Managing Large Collections of Business Process Models - Current Techniques and Challenges

Remco Dijkman[a], Marcello La Rosa[b,c], Hajo A. Reijers[a]

[a] *Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands*
[b] *Queensland University of Technology, GPO Box 2434, Brisbane Qld 4001, Australia*
[c] *NICTA Queensland Lab, PO Box 6020, St Lucia Qld 4067, Australia*

Abstract

Nowadays, business process management is an important approach for managing organizations from an operational perspective. As a consequence, it is common to see organizations develop collections of hundreds or even thousands of business process models. Such large collections of process models bring new challenges and provide new opportunities, as the knowledge that they encapsulate requires to be properly managed. Therefore, a variety of techniques for managing large collections of business process models is being developed. The goal of this paper is to provide an overview of the management techniques that currently exist, as well as the open research challenges that they pose.

Key words: Business Process Management, Model Collection, Model Management

1. Introduction

As it becomes increasingly common for organizations to work in a process-oriented manner, single organizations may be dealing with collections of hundreds or thousands business process models. Examples of such collections, which are often mentioned in the literature, include: the BIT Process Library (735 process models) [1], the SAP Reference Model (604 process models) [2], a reference model for Dutch municipalities (around 600 process models) [3], IBM's Insurance Application Architecture (around 250 process models) [4] and Suncorp's process model repository for insurance (6,000+ process models) [5].

As organizations develop such large collections of business process models, new challenges and opportunities arise. Consequently, it is not surprising to see that also more and more research is done on the topic of process model collections. This is demonstrated by Figure 1, which shows the number of publications on the topic from the year 2000 onward. The graph in this figure was created by searching for different permutations of the words 'business process [model] collection' in Google Scholar and retaining those publications that indeed present techniques for managing a collection of business process models.

The goal of this paper is to introduce the topic of managing large collections of business process models and to provide an overview of the various management techniques that currently exist. To this end, we performed a literature study into these techniques, of which we present the results in this paper. In this way the paper serves as a comprehensive discussion of the current state of the art and as an introduction to this special issue on managing large collections of business process models.

The rest of this paper is structured as follows. Section 2 presents each of the techniques that we found in the literature study, Section 3 briefly introduces the papers in this special issue and Section 4 concludes the paper.

2. Management Techniques and Research Challenges

Various areas that relate to managing process model collections can be identified in the literature. This section presents nine of these, along with its associated techniques, the progress that has been made within each of them, and

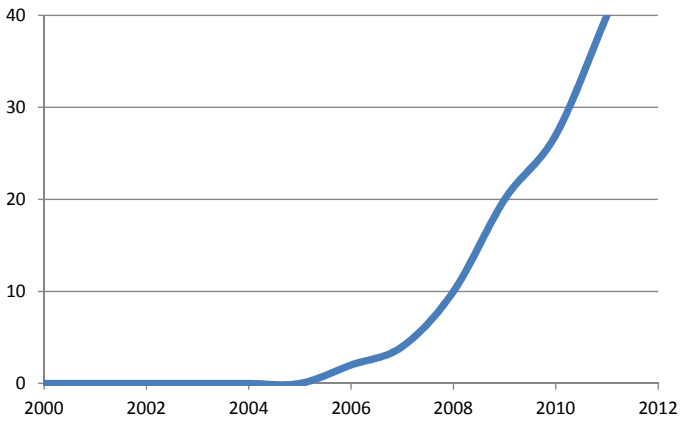

Figure 1: Publications on 'process model collections' since 2000.

Email addresses: r.m.dijkman@tue.nl (Remco Dijkman), m.larosa@qut.edu.au (Marcello La Rosa), h.a.reijers@tue.nl (Hajo A. Reijers)

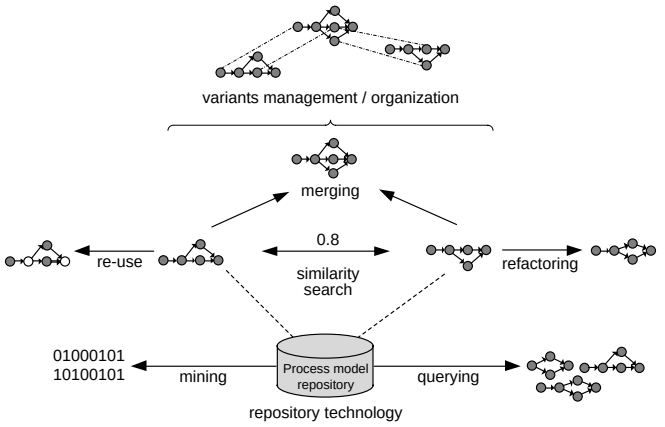

Figure 2: An overview of the discussed research areas.

the research challenges still left open. These areas are visualized in Figure 2, and are: querying, similarity search, variants management, merging, mining, refactoring, reuse, collection organization and repository technology.

