International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:6, No:6, 2012

Assessing Semantic Consistency of Business Process Models

Bernhard G. Humm, Janina Fengel

***Abstract*—**Business process modeling has become an accepted means for designing and describing business operations. Thereby, consistency of business process models, i.e., the absence of modeling faults, is of upmost importance to organizations. This paper presents a concept and subsequent implementation for detecting faults in business process models and for computing a measure of their consistency. It incorporates not only syntactic consistency but also semantic consistency, i.e., consistency regarding the meaning of model elements from a business perspective.

***Keywords*—**Business process modeling, model analysis, semantic consistency, Semantic Web

I. Introduction

BUSINESSES all over the world are faced with the challenge of having to flexibly react to change and to dynamically work with varying business partners. Continuous shaping and reshaping of business processes is a critical success factor for a business's competitiveness [1], [2]. Over the past decades, *business process modeling* has become an accepted means for designing and describing business operations in enterprises within and across company boundaries. Additionally, business process models may be transformed and executed by process engines as part of IT applications. Business process models describe interrelated business objects and business activities in a specific sequence, expressed in a certain modeling language with elements labeled in natural language [3], [4].

The *quality* of business process models is of upmost important to organizations – particularly their *consistency* [5]. The notion of consistency refers to the absence of modeling faults within single models, models that are interlinked within organizations as well as models that are interlinked between organizations.

In this paper, we present a concept and implementation for detecting faults in business process models and for computing a measure of consistency of business process models. It incorporates not only *syntactic consistency* but also *semantic consistency*, i.e., it also takes into account the meaning of model elements from a business perspective.

Automatedly computing consistency metrics has a number of benefits. Firstly, it may aid the business process modeler during the modeling process in order to avoid modeling faults. Secondly, it helps quality managers assess the measure of consistency of business process models – within and between companies.

Thirdly, it is a perfect and necessary complement to systems that (semi-)automatically align business process models (e.g., [6], [7].

This paper is structured as follows. Section II introduces necessary terminology. In Sections III and IV, we present the consistency metric and its implementation, explained by means of an example in Section V.

B. G. Humm and Janina Fengel are with Hochschule Darmstadt – University of Applied Sciences, Darmstadt, Germany (e-mail: bernhard.humm@h-da.de, janina.fengel@h-da.de)

Sections VI and VII present related work, conclusions and future work.

II. Terminology

A. Business Process Model

There are numerous definitions of *business process model* in the literature [8], [9], [10], [2]. For our purposes, it is sufficient to postulate the following characteristics of a business process model (or simply *model* if the context is clear):

1. It is defined in a business process modeling language like, e.g., BPMN, EPC, or UML activity model.
2. It consists of labeled nodes and edges as elements. The node and edge types are defined by the modeling language used.
3. The element labels express business logic in business-domain-specific natural language.

B. Consistency, Consistency Rules, and Faults

Consistency is a model quality characteristic and denotes the absence of faults within a model or a set of models. We distinguish two kinds of consistency:

- *Syntactic consistency* relates the usage and correct ordering of specific node and edge types within a model, independent of specific element labels, according to a certain modeling language grammar [11].
- *Semantic consistency* relates to the business logic expressed in element labels, i.e., the intended meaning of model elements and their domain-specific sequencing.

Consistency can individually be defined via consistency rules. A *consistency rule* is a regulation regarding model elements, their types, labels, order and interconnections to ensure consistency. Table I shows exemplary consistency rules.

TABLE I
Consistency rules examples

	General	Organization-specific
Syntactic	"Events and Functions must alternate" (EPC)	"Do not use Intermediate Multiples" (BPMN)
Semantic	"A data item may never be used before it has been created"	"A credit check must be performed before any financial transaction"

Consistency rules may be *general*, i.e., generally agreed in the business modeling community, or *organization-specific*.

General syntactic consistency rules are published in numerous articles and guidelines (e.g., [12], [13], [14], [15]. They are usually modeling-language specific. Examples for EPC are [13]:

- Events and Functions must alternate,
- Functions or Events must not have more than one outgoing or incoming connection,
- An XOR Split must not follow an Event.

Organization-specific syntactic consistency rules may, e.g., reduce the number of modeling element types to be used within an organization. For example, rarely used BPMN element types may be explicitly excluded, such as Intermediate Flow, Off-Page Connector, Intermediate Multiple, and Compensation Association.

General semantic consistency rules address faults that are independent of a particular business domain. Examples are:

- Activity labels must be of the form "verb phrase – noun phrase", e.g., "book flight" as opposed to "flight booking",
- Event labels must be of the form "noun phrase – verb participle", e.g., "booking information received",
- A data item may never be used before it has been created, e.g., "send invoice" before "create invoice".

Organization-specific semantic consistency rules are usually business-domain specific and/or company-specific. Examples are:

- All data item names must conform to a company-wide data dictionary, e.g., "book vehicle" instead of "book car" if "vehicle" is the term used in the data dictionary,
- A credit check must be performed before any financial transaction.

A *fault* denotes the violation of a consistency rule in a concrete model. Faults are categorized by *fault types*. An example for a fault would be an EPC function labeled "check credit card" that is immediately following another EPC function labeled "get travel data", as such a sequence violates the rule "Events and Functions must alternate". The fault type would be "function follows function".

C. Consistency Measures and Metrics

A *consistency measure* is the level of consistency of a concrete model. We express the *consistency measure* of a model as a numeric value between and including *0* and *1*. The measure *1* denotes the absence of any faults in the model. A smaller measure indicates a lower degree of consistency: the more faults, the lower the consistency measure.

A *consistency metric* is a formula to compute consistency measures for models.

D. Characteristics

In the following sections, we describe a consistency metric and its implementation with the following characteristics:

1) It incorporates syntactic as well as semantic consistency rules.
2) Consistency rules, particularly semantic ones, may be defined independently of the concrete business process modeling language used.
3) It is extensible in that new consistency rules, e.g., organization-specific ones, may be added.
4) Consistency measures and the faults detected can be explained.

III. Consistency Metric

The consistency metric *C* sums up the number of all faults (syntactic and semantic), weighted by the fault type and normalized by the mean weight and the model size, as shown in (1).

$$C\ (m) = \max\left(1 - \frac{\sum_{i=1}^{n} f_i(m)\ \cdot\ w_i}{|m|\ \cdot\ \sum_{i=1}^{n} w_i / n}, 0\right) \qquad (1)$$

where

- m : model to be evaluated for consistency,
- $f_i(m)$: number of faults of type i in model m,
- w_i : weight for fault type i as a numerical value > 0. Individual weights can be assigned statically to indicate severity, e.g., $w_{EPC\ function\ follows\ function} = 0.9$,
- n : number of fault types,
- $|m|$: number of elements in model m.

If there is no fault in model *m* then *C(m)=1*. Each fault reduces the consistency measure. If every model element has a fault, then *C(m)=0*. If, additionally, some elements have several faults, then the fraction would be greater than *1*. The use of the maximum function ensures that *C* has *0* as the lower bound and, thus, avoids negative consistency measures.

IV. Implementation

We have implemented the consistency metric and consistency rules in a research project called KINO (German acronym for "Artificial Intelligence for Enterprise Use"). Fig. 1 gives an overview of the processing steps to compute consistency measures.

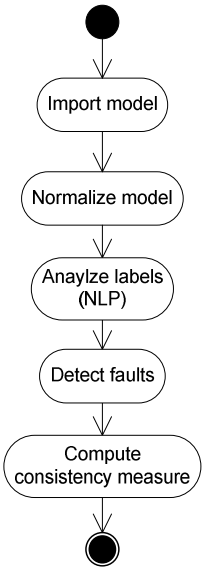

Fig. 1 Processing steps

Models can be imported from various formats. Nodes are then represented internally in a normalized format. Labels are analyzed via natural language processing (NLP). Then, faults are detected using consistency rules. Finally, the consistency measure can be computed.

A. Model Import

Business process models can be imported into the KINO

application from various formats, e.g., Star UML Activity model format [16], BFlow EPC native model format, and XMI [17]. The specific formats are transformed via XSLT into the internal KINO format.

B. Model Normalization

The KINO application follows a layered architecture, as shown in Fig. 2.

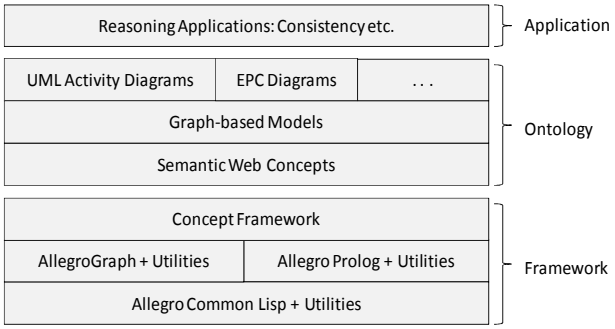

Fig. 2 KINO Layered Architecture

We use *Semantic Web* technology for representing and reasoning over business process models, namely AllegroGraph, AllegroProlog, and Allegro Common Lisp by Franz Inc. The *concept framework* described in [18] has been used to allow for the formulation of concise rules.

Data structures in Semantic Web applications are represented in *ontologies*. The ontology for business process models is organized in layers. A general ontology for graph-based models is constructed on top of basic Semantic Web concepts. It contains concepts like *node* and *edge*. Specific business process modeling languages like EPC and UML activity models are formulated on top of the ontology for graph-based models. An EPC function, for example, is modeled as a specialization of a node.

Reasoning applications as the consistency metric described in this paper may access various ontology layers.

Normalizing business process models takes place in two ways. Firstly, different formats of the same modeling language, e.g., BFlow EPC native model format, and XMI, are transformed into the same ontology format and, hence, can be compared. Secondly, models in different modeling languages can be compared since the ontology for the concrete modeling languages is based on the ontology for graph-based models. So, e.g., querying for all nodes and their labels is possible, regardless whether the nodes are UML activities or EPC functions.

C. Natural Language Processing

We use GATE (General Architecture for Text Engineering) [19] for natural language processing of labels. Thereby, a simple pipeline with the following processing resources is utilized:

- Tokenizer: identification of words,
- Part-of-Speech Tagger: categorization of words, e.g. noun, verb,
- Parser: identification of dependencies between words, e.g., verb phrase, noun phrase.

For synonym resolution WordNet is in place [20].

D. Fault Detection

Consistency rules are implemented in AllegroProlog. The main Prolog predicate is

```
fault (element fault-type reason)
```

The parameter `element` holds the URI (Uniform Resource Identifier) of a model element that is faulted with respect to the fault type in parameter `fault-type`. The parameter `reason` contains additional information about the detected fault. It may be used for explaining the consistency measure.

See, for example, the implementation of a *syntactic* consistency rule: flaw type *"EPC function follows function"*.

```
(<- (fault ?f2 "funct-follows-funct" ?f1)
  (epc-function ?f1 ?label1 ?model)
  (epc-function ?f2 ?label2 ?model)
  (follows ?f2 ?f1))
```

The rule reads as follows. There is a fault of type `"funct-follows-funct"` in element `?f2` if the following conditions hold:

1) `?f1` is an EPC function with some label `?label` in some model `?model`.
2) `?f2` is an EPC function in the same model `?model`.
3) `?f2` immediately follows `?f1`.

The implementation is straight forward and concisely expresses the consistency rule in a comprehensible way.

See as a second example the implementation of a *semantic* consistency rule of flaw type *"use before creation"*.

```
(<- (fault ?n1 "use-before-creation" ?n2)
  (node ?n1 ?label1 ?model)
  (node ?n2 ?label2 ?model)
  (follows-trans ?n2 ?n1)
  (identical-nouns ?label1 ?label2)
  (verb-synonym-to ?label2 "create"))
```

The rule reads as follows. There is a fault of type `"use-before-creation"` in `?n1` if the following conditions hold:

1) `?n1` is a node in some model `?model` – independent of the modeling language.
2) `?b2` is a node in the same model `?model`.
3) `?f2` follows transitively `?f1`, i.e., directly or indirectly.
4) The labels of `?f1` and `?f2`, `?label1` and `?label2`, have identical nouns, e.g., "invoice" in "create invoice" and "send invoice".
5) The verb in `?label2` is a synonym of the verb "create", e.g., "make" or "instantiate". `identical-nouns` and `verb-synonym-to` are Prolog predicates that use GATE for identifying nouns and verbs, respectively WordNet for synonym resolution.

The implementation of the semantic consistency rule is as straight forward as the syntactic one: concise and

comprehensible. It uses the ontology for graph-based models as opposed to a modeling-language specific ontology. Thereby, it is independent of a particular business process modeling language and, hence, can be used for models in all languages alike.

E. Consistency Metric

The consistency metric *C* is implemented as a Lisp function

```
consistent (model)
```

which returns the consistency measure of the parameter `model`. The function queries `fault` predicates for all fault types and computes the formula shown in (1).

V. Example

We show the application of the consistency metric by means of the example EPC diagram "travel reservation", adopted and modified from [21], as shown in Fig. 3.

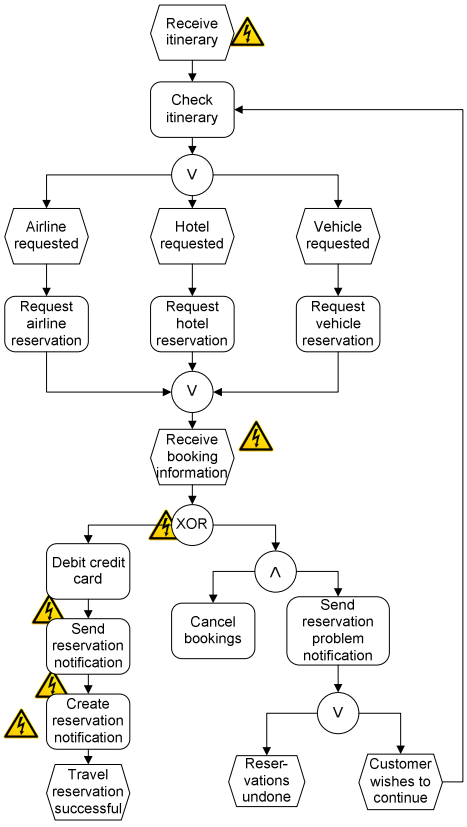

Fig. 3 Example business process model

The execution of the consistency rule `consistent !trvr:travel-reservation)` returns a consistency measure of `0.73`. This resulting measure 0.73 may be interpreted as roughly one quarter of the model being inconsistent. `consistent` returns, as second result, the explanation as a list of fault descriptions. It contains three syntactic faults and three semantic faults, each described by a list containing the faulty node, fault type, and reason:

```
((!trvr:Receive-itinerary
   "event-without-participle"
   "Receive")
(!trcr:Receive-booking-information
 "event-without-participle"
  "Receive")
(!trvr:xor-split
 "xor-split-after-event"
 !trvr:Receive-booking-information)
(!trvr-Send-reservation-notification
 "funct-follows-funct"
 !trvr:Debit-credit-card)
(!trvr-Create-reservation-notification
 "funct-follows-funct"
 !trvr-Send-reservation-notification)
(!trvr-Send-reservation-notification
 "use-before-creation"
 !trvr-Create-reservation-notification))
```

VI. Related Work

Reasoning over business process models is an active field of research since many years. Even though various guidelines for sound business process models have been developed over the years, e.g., [22], [23], models can differ considerably [24], [25] or may be inconsistent [26], [27]. As a means for detecting differences, reasoning is applied. For example, in [28], the use of rules is explored for supporting process design and for reasoning about process alternatives when redesigning processes. Recently, the use of ontologies for easing these tasks has been introduced. In [29], [30], and [31], ontologies are used for querying and reasoning over business process models in order to support process redesign. Such approaches are focusing on the tasks involved in creating or changing business process models and are interested in supporting the engineering of consistent models. In contrast to our approach, often manual efforts are needed for developing the domain-specific background information or the semantic annotation of the business process models. Furthermore, the results do not provide a measure for the results of consistency checks performed and, thus, do not allow an assessment of their quality. For assessing the quality of business process models and thereby also their consistency, various metrics have been developed. Mostly, these measures are developed concentrating on business process models of a certain type, e.g. EPC [32] or BPMN [33]. Thereby, research in the field of consistency regarding the model syntax using meta model rules has been done, for example in [9]. Some research proposes measures for business process models independently of the modeling language used [34]. These metrics support the assessment of quality, among them structural consistency. However, they do not consider semantic consistency as included in our suggested metric as well.

Semantic analysis concerning the domain language presently focuses on detecting similarity between models or part of models. Thereby, the aim is to facilitate restructuring or merging of models [35], [36], [37], [6].

Another application is the requirement of having to check models' compliance to rules and regulations [38]. However, general semantic consistency rules such as "a data item may never be used before it has been created" cannot be provided

as in our approach. Furthermore, differing use of the domain language leading to heterogeneity or ambiguity is not addressed. So far, metrics taking into account syntactic as well as semantic consistency have not yet been presented in the literature. In this, our approach of assessing consistency could complement the existing efforts in model analysis.

VII. Conclusions and Future Work

We have presented a concept for computing a consistency measure of business process models, taking into account syntactic as well as semantic consistency. We have implemented the approach as part of the KINO application for reasoning over business process models.

While there are many approaches for computing syntactic consistency in the literature, the analysis of semantic consistency is novel. Consistency rules may be implemented in Prolog in a straight forward, concise, and comprehensible way. This allows organizations to implement organization-specific consistency rules (syntactic and semantic) in addition to general rules that may be built into a business process modeling tool.

We plan the following extensions as future work:

- *Hierarchical and connected models*: most business process modeling languages offer mechanisms to hierarchically embed models within other models or to connect models. The proper handling of hierarchical or connected models needs to be implemented.
- *Semantic consistency via ontologies*: So far, semantic consistency rules may use results from natural language processing, e.g., synonym resolution, only. To improve expressibility, business-domain specific ontologies may be used. This will allow expressing consistency rules like "A credit check must be performed before any financial transaction" – which we cannot express so far.
- Integration: The concistency analysis may be integrated with other semantic analyses like, e.g., similarity, in business process modeling tools.

The concepts presented and the planned may, eventually, help improve the quality of business process models within organizations.

Acknowledgment

This research has been funded by zfe (Zentrum für Forschung und Entwicklung, Hochschule Darmstadt) under grant number 41932701.

References

[1] Frank, U.: Informationstechnologie und Organisation. In: Schreyögg, G., Werder, A.v. (eds.) Handwörterbuch Unternehmensführung und Organisation (HWO), pp. 472–480. Schäffer-Poeschel, S.l. (2004)

[2] Scheer, A.W., Nüttgens, M.: ARIS Architecture and Reference Models for Business Process Management. In: van Aalst, W.d. (ed.) Business process management. Models, techniques, and empirical studies, pp. 376–389. Springer, Berlin (2000)

[3] Weske, M.: Business Process Management. Concepts, Languages, Architectures Springer Berlin Heidelberg (2007)

[4] Scheer, A.-W.: ARIS - business process modeling Springer, Berlin (2000)

[5] Aagesen, G., Krogstie, J.: Analysis and Design of Business Processes Using BPMN. In: Vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1, pp. 213–235. Springer, Berlin Heidelberg (2010)

[6] Simon, C., Mendling, J.: Integration of Conceptual Process Models by the Example of Event-driven Process Chains. In: Oberweis, A., Weinhardt, C., Gimpel, H. (eds.) eOrganisation. Service-, Prozess-, Market-Engineering. Bd. 1. 8. Internationale Tagung Wirtschaftsinformatik (WI 2007), pp. 677–694. Univ.-Verl. Karlsruhe, Karlsruhe (2007)

[7] Gottschalk, F., van der Aalst, W.: Merging Event-Driven Process Chains. In: Meersman, R., Tari, Z. (eds.) On the move to meaningful internet systems: OTM 2008. OTM 2008 confederated international conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 2008 ; proceedings, part I, pp. 418–426. Springer, Berlin (2008)

[8] Schmelzer, H.J., Sesselmann, W.: Geschäftsprozessmanagement in der Praxis. Hanser, München (2008)

[9]. Mendling, J.: Metrics for Process Models. Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness Springer Berlin Heidelberg (2009)

[10] Harmon, P.: Business process change. A guide for business managers and BPM and six sigma professionals, Elsevier, Amsterdam (2007)

[11] Esswein, W., Gehlert, A., Seiffert, G.: Towards a Framework for Model Migration. In: Persson, A., Stirna, J. (eds.) Advanced information systems engineering. Proceedings of the 16th international conference (CAiSE 2004), pp. 463–476. Springer, Berlin (2004)

[12] Keller, G., Nüttgens, M., Scheer, A.W.: Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK). Technical Report: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes, Heft 89. Saarbrücken (1992)

[13] Aris Community: Event-driven process chain (EPC), http://www.ariscommunity.com/event-driven-process-chain, last accessed 30.01.12

[14] Ambler, S.W.: The elements of UML 2.0 style. Cambridge Univ. Press, Cambridge (2005)

[15] Allweyer, T.: BPMN 2.0. Introduction to the standard for business process modeling. Books on Demand, Norderstedt (2010)

[16] StarUML - The Open Source UML/MDA Platform, http://staruml.sourceforge.net/en/, last accessed 30.01.12

[17] bflow* Toolbox, http://sourceforge.net/projects/bflowtoolbox/, last accessed 30.01.12

[18] Humm, B., Korobov, A.: Introducing Layers of Abstraction to Semantic Web Programming. In: Robert Meersman, Tharam S. Dillon, Pilar Herrero (eds.) On the Move to Meaningful Internet Systems: OTM 2011 Workshops - Confederated International Workshops and Posters: EI2N+NSF ICE, ICSP+INBAST, ISDE, ORM, OTMA, SWWS+MONET+SeDeS, and VADER 2011, Hersonissos, Crete, Greece, October , 2011. Proceedings, pp. 412–423. Springer (2011)

[19] The University of Sheffield: GATE general architecture of text engineering, http://gate.ac.uk/, last accessed 30.01.12

[20] Princeton University: WordNet - About WordNet, http://wordnet.princeton.edu/, last accessed 30.01.12

[21] Havey, M.: Essential business process modeling. O'Reilly, Beijing (2005)

[22] Becker, J., Rosemann, M., Uthmann, C.: Guidelines of Business Process Modeling. In: van Aalst, W.d. (ed.) Business process management. Models, techniques, and empirical studies, pp. 30–49. Springer, Berlin (2000)

[23] Schütte, R.: Grundsätze ordnungsmäßiger Referenzmodellierung. Gabler, Wiesbaden (2001)

[24] Hadar, I., Soffer, P.: Variations in Conceptual Modeling: Classification and Ontological Analysis. Journal of the AIS 7, 568–592 (2006)

[25] Becker, J., Delfmann, P., Herwig, S., Lis, L., Malsbender, A., Stein, A.: Ein automatisiertes Verfahren zur Sicherstellung der konventionsgerechten Bezeichnung von Modellelementen im Rahmen der konzeptionellen Modellierung. In: Engels, G., Karagiannis, D., Mayr, H.C. (eds.) Modellierung 2010, pp. 49–65 (2010)

[26] Pablo A. Straub, Carlos Hurtado L: Behavioral Consistency in Business Process Models. Technical Report RT-PUC-DCC-95-4,Catholic University of Chile, Computer Science Department (1995)

[27] van Dongen, B.F., Mendling, J.: Structural Patterns for Soundness of Business Process Models. In: In EDOC '06: Proceedings of the 10th

IEEE International Enterprise Distributed Object Computing Conference (EDOC'06), pp. 116-128. IEEE Computer Society (2006)
[28] Yu, E.S., Mylopoulos, J.: Using Goals, Rules and Methods to Support Reasoning in Business Process Reengineering. Intelligent Systems in Accounting, Finance & Management 5 (1996)
[29] Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: Roddick, John F., Hinze, A. (eds.) Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM 2007), pp. 71–80. Ballarat, Victoria, Australia (2007)
[30] Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., Tonella, P.: Reasoning on Semantically Annotated Processes. In: Bouguettaya, A., Krueger, I., Margaria, T. (eds.) Service-Oriented Computing – ICSOC 2008, 5364, pp. 132–146. Springer Berlin Heidelberg (2008)
[31] Markovic, I.: Semantic business process modeling. Dissertation, KIT Scientific Publ, Karlsruhe, Karlsruher Inst. für Technologie (2009)
[32] Dijkman, R., van Dongen, B., Dumas, M., Käärik, R., Mendling, J.: Similarity of Business Process Models: Metrics and Evaluation. Journal of Information Systems 36(2): 498-516 (2011)
[33] Khlif, W., Makni, L., Zaaboub, N., Ben-Abdallah, H.: Quality metrics for business process modeling. In: Proceedings of the 9th WSEAS international conference on Applied computer science, pp. 195-200. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA (2009)
[34] Vanderfeesten, I., Cardoso, J., Mending, J., Reijers, H.A., van der Aals, W.: Quality Metrics for Business Process Models. In: Fischer, L. (ed.) BPM & Workflow Handbook. Methods, concepts, case studies and standards in business process management and workflow. Future Strategies, Lighthouse Point, Fla (2007)
[35] Yan, Z., Dijkman, R., Grefen, P.: Fast Business Process Similarity Search with Feature-Based Similarity Estimation. In: Dillon, T., Herrero, P. (eds.) On the move to meaningful internet systems: OTM 2010. Confederated international conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 2010 ; proceedings, part I, pp. 60–77. Springer, Berlin (2010)
[36] Kunze, M., Weske, M.: Metric Trees for Efficient Similarity Search in Large Process Model Repositories. In: Zur Mühlen, M., Su, J. (eds.) Business Process Management Workshops. International Workshops and Education Track. Springer, Berlin, Heidelberg (2011)
[37] La Rosa, M., Dumas, M., Käärik, R. and Dijkman, R.: Merging Business Process Models. Extended Version, BPM Center Report BPM-09-22, BPMcenter.org, bpmcenter.org/wp-content/uploads/reports/2009/BPM-09-22.pdf, last accessed 30.01.2012
[38] Marwane El Kharbili, Sebastian Stein, Ivan Markovic, Elke Pulvermüller: Towards a Framework for Semantic Business Process Compliance Management. In: Proceedings of the 1st International Workshop on Governance, Risk and Compliance - Applications in Information Systems (GRCIS'08) held in conjunction with the CAiSE'08 Conference, pp. 1–15. Vol-339. CEUR Workshop Proceedings (2008)