Net can be identified. The process model that is generated from the AST is currently a Petri Net but can be transformed into other languages such as BPMN easily.

When a process modeler uses the tool, he will see a text-box on the left and the sentence templates on the right side of the tool. Synchronously to typing in the natural language text, a process model is generated at the bottom side of the tool and the sentence template is dynamically adjusted to the current context. In addition a recommender suggests possible formulations with respect to the current parser's state similar to an auto-completion function under the text-box (Fig. 1).

2.2 Workflow oriented business process modeling

The second method addresses the modeling process itself. Most of the existing modeling approaches aim to extract the process knowledge of an organization through e.g. expert interviews or workshops, which can be considered as a top down approach. In contrast, the method presented is a bottom up techniques for process discovery. The advantage of this method is to bring process modeling activities closer to the knowledge carriers. Assuming, that the tool for generating process models from natural language text can be used by knowledge carriers successfully, the following adaptions will lead to a novel collaborative setting.

The first adaption is, that the user only describes his own activities and has to provide information about the precondition and postcondition. A related approach is [1]. Information about the precondition can be formulated in natural language as well and include information about objects and persons. An example for a precondition in natural language is: *"As soon as I get a KPI-report (Ms Excel Document) from my colleague Linda (linda@example.com) I start with this process called KPI-report analysis."* Out of this sentence the text parser extracts information about the object 'KPI-report', the person Linda 'linda.example@example.com' and the process name 'KPI-report analysis'. An example of a postcondition is: *"Finally, I send the result of my KPI-analysis (Ms Word Document) to my boss (boss@example.com)."* Out of this sentence the parser extracts information about the object KPI-analysis and about the boss.

Whenever the user describes a process using the tool containing such preconditions or postconditions, the system will execute a workflow, which will inform the mentioned persons by e-mail. The e-mails will include information about the just created process and ask the receiver to provide more information to the system by clicking on a specially generated link within the e-mail. Clicking on the link will trigger a validation workflow on server-side and create a new process description for the new user that has been addressed within the e-mail.

When knowledge carriers use this collaborative approach and describe their own activities, the system will store many process models with connections between them. As this structures are similar to event logs, process mining techniques are applicable to discover more general process models.

3 Conclusion and Outlook

The presented tool combines a method for generating process models from natural language with a workflow oriented collaborative setting and shows new possibilities for process modeling.

As an outlook we assume, that the workflow oriented approach for business process modeling has several advantages. First, the fact that every knowledge carrier will only describe his own activities and will not make any assumptions about activities from other people could possibly reduce misunderstandings, which occur with other techniques when a process modeler has to understand the peoples' tasks. Second, the bottom up approach could be a new technique for discovering undescribed and unmentioned processes throughout the organization. It even even represents a new approach on Adaptive Case Management. Last, it is reasonable, that the e-mails send by the workflow tool could possibly cause a chain reaction for a new way of collaborative business process modeling.

The next steps include a first evaluation to get feedback about the described methods and improve the underlying modules and user interface.

References

1. Fleischmann, A., Schmidt, W., Stary, C., Augl, M.M.: Agiles prozessmanagement mittels subjektorientierung. HMD Praxis der Wirtschaftsinformatik 50(2), 64–76 (2013)
2. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: CAiSE 2011. pp. 482–496. Springer (2011)
3. Keuter, B.: Bidirektionale Abbildung zwischen Geschäftsprozessmodellen und IT-Kommunikationssystemen. KIT Scientific Publishing (2014)
4. Koschmider, A., Caporale, T., Fellmann, M., Lehner, J., Oberweis, A.: Business process modeling support by depictive and descriptive diagrams. In: EMISA 2015. pp. 31–44. LNI (2015)
5. Leopold, H., Mendling, J., Polyvyanyy, A.: Supporting process model validation through natural language generation. Transactions on Software Engineering 40(8), 818–840 (2014)

(Business Process) Models from an Educational Perspective

Meike Ullrich and Andreas Schoknecht

Institute for Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology (KIT), first_name.last_name@kit.edu

1 Background and Motivation

A vast amount of scientific research and literature is devoted to business process models. The spectrum includes languages, style, quality, analysis, simulation methods, tool support, and even further aspects. However, most of this work only considers business process models from an industrial perspective: the ultimate goal is to support organizations and companies through the development, use and implementation of process models. But as business process modeling, or even more generally modeling itself, has become an integral part of many programs in higher education institutions and universities, we have identified yet other challenges and usage opportunities for research with regards to models in the context of modeling education.

Year by year during our university courses concerned with modeling, a growing number of students creates models in exercises and exams. Each semester we struggle with handling and grading these big yet analogous datasets with hundreds of modeling solutions for, e.g., business processes and ER-diagrams. However, a manual correction and grading procedure is usually a time-consuming and error-prone task. Also, the consistent application of a predefined grading scheme is hard to enforce. Currently we often distribute the correction of exam questions so that only one person is responsible for one question to increase consistency while trying to be as efficient as possible. However, we still detect inconsistent or even erroneous corrections.

In the age of growing digitization and Massive Open Online Courses (MOOCs) many universities start integrating IT support into the academic processes, e.g., by offering e-assessments [6]. As opposed to written exams or tests, in an e-assessment students create digital solutions using a software tool so that the results can be analyzed (semi-)automatically. Thus, having models in digital form rather than handwritten on a sheet of paper opens up new opportunities for the educational context. First, an **automation of the exam correction process** enables an efficient and consistent grading. Secondly, we can conduct what is widely denoted with the buzzword **Learning Analytics** [3]. The results of automated analyses can be easily aggregated over large sets of models to, e.g., detect the most common mistakes. This in turn allows us to draw conclusions about the underlying teaching concept and can be used to improve university courses. With this article, we aim to illustrate those opportunities.

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

2 Towards a Technical Implementation

To enable an automated approach for processing digital process models, those should not only be digitized as an image or photograph, but encoded in a formal representation which allows an easy and straightforward access to all model elements for a subsequent analysis. An example is the Petri Net Markup Language (PNML)[1] which serves as an XML-based interchange format for Petri nets. Regarding the educational context, a fundamental decision has to be made about the model creation process. Should students enter their models with a modeling tool including modeling support or rather without any intelligent features? If we want to determine the learning outcome of our students, it might be counterproductive to have the modeling tool help them with the creation of correct models (e.g., by not allowing to draw arcs between nodes of the same type in Petri nets). Thus, in an exam setting we need (i) modeling tools without modeling restrictions or support as well as (ii), an interchange format which explicitly allows a representation of incomplete and incorrect models.

Quality assessment of conceptual models has been described extensively in the scientific literature (e.g. [4,5]). Also, guidelines have been proposed on how to create models which can be easily understood. As models are primarily interpreted by humans, the importance of this factor has been stressed many times. Despite these efforts, we witness that during exam corrections, the models created by students are mostly checked in terms of syntactical and semantical correctness without considering other relevant aspects like pragmatic quality, e.g., understandability of the model or the compliance to modeling guidelines. Hence, we want to accentuate the need for defining more suitable learning objectives and quality criteria for models created in an educational context. On the basis of a digital interchange format, several algorithms towards a quality measurement could be implemented. E.g., checking the compliance with modeling guidelines or determining the degree to which certain quality criteria are fulfilled. Even the semantic quality, which describes the degree to which a model is compliant to real-world facts, can be determined automatically by comparing a model to a digital representation of such real-world facts, e.g., in form of an ontology. Altogether, these algorithms can support the correction and grading process. Even if not all necessary steps can be performed automatically, the number of tasks to be performed manually is expected to decrease drastically. Also, to support learning analytics, it could be feasible to detect single aspects in models responsible for common mistakes, which might be addressed in future lectures.

Automatic clustering of models might help to investigate common mistakes in exams. As it takes a lot of effort to manually investigate errors in hundreds of exam solutions, these data are only rarely used to identify common mistakes which should be addressed in a corresponding lecture to improve the learning outcomes. An automatic clustering could be used to point out such mistakes through the

[1] `http://www.pnml.org/`

identification of similar models. I.e., we assume that similarity and clustering techniques for models can be adapted so that it is possible to identify clusters in which models contain the same error. Another application of clustering could be during the correction phase of an exam. According to our experiences, some mistakes are frequently made by students, which should be graded consistently. Yet, this is difficult to achieve with hundreds of solutions as a corrector usually has to keep track of certain errors and their grading scheme. In this context, clusters of similar solutions might speed up this process besides increasing the consistency of corrections as each cluster could be corrected at a time. Suitable techniques might be process matching and similarity approaches (e.g. [1,2]), which could be adapted to the education context.

3 Outlook

Our next goal is the setup of a platform to manage, organize and analyze large collections of digital process models. Analysis algorithms will be added step by step. A key element of the platform is the possibility to arrange quality criteria flexibly by choosing relevant algorithms and weighting their individual outcome according to the requirements. Plus, it is possible to run an analysis not only on a single model of a collection, but over each model in a collection to be able to aggregate the results. Besides this, we started investigating the adaptation of similarity techniques for clustering of modeling exams. Finally, we want to emphasize that we are aware about the controversial debate about the influence of digital and online elements into the traditional learning processes. We believe that, while the inevitable change from analog to digital is ongoing, it is necessary to address threats and fears like privacy issues or qualitative shortcomings of automated approaches associated with this topic right from the start.

References

1. Antunes, G., Bakhshandeh, M., Borbinha, J., Cardoso, J., Dadashnia, S., Francescomarino, C.D., Dragoni, M., Fettke, P., Gal, A., Ghidini, C., Hake, P., Khiat, A., Klinkmüller, C., Kuss, E., Leopold, H., Loos, P., Meilicke, C., Niesen, T., Pesquita, C., Péus, T., Schoknecht, A., Sheetrit, E., Sonntag, A., Stuckenschmidt, H., Thaler, T., Weber, I., Weidlich, M.: The process model matching contest 2015. In: 6th EMISA Workshop. pp. 127–155 (2015)
2. Becker, M., Laue, R.: A comparative survey of business process similarity measures. Computers in Industry 63(2), 148–167 (2012)
3. Ferguson, R.: Learning analytics: drivers, developments and challenges. International Journal of Technology Enhanced Learning 4(5/6), 304–317 (2012)
4. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding quality in conceptual modeling. IEEE Software 11(2), 42–49 (1994)
5. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions. Data Knowl. Eng. 55(3), 243–276 (2005)
6. Tomas, C., Borg, M., McNeil, J.: E-assessment: Institutional development strategies and the assessment life cycle. Brit J Educ Technol 46(3), 588–596 (2015)

A Framework for Data Stream Applications in a Distributed Cloud

Matteo Nardelli

Department of Civil Engineering and Computer Science Engineering
University of Rome Tor Vergata, Italy
nardelli@ing.uniroma2.it

Abstract The ever increasing diffusion of sensing and computing devices enables a new generation of data stream processing (DSP) applications that operate in a distributed Cloud environment. Despite this, most of the existing solutions, such as Apache Storm, are designed to run in a local cluster. In this paper we present our extension of Storm, which provides distributed monitoring, scheduling and management capabilities. Exploiting these new functionalities, the system can improve its performance and react to internal and external changes. Finally, we analyze open challenges of placing and adapting DSP applications.

Keywords: Data Stream Processing, Adaptation, Placement, Apache Storm

1 Introduction

With the disruptive diffusion of sensing devices (e. g., smartphones, cars, monitoring stations), the almost ubiquitous Internet connection, and the Fog Computing [13] paradigm, urban environments are today permeated by an ever increasing number of diffused and networked sensing and computing devices. All these sensing devices continuously produce streams of data that can be collected by distributed data stream processing (DSP) applications, to timely extract valuable information about many fundamental aspects of the environment we live in (e. g., urban mobility, public decision making, energy management). As data increases, we cannot push it toward the core of Internet. To increase scalability and reduce latency, a possible solution is to rely on distributed and near-edge computation. Furthermore, determining the computational resources that should host and execute each operator of the DSP application, i. e., solving the *operator placement* problem, is challenging because the characteristics of computational tasks are not known a-priori, the properties of the input streams change continuously, and the load imposed has to be sustained for long provisioning times. Therefore, we extended Storm [14], an open source DSP system, with policies and mechanisms that allow to find a placement that optimizes a utility function and to continuously adapt the placement when changes occur in the execution environment.

C. Hochreiner, S. Schulte (Eds.): 8th ZEUS Workshop, ZEUS 2016, Vienna, Austria, 27-28 January 2016, Proceedings – published at http://ceur-ws.org/Vol-1562

The main contributions of this paper are as follows: a) we describe how our extension implements the MAPE (Monitor, Analyze, Plan, and Execute) reference model for autonomic systems (Sect. 4); b) we show its benefits when the placement is determined according to the distributed policy proposed by Rizou et al. [12] (Sect. 5); and c) we illustrate some of the open challenges for DSP systems when they are executed in distributed environments (Sect. 6).

2 Related Work

As technologies and needs evolve in time, the DSP paradigm has experienced different generation of architectures [7], where the last one relies on Cloud-based resources. Despite this, most DSP systems are still designed to run in a local cluster, where the often homogeneous nodes are interconnected with negligible network delays (e. g., [14, 15, 17]). These assumptions do not hold any more when the DSP system runs in geographically distributed and dynamic environments, where a great heterogeneity of devices are interconnected with not-negligible network latencies. Storm, a framework of the last generation, is attracting increasing interests. However, most of the proposed Storm extensions are all centralized solutions (e. g., [1]), implicitly designed for clustered environments, which do not scale well as the number of applications increases. Our extension, instead, provides distributed monitoring, scheduling and management capabilities [3].

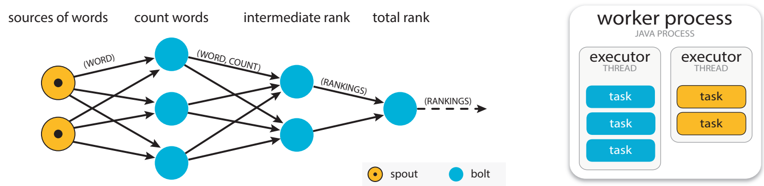
A great variety of placement algorithms have been proposed in literature. Lakshmanan et al. [10] provide a comprehensive overview of them, but, as the authors show, they differ each other on assumptions and optimization goals. Being interested in a network-aware solution, in our previous work [3] we implemented the Pietzuch's algorithm [11], and here we evaluate the strategy proposed by Rizou et al. [12]. Both the solutions minimize network usage, however the authors of [12] claim that their formulation has better convergence properties and works better in a distributed environment than [11].

Recently, another framework for large-scale processing is gaining interest: Spark [17]. It extends and improves the MapReduce approach (batch processing), and, using the Spark Streaming module, can reduce the size of each batch and process streams of data (micro-batch processing). This alternative is throughput oriented, whereas Storm, which is a pure DSP system, can further minimize the application latency, therefore is preferred in latency sensitive scenarios. Apache Flink[1] proposes a unified framework for batch and stream processing. Similarly to Storm, Flink has been originally designed to run in a cluster environment and shows the drawbacks we discuss in Sect. 4.1. DSP systems are also offered as Cloud services. Google Cloud Dataflow[2] provides a unified programming model to process batch and streaming data on top of Google cloud platform. Amazon offers Kinesis[3], which resembles an evolved publish-subscribe system, suitable to process near real-time streams of data. Both these Cloud-based services abstract

[1] `https://flink.apache.org/`

[2] `https://cloud.google.com/dataflow/`

[3] `https://aws.amazon.com/kinesis/`



(a) An example of Storm topology. The depicted application identifies the most frequent words (`http://bit.ly/1QxMn71`)

(b) Execution model

Figure 1: Storm abstractions

the underlying infrastructure, but it is reasonable to believe that they execute in a centralized data center, conversely to the context investigated in this paper.

3 Apache Storm

Storm[4] is an open source and scalable DSP system maintained by the Apache Software Foundation. It provides an abstraction layer where event-based applications can be executed over a set of worker nodes interconnected by an overlay network. A *worker node* is a generic computational resource, whereas the overlay network comprises the logical links between these nodes. In Storm, an application is represented by its *topology*, which is a directed acyclic graph with spouts and bolts as vertices and streams as edges. A *spout* is a data source that feeds the data into the system through one or more streams. A *bolt* is either a processing element, which extracts valuable information from incoming data and generates new outgoing streams, or a final information consumer. A *stream* is an unbounded sequence of *tuples*, which are key-value pairs. We refer to spouts and bolts as operators. Figure 1a shows an example of a DSP application. Storm uses three types of entities with different grain to execute a topology. A *task* is an instance of an operator in charge of a share of its incoming streams. An *executor* can execute one or more tasks related to the same operator. A *worker process* is a Java process that runs a subset of executors of the *same* topology. As represented in Fig. 1b, there is a hierarchy among these entities: a group of tasks runs sequentially in the executor, which is a thread within the worker process that serves as container on the worker node. Besides the computational resources (i. e., worker nodes), Storm includes two centralized components: Nimbus and ZooKeeper. *Nimbus* coordinates the topology execution and defines the placement of its operators on the available worker nodes. This assignment plan is communicated to all the worker nodes through *ZooKeeper*, which is a shared memory service that enables distributed coordination. Since each worker node can execute one or more worker processes, a *Supervisor* component on the node starts and terminates worker processes on the basis of the Nimbus decisions.

[4] `http://storm.apache.org/`

4 Distributed Storm

4.1 From Cluster to Distributed Cloud: A Gap to Close

Storm has been originally designed to run in a local cluster, where network delays are negligible. If we deploy Storm in a distributed Cloud, it shows poor performances, because of the assumption that data can quickly move between computational nodes. We can summarize the limitations that Storm shows in this new environment as follows: 1) it is unaware of QoS attributes (e. g., resource utilization, network delays) of computational and network resources; 2) its placement decision is static, therefore the system cannot adapt to internal (i. e., application) and external (i. e., environmental) changes; and 3) if we create a custom centralized scheduler that collects the QoS attributes for each node and periodically evaluates the placement of each application, it will not scale well as the number of applications and network resources increases. In a geographically distributed environment, we would like to have a framework that considers network delays and resource heterogeneity while determining placement decisions.

4.2 Distributed Scheduling in Storm

We have extended the Storm architecture to run distributed, adaptive, and QoS-aware scheduling algorithms [3]. The newly introduced components, illustrated in orange in Fig. 2, are: the AdaptiveScheduler, the QoSMonitor, and the WorkerMonitor. We preserved the centralized scheduler, named BootstrapScheduler, which defines the initial placement of the application. The *AdaptiveScheduler* is the distributed scheduler that coordinates the MAPE control cycle. It executes on each Supervisor together with the *QoSMonitor*, an infrastructure level monitoring component. The *WorkerMonitor* is an application level monitor and runs on each worker process. Exploiting the feedback control loop, the distributed scheduler can react to internal and external changes of the operating conditions. In a single loop iteration, it monitors the environment and the locally executed executors,

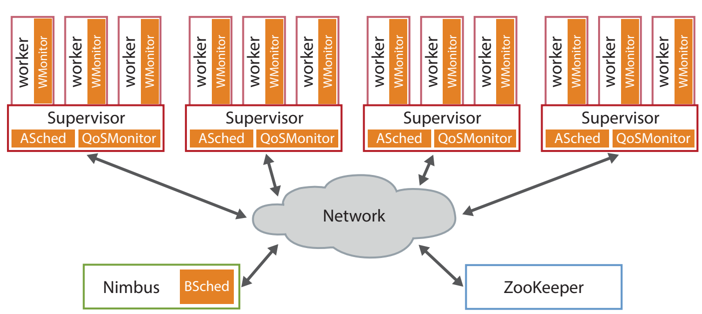


Figure 2: Storm architecture with new components in orange: AdaptiveScheduler (abbreviated as ASched), WorkerMonitor (WMonitor), and BootstrapScheduler (BSched).

analyzes if there are candidate executors for a new reassignment, and, in positive case, plans and executes the corresponding repositioning actions.

Monitor. The AdaptiveScheduler acquires the information on computational resources and on executors that run locally through the QoSMonitor and the WorkerMonitors respectively. The QoSMonitor provides the QoS awareness to each distributed scheduler, thus it is responsible of obtaining intra-node information (i. e., utilization and availability) and inter-node information (i. e., network delays). For the latter, it resorts on a network coordinates system [5] that provides an accurate estimate of the delays between any two computational nodes without the need of an exhaustive probing. The WorkerMonitor computes the exchanged data rate for each executor that runs on the node.

Analyze and Plan. A distributed scheduling policy drives these two phases. Our previous work [3] relies on the Pietzuch's placement algorithm [11]. In this paper, we use the scheduling solution designed by Rizou et al. [12], which places the application minimizing the network usage (i. e., sum of bandwidth-delay product for each application link). Implementing the Rizou's algorithm within the extended Storm requires just few changes. Basically, it needs to account for the specific Storm application model, where a processing operator can be instantiated in one or more executors and pinned operators are not modeled. Furthermore, the algorithm can readily obtain QoS information (i. e., latency, bandwidth) relying on the monitoring components.

Execute. Finally, if a new assignment must take place, the executor is moved to the new candidate node. The new assignment decision is shared with the involved worker nodes through ZooKeeper. We note that in Storm an executor reassignment does not preserve its state; thus, the executor is stopped on the previous worker node and started on the new one.

Thanks to the adaptation cycle, the distributed scheduler can manage changes that may occur both in the infrastructure layer (e. g., a worker node appears or fails) and application layer (e. g., data rate fluctuations).

The source code of our extension is available at `http://bit.ly/extstorm`.

5 Experimental Results

We show the improvements and the self-adaptation capabilities of our distributed scheduler equipped with the Rizou's algoritm (named as dRizou) with respect to the centralized and default EvenScheduler of Storm (named as cRR). For a better evaluation, we also indicate the behaviour of dQoS, that is the distributed scheduler equipped with the Pietzuch's algoritm (further details in [3]). dRizou and dQoS place operators exploiting QoS attributes, whereas cRR uses a round-robin policy. The evaluation uses a cluster of 8 worker nodes (each can host at most 2 worker processes) and 2 further nodes for Nimbus and ZooKeeper. We emulated wide-area network latencies among the Storm nodes applying to outgoing packets a Gaussian delay with mean and standard deviation in the ranges $[12, 32]$ ms and $[1, 3]$ ms, respectively. The DSP application is composed of a source, which generates 10 tuples/s, followed by a sequence of 5 operators before

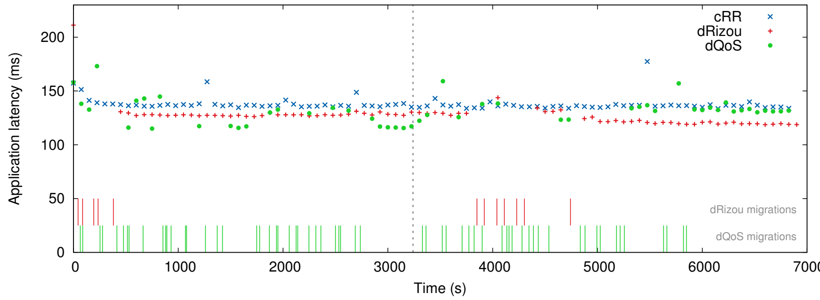


Figure 3: Performance of the tag-and-count topology when the nodes' utilization changes

reaching the final consumer. The placement of source and consumer is fixed. The other operators are unpinned and replicated (i. e., two executors are assigned to each of them). Figure 3 shows the evolution of the application end-to-end latency; on its bottom, we indicate the run-time reassignments performed by the distributed schedulers (cRR does not intervene during the execution). We start the application and, after 3240 s, we artificially increase the load on a subset of three nodes using the Linux tool `stress`. The subset is composed by one worker node running some application executors and two free worker nodes. This event is represented in Fig. 3 with a vertical dotted line. As the distributed scheduler (both dQoS and dRizou) perceives the change, it moves the application operators on lightly loaded nodes. cRizou reduces the application latency with respect to cRR of about 12.6 % (measured between 5000 s and the end of the experiment). Furthermore, differently from dQoS, dRizou converges with a lower number of reassignments, increasing the application availability.

6 Open Challenges

Although our extension enables the execution of the Storm, as a generic DSP system, in a distributed environment, the peculiarities of Cloud computing require an efficient management of scalability, elasticity and fault tolerance. With no claim of completeness, we summarize some of the needed mechanisms.

Stateful Migration: an operator is stateful if its behavior depends also on its internal state. Therefore, moving a stateful operator requires an efficient relocation of its internal, possibly extremely large, state across the network. In literature, the general tendency is to use the strategy *stop-move-play*, which stops the incoming streams, moves the operator and its state, and redirects the streams to the new operator location (e. g., [4, 6]). Wu et al. [16] improve this technique by aggressively dividing the application-level state in computation slices, which are asynchronously checkpointed to remote machines, enabling parallel state migrations between nodes. However, most of the existing techniques do not fit well in a latency sensitive scenario, because they do not explicitly consider QoS attributes of communication links and computational nodes. A fast, live, and QoS-aware migration strategy could bring important improvements to these systems.

Elastic Replication: the ability of the system to autonomously adapt the number of replicas for each operator. This mechanism can increase non-functional attributes of the applications (e. g., availability) with the penalty of a higher cost and resource overhead. Bellavista et al. [2] present a prototype that allows to trade-off monetary cost and active replication. An alternative to active replication is upstream backup, which achieves fault-tolerance using an upstream server that stores a copy of the operator state. However, since this technique imposes a higher recovery time, it is used as a second-class mechanism. For example, Heinze et al. [8] combine these two mechanisms to reduce the overall resource consumption with respect to a recovery time threshold.

Elastic Sharding: the ability of automatically scaling in and out the number of shards for an operator based on the incoming load. Each shard of an operator is in charge of a partition of its incoming stream; therefore, this mechanism can increase the application scalability by handling a growing workload. As a consequence, the system can acquire and release resources when needed, without resorting in over- or under-provisioning (i. e., resource elasticity [9]). Increasing the number of shards is critical for stateful operators, because the system needs to preserve the consistency of the operations. In literature different works investigate this issue. Some solutions define a-priory the maximum number of shards [14], expose some API to manually manage the state [4], or automatically determine the optimal number of state partitions to be used [6].

Solutions to the above mentioned issues are almost consolidated in a clustered environment, however the emerging distributed Cloud scenario imposes a new perspective. In a distributed environment, aside the number of replicas or shards, the scheduler should also define their optimal placement with respect to some QoS metrics (e. g., latency, bandwidth, reliability), considering the heterogeneity of applications and resources. For example, replicas should be placed in different availability zones; different shards of the same operator should let users experience similar response times.

7 Conclusion

The ever increasing diffusion of sensing and computing devices enables a new generation of DSP systems. Starting from the major drawbacks of an existing framework to the execution in distributed and dynamic environments, we developed an extension of Apache Storm that provides distributed monitoring, scheduling and management capabilities. The evaluation results showed that our extension of Storm is suitable to operate in a distributed environment, where QoS-awareness and adaptation capabilities can be truly beneficial to the application performances. Finally, we highlighted some core mechanisms that can improve performances of DSP systems when executed in a distributed Cloud.

As future work, we will provide a formal definition of the placement problem for DSP applications and design a new placement algorithm that better leverages the potentialities of a distributed Cloud model.

Acknowledgments. Thanks to the anonymous reviewers for the valuable comments, to Valeria Cardellini, and to Gabriele Scolastri for the implementation of the Rizou's algorithm.

References

1. Aniello, L., Baldoni, R., Querzoni, L.: Adaptive online scheduling in Storm. In: Proc. of ACM DEBS '13. pp. 207–218 (2013)
2. Bellavista, P., Corradi, A., Kotoulas, S., Reale, A.: Adaptive fault-tolerance for dynamic resource provisioning in distributed stream processing systems. In: EDBT. pp. 85–96 (2014)
3. Cardellini, V., Grassi, V., Lo Presti, F., Nardelli, M.: Distributed QoS-aware Scheduling in Storm. In: Proc. of ACM DEBS '15. ACM (2015)
4. Castro Fernandez, R., Migliavacca, M., Kalyvianaki, E., Pietzuch, P.: Integrating scale out and fault tolerance in stream processing using operator state management. In: Proc. of ACM SIGMOD'13. pp. 725–736. ACM (2013)
5. Dabek, F., Cox, R., Kaashoek, F., Morris, R.: Vivaldi: A decentralized network coordinate system. SIGCOMM Comput. Commun. Rev. 34(4), 15–26 (2004)
6. Gedik, B., Schneider, S., Hirzel, M., Wu, K.L.: Elastic scaling for data stream processing. IEEE Trans. Parallel Distrib. Syst. 25(6), 1447–1463 (2014)
7. Heinze, T., Aniello, L., Querzoni, L., Jerzak, Z.: Cloud-based data stream processing. In: Proc. of ACM DEBS '14. pp. 238–245 (2014)
8. Heinze, T., Zia, M., Krahn, R., Jerzak, Z., et al.: An adaptive replication scheme for elastic data stream processing systems. In: Proc. of ACM DEBS '15. pp. 150–161. ACM (2015)
9. Hochreiner, C., Schulte, S., Dustdar, S., Lecue, F.: Elastic stream processing for distributed environments. Internet Computing, IEEE 19(6), 54–59 (2015)
10. Lakshmanan, G.T., Li, Y., Strom, R.: Placement strategies for internet-scale data stream systems. Internet Computing, IEEE 12(6), 50–60 (2008)
11. Pietzuch, P., Ledlie, J., Shneidman, J., Roussopoulos, M., et al.: Network-aware operator placement for stream-processing systems. In: Proc. of IEEE ICDE '06 (2006)
12. Rizou, S., Durr, F., Rothermel, K.: Solving the multi-operator placement problem in large-scale operator networks. In: Proc. of IEEE ICCCN 2010 (2010)
13. Satyanarayanan, M., Schuster, R., Ebling, M., Fettweis, G., et al.: An open ecosystem for mobile-cloud convergence. IEEE Communications 53(3), 63–70 (2015)
14. Toshniwal, A., Taneja, S., Shukla, A., Ramasamy, K., et al.: Storm@Twitter. In: Proc. of ACM SIGMOD '14. pp. 147–156 (2014)
15. Urbani, J., Margara, A., Jacobs, C., Voulgaris, S., et al.: AJIRA: a lightweight distributed middleware for MapReduce and stream processing. In: Proc. of IEEE ICDCS '14. pp. 545–554 (2014)
16. Wu, Y., Tan, K.L.: Chronostream: Elastic stateful stream computation in the cloud. In: IEEE Int'l Conf. ICDE 2015 (forthcoming) (2015)
17. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., et al.: Spark: Cluster computing with working sets. In: Proc. of USENIX HotCloud'10. p. 10 (2010)

ECHO – An mHealth Solution to Support Treatment of Chronic Patients

Frank Steimle and Matthias Wieland

University of Stuttgart, Institute of Parallel and Distributed Systems,
Universitätsstraße 38, 70569 Stuttgart, Germany
`firstname.lastname@ipvs.uni-stuttgart.de`

Abstract More and more people all over the world suffer from chronic diseases, like asthma. The German-Greek bilateral research project Enhancing Chronic Patients Health Online developed online services for physicians and patients for use on smart phones or web browsers, in order to improve monitoring of those patients and to be able to detect possible exacerbations earlier. During the project we have developed smart phone applications and websites for both patients and physicians and a cloud-based health data management system. This demonstration shows how our system supports physicians and patients.

Keywords: mHealth, eHealth, Monitoring, Cloud Computing, Analysis

1 Introduction

Chronic diseases like diabetes, asthma, or chronic obstructive disease (COPD) are on the rise. For the proper treatment of chronic patients regular check-ups are inevitable. But due to lack of time or economic difficulties many patients do not get regular check-ups, which possibly leads to an exacerbation of their condition or even hospitalization.

In the project *Enhancing Chronic Patients Health Online (ECHO)*[1] smart phones, cloud computing, and data analytics are used to enable regular monitoring of COPD patients and avoidance of exacerbations. Patients who use the ECHO System are able to answer questions on their condition on a daily basis using their smart phones. The ECHO System even enables patients to enter measurements, like heart rate or body temperature. After submitting the daily report to the ECHO System, the data is analyzed by the ECHO System. If the system detects an imminent aggravation of the patients' health, the patient and the corresponding physician get notified and, if applicable, a treatment recommendation is given. Additionally, the ECHO System is able to store medical data like results of examinations or prescribed drugs. This extra data can be used to improve the analytics, such that imminent aggravations could be detected even earlier.

Section 2 presents a system overview and Section 3 presents our demonstration scenario.

[1] Project Website: `http://www.chroniconline.eu`

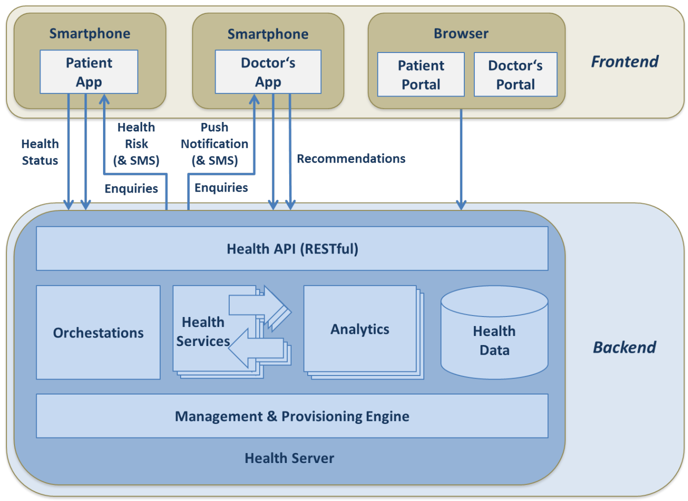


Figure 1. Architecture of the ECHO System [1]

2 System Overview

As shown in Fig. 1, the ECHO System is made up of 2 layers: Frontend-Layer and Backend-Layer. The Frontend-Layer consists of web and mobile applications for patients and physicians. The Backend-Layer contains the Health Server which is a cloud service. The Health Server is accessible from the Frontend-Layer via the Health API, which is a RESTful HTTP-API. The Health API enables the applications of the Frontend-Layer to use the Health Services and the Analytics. The Health Services can be used to store and query data from the Health Data Repository, where all patients' data is stored. The Analytics provide procedures to analyze data in the Health Data Repository, e.g., simple procedures to analyze the incoming daily reports or complex procedures which perform data mining. The Orchestrations can be used to orchestrate Health Services and Analytics to new complex services. Finally, the Management and Provisioning Engine is responsible for managing all the before mentioned components. Details on the implementation can be found in [3].

3 Demonstration Scenario

In our Demonstration Scenario we will show how the system can be used by physicians and patients. Fig. 2 shows the steps that need to be performed in order to monitor a single patient. The grey colored activities are administrative tasks. The first step, which is deploying the TOSCA[2] cloud service in a secure private cloud, will not be part of our demonstrations since it would take too much time. After that the administrator of the system creates an account for

Figure 2. Required steps for monitoring a patient

each physician and each patient. The physician is also able to create accounts for his patients using the web application. Fig. 3 shows the physicians' patients list in the web application. The next activity is the creation and maintenance

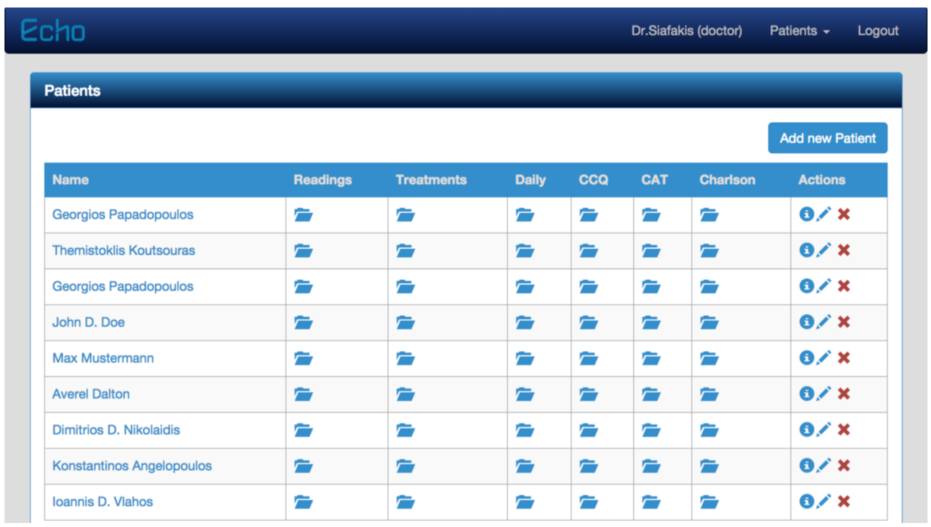
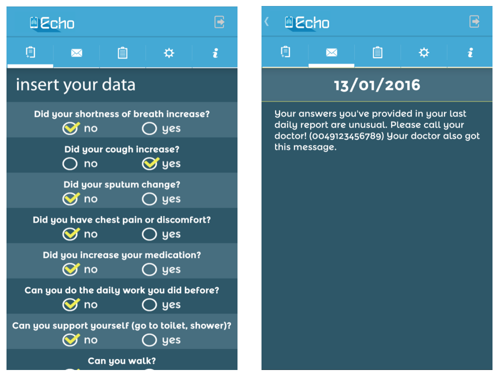


Figure 3. List of patients in the web application including an overview over all parts of the health record

of the patients' electronic health record by the doctor. Before the patient uses the mobile application, the physician can enter all relevant medical data into the system. Since the ECHO System can not replace the physician, it is still needed that the physician examines the patient from time to time. The results of those following examinations can also be added to the electronic health record of the patient. We will show how physicians' and patients' account can be created and how maintain the patients' electronic health record.

After the patient got introduced to the mobile application he can now fill in his data on a daily basis. This is the first of the dark-blue activities in Fig. 2. Fig. 4a shows the questionnaire on the smart phone which was already answered by the patient. After the submission of the report to the server, it is analyzed.

If the analytic functions detected a possibly worsening of the patients' health state, a notification is send to the physician and the patient via E-Mail, SMS or push notification. If there is a known treatment recommendation, the system will also send it along with the notification. Fig. 4b shows a notification in the mobile application. We will show how the mobile applications works and show the different outcomes of the analytics.



(a) (b)

Figure 4. Mobile Application: (a) shows the daily questionnaire (b) shows a notification received by the patient

Acknowledgments This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept "Innovationen mit Dienstleistungen" (01XZ13023G) and is managed by the Project Management Agency at the German Aerospace Center (PT-DLR) Unit "Work Design and Services" (PT-AD).

References

1. Bitsaki, M., et al.: An Integrated mHealth Solution for Enhancing Patients' Health Online. In: Proceedings of the 6th European Conference of the International Federation for Medical and Biological Engineering (MBEC'14). pp. 695–698. International Federation for Medical and Biological Engineering (IFMBE) (2014)
2. OASIS: Topology and Orchestration Specification for Cloud Applications (TOSCA) Version 1.0 (November 2013)
3. Steimle, F., et al.: Design and Implementation Issues of a Secure Cloud-Based Health Data Management System. In: Proceedings of the 9th Symposium and Summer School On Service-Oriented Computing. Technical Paper, vol. RC25564, pp. 68–82. IBM Research Report (2015)

Author Index