2.1. Querying

Querying for a specific process model or model fragment within a (large) collection of organizational process models has multiple uses. It can be used to identify process models that do or do not comply with given standards or internal practices, process models that are candidate for refactoring, or process models that can be reused as a template to build new ones. In general, given a collection of business process models, querying can be used to retrieve models that have specific properties, such as a specific activity or a specific relation among activities that they contain.

There exist various approaches that can be used to express and execute queries over a collection of business process models [6, 7, 8, 9]. These approaches vary with respect to their expressive power. Most notably, there is a distinction between approaches that enable the formulation of a query as a business process model fragment [7, 8] and those that provide specific constructs for expressing process model queries [6, 9]. The first class of approaches aim to identify all models in a repository that contain the query fragment modulo similar activity labels. The second class uses a declarative approach whereby one can specify the existence or absence of specific (transitive) paths between process model activities. Queries can be formulated over control-flow aspects of process models [6, 7, 8], although some approaches also embrace other aspects, including the resources that are used to perform the process, the process trigger, the process goal, the location(s) at which the process is executed, or the categories in which the process is classified [7].

There are currently two ways in which queries are executed on a collection of business process models. The first way is to translate a query into a language that can be understood by the technology that is used to store the business process models. Considering that this technology is typically a relational database or an XML database, business process query languages are typically transformed into SQL [6] or XML query languages [7, 9]. The second way is to employ subgraph isomorphism algorithms [8]. In this case, indexes can also be used to speed up the query execution, by following a filter-and-search approach typical of graph databases.

2.2. Similarity Search

Given a collection of business process models and a single process model, similarity search returns those models from the collection that are similar to the input process model. It is related to querying, in that both querying and similarity search are used to search a collection for models with certain properties. The main difference between querying and similarity search is that querying searches for exact matches of the query to a part of a process model, while similarity searches for inexact matches of the query to a complete process model.

Similarity search can be applied in various practical situations in which similar business process models need to be found. For example, if an organization wants to implement standardized (reference) processes, similarity search can be used to determine which of its own processes fits best with which standardized process models.

The two main research streams in this area are: (i) the development of similarity measures that, given two process models, return the similarity of those two models, typically on a scale from 0 to 1 [10, 11, 12, 13, 14, 15]; and (ii) the development of efficient algorithms to find the models in a collection that are most similar to a given input model [11, 12, 14, 15, 16].

There exist a number of similarity measures, which can be classified based on the information that is used to determine whether two models are similar. Information that can be considered includes: labels, structure and behavior [13]. Label similarity is based on pairwise comparison of the activity labels in the business processes, structural similarity is based on comparison of the graph structure of the business processes and behavioral similarity is based on comparison of the behavior that is represented by the business processes, as it can, for example, be represented as a set of activity traces that are possible in the process model or an abstraction of the causal relations between the activities in the process model [11, 12].

Efficient algorithms for similarity search are currently being developed in four directions (not necessarily mutually exclusive or complete). First, there is an initiative to develop similarity measures in such a way that process models in a collection can be organized in a tree-based index, based on their similarity to each other [11, 12]. Second, there is an initiative to estimate the similarity of the search model to the models in a collection, using a computationally inexpensive metric [16, 14]. Third, there is an initiative to split a query model into smaller fragments [15], which can be compared computationally inex-

pensively and incrementally. Fourth, clustering techniques are being used to group similar process models, enabling the comparison of clusters of processes rather than each individual process separately [17, 18].

Since similarity search and querying are related, techniques for similarity search can in part also be used for querying process model collections and vice versa. This holds in particular for indexing techniques [8, 15]: indexing techniques that make similarity search more efficient can also be used to make querying more efficient and vice versa. However, the way in which indexing techniques can be used differs, due to the different nature of querying and similarity search: searching for partial models versus searching for complete models; and searching for exact matches versus searching for approximate matches.

One of the main open research challenges in the area of similarity search is determining similarity of process models that use synonymous or otherwise related terms. This challenge has been addressed with respect to synonyms [11, 12], but not with respect to other relations. In addition, efficient algorithms to determine similarity of process models with synonymous terms have not been developed. Another challenge, also party addressed [19], is that of matching process models that are defined at different levels of abstractions. In this case it is often required to match a single node in one model with a set of nodes in another model (1:m matches), or even a set of nodes in one model with a set of nodes in another model (n:m matches).

2.3. Variants Management

Process model collections are often the result of organizational mergers or acquisitions, or depict the business procedures of organizations that operate across different settings, e.g. different countries each with their own regulations. Thus, it is not uncommon that various variants of a same process model co-exist within the same collection. In such contexts, techniques are needed to keep track of such variants, understand their touch points and differences, and co-evolve them over time avoiding inconsistencies. These techniques fall under the area of *variants management* for process model collections.

A number of techniques are available to deal with the management of process model variants. They mainly differentiate into two classes: i) those that envisage the use of a single *consolidated model* to capture the whole set of process variants, and ii) those that keep the process variants separate. A consolidated model is essentially a process model which subsumes the (most significative) behavior of all variants. An important characteristic of this model is to incorporate *variation points* to distinguish the parts that are common to all variants (also known as *commonalities*) from those that are variant-specific (variability). The idea is to co-evolve the various process variants by working on a single artifact—the consolidated model—rather than on each variant separately, and then *configure* the consolidated model via its variation points, so as to obtain one of its input variants when needed.

There are various ways of representing variation points in process models. For example, some approaches rely on *configurable nodes* such as splits and joins to indicate where the various variant-specific paths branch out or merge [20, 21, 22, 23]. Other approaches attach domain parameters to nodes [24], mark nodes with stereotypes [25, 26], assign cardinalities to arcs and nodes [27] or use aspect-oriented principles [28] to capture variation points. Moreover, high-level interfaces such as questionnaire models [29] or feature diagrams [25, 26, 28] can be linked to consolidated models in order to facilitate the configuration of the latter via domain concepts. A variant of the notion of consolidated model is presented in [30] where a reference model is created which subsumes the most representative behavior of all variants.

The second class of techniques leaves the various variants separate, but provides an infrastructure to identify and keep track of their commonalities in order to maintain consistency across variants when updating them. [31] uses a notion of inheritance to derive variants from parent processes, where the common behavior is expressed declaratively by means of a query. A hierarchical structure to incrementally store process model variants based on process goals is illustrated in [32]. [33] proposes a version control technique to store common parts across process model variants only once. In this way, changes to one process variant can selectively be propagated to the affected variants, thus keeping the variants synchronized. Finally, [34] defines an algebra for identifying behavioral relations among process variants such as behavioral subsumption or generalization which could be used to build an infrastructure to manage the evolution of process variants.

The next section describes techniques to automate the construction of consolidated models by *merging* the various variants.

2.4. Merging

In the context of organizational mergers, acquisitions or restructurings, it may become useful to merge process models that formerly belonged to different organizations or branches/units into a single process model for that specific business context. This may stem from the need to standardize business operations or to rationalize an organization's IT infrastructure, in order to cut operational costs. Accordingly, process model merging is about merging a collection of process variants into a consolidated process model. Merging process models leads to a process model collection of reduced size since the input variants are no longer relevant and can be removed. A reduced size can in turn improve the maintainability of the collection as a whole.

Various techniques exist to merge process models. Here we classify them based on three aspects: i) process behavior, ii) label similarity and iii) language support. A first class of techniques require that the merged model

subsumes the behavior of all input models [35, 36, 37]. Accordingly, it should be possible to replay the behavior of each input model in the merged model. In these approaches the merged model is built by computing the set union of the nodes in the input models. A second class of techniques does not enforce the behavior-preservation requirement [38, 39, 40]. In other words, it is not guaranteed that the behavior of each input model can be correctly replayed in the merged model. Further requirements besides behavior-preservation are the ability of tracing back which input model each activity in the merged model comes from (traceability), and the ability to revert the merged model into one of its input models (reversibility) [5].

Another distinguishing aspect is whether notions of label similarity are taken into account during the merging. In fact some approaches [35, 37] allow the merging of activities from different models that have similar (but not identical) labels, whereas others only merge process models based on their identical labels [38, 39, 36, 40].

Finally, some techniques impose restrictions on the language used to capture the input models: they work only with block-structured process models ([38, 40]), or with specific languages like EPCs ([39, 35, 36]), while others impose no restrictions on the input models ([37]).

One of the main challenges in this area is managing the complexity of merged models. For example, when the collection of input models is too large and varied, the merged model may become too complex and may thus need to be simplified, so that it can still be used as a means of communication among its stakeholders.

2.5. Mining

While a highly active stream of research emphasizes the derivation of process models from event logs, also known as *process mining*, it is also possible to view a collection of process models itself as a source of data from which knowledge can be harvested. An approach in this respect aims to extract *business rules* from a collection of process models [41], where a business rule represents a business constraint that has been encoded in a process model. The research in [42] explores the mining opportunities that arise from the availability of large collections of similar process models.

Mining approaches can also be used to gain insights into the characteristics of artifacts that need to be aligned with the business processes that are captured as models. For example, for the identification and specification of software architecture components different mining approaches that work on process model collections are proposed [43, 44]. To circumvent the problem that a process model collection at times may be too small to conveniently reason about its properties, researchers have also looked at mining algorithms for the purpose of extracting its characterizing parameters [45]. Such parameters can then be used to expand the model collection as to facilitate other types of analyses, albeit under the restrictions that artificially created models impose.

A clear venue for further research is to look into mining approaches that combine the elements as found within process models with other sources of data (e.g. work instructions and data models).

2.6. Refactoring

The term *refactoring* usually refers to techniques from the software engineering discipline that can be used to restructure source code or databases without changing their behavioral or informational semantics. The rationale behind refactoring is that the quality of such artifacts can be substantially improved, in particular in terms of their maintainability and readability.

Various researchers have recognized that a collection of process models, as it evolves over time, at some point may start to display unnecessary internal complexities as well. This has led to the development of techniques to detect and follow up on refactoring opportunities within process model collections, also known as *smells* [46]. A prominent example of a potentially undesirable state of a process model collection is given by the existence of process model replicas or *clones* [47]. Clones may hint at unintended redundancy, which negatively affects the consistency and maintainability of the collection as a whole. The detection of such clones are dealt with in works like [48, 46, 49, 50, 47]. Actual remedies for undesirable internal properties of process model collections can be found in [46, 51, 52, 53]; these papers also often offer a reflection on the various types of smells that can be encountered.

While the previously mentioned techniques are aimed at the retrospective detection and correction of undesirable characteristics of a process model collection, in other research efforts the emphasis is on preventing the occurrence of specific smells altogether. A good example of an undesirable aspect of a process model collection is that the activities within its process models are improperly labeled [46]. The potential drawbacks may be decreased comfort in using such process models and their incorrect interpretation. Approaches that promote the use of a glossary [54, 55] or a homogenous naming of labels [54, 33, 55] when creating process models can be seen as ways to circumvent this particular issue. Similarly, issues that involve overly complex structures of process models (see [46, 53]) can be preempted when following approaches that promote the homogeneity of process model structure [56, 57, 33].

While the detection of refactoring opportunities can be automated to a high extent, the proper application of refactoring techniques is at this point mostly an activity that requires the involvement of a human expert. This is an area where considerable progress can still be made towards "self-healing" process model collections. While the use of process model collections increases, it is also safe to expect that our insights into undesirable side-effects of the evolvement of such collections will grow. This, in turn, will require the development of new detection and correction techniques. Initial work in this direction includes,

for example, an algorithm for the automatic correction of process models which suffer from behavioral errors [58].

2.7. Re-use

A collection of process models can serve as a valuable source for new process models, especially when existing models contain behavior that also occurs in other processes (e.g. the way customers are billed). An approach to implement this idea is, for example, presented in [59]: The process modeling environment supports a modeler to browse through fragments and incorporate these in a new business process model. In this work specifically, recommendations on the value of such fragments or their popularity can guide the modeler towards choosing the most appropriate fragments. Related approaches that also support a modeler in selecting process model fragments as part of their modeling effort are described in [60, 61]. The ICT knowledge base that is presented in [62] offers support for re-use beyond process model fragments (e.g. service or IT systems specifications).

The emphasis on the re-use of *entire* process model structures is clearly what is behind the popularity of reference process models. The approach behind the MIT Handbook is slightly different, since it allows a user to navigate through a large collection of process models to arrive at a process model that fits with the domain of choice as well as with the desirable level of abstraction [63]. This is a good illustration of an approach that aims to achieve the re-use of process models in full.

Interestingly, reference process models can be seen as valuable artifacts to be preserved and re-used *themselves*. In [64] a framework is presented for the storage of a collection of reference process models, each of which can be re-used as source for new business process models or as an element of overarching enterprise architectures.

While not aiming to re-use process models or parts thereof, it is worth to mention approaches for the preservation of artifacts that are closely related to the enactment of such models. For example, *instances* of process execution may be re-used to achieve a desirable operational performance in a situation that resembles a historic occasion [15]. Storing the *changes* that have been made to process models (workflow schemas) may offer insights into their impact for users that consider similar model changes [65]. These approaches hint at the value of supporting users with contextual information when they consider the re-use of process models or their fragments and may be at the basis for developing more sophisticated re-use features.

2.8. Collection Organization

Given a collection of business process models, techniques are required to keep the overview of the collection, such that users can browse the collection to easily find the processes they are looking for. A collection of process models is organized around relations that these processes have with each other.

There exist various possible relations among business process models of a collection. These relations include the aggregation and the generalization relations [66, 67]. An aggregation relation exists between a business process model and it parts (usually sub-processes) and the generalization relation exists between a more general process model and a more specific one. Aggregation and generalization are typically used to develop a hierarchical classification of processes models, enabling users to navigate a collection of process models by traversing the hierarchy. Alternative ways of developing a hierarchical classification of process models also exist [68]. Another frequently used relation is that among different versions of business processes [7, 69, 70, 71, 72, 33]. When a process model has an aggregation relation with other process models, this aggregation relation can be extended with version information. This makes it possible, for example, to express relations such as: version 1.02 of business process A has version 1.10 of business process E and version 1.21 of business process G as its subprocesses. This is also called configuration management. The last relation is that among different process model variants within the same collection, which we already described in Section 2.3 when discussing techniques for managing the evolution of process model variants as separate artifacts.

Once relations among business process models are stored, they need to be enforced and they can be used to visualize (coherent parts of) the collection. A number of techniques exist to visualize parts of the collection. Enforcing has been proposed for example in [33] via a change propagation mechanism: Each process model can be assigned a propagation policy based on which a change in a process model may or may not be propagated to all models that are potentially affected by that change. Navigating a collection is the activity of traversing the relations that the different process models, or classes of models, have with each other [73]. Filtering a collection is the activity of showing only relevant detail. Filtering is a weak form of view management [71, 72]. A view is a visualization of certain relevant parts of a collection. However, views support more than filtering. In particular, while filtering of a collection can be done arbitrarily, views are typically predefined and targeted towards a specific stakeholder. In addition, relations between a view and the process model which the view belongs to, are typically maintained. This allows changes that are made to the view to be propagated to the process model.

2.9. Repository Technology

Repository technology provides the actual infrastructure for storing a collection of process models. In addition, repositories are meant to support many of the management techniques that are presented in this paper, such that they serve as the central point in an organization, from which the collection of business process models can be managed.

A large number of business process model repositories exist [74, 75, 76, 7, 69, 77, 78, 9, 79, 15, 70, 80, 81, 82, 71,

72, 83, 84], which vary with respect to the management techniques and the storage facilities that they support.

In previous work [85], we created an overview of the management techniques that business process model repositories can support. These include management techniques that apply to the management of models in their relation to each other as well as management techniques that apply to single models. The management techniques that apply to the management of models in their relation to each other are also addressed in this paper. They include: search (Section 2.2), querying (Section 2.1), navigation and view, version and configuration management (Section 2.8). Management techniques that apply to single models include import, export, check-in and check-out of models.

A repository can provide storage facilities for business process models in their internal and their external forms. The external form is the file format by which business process models are imported and exported. The internal form is a format that is optimized for the internal efficiency of the repository and that is not exposed outside of the repository. Consequently, when models are stored in an internal format, they need to be converted from the external to the internal format when importing them and vice-versa when exporting them. Models may be stored in the repository in their internal format, in their external format, or in both. In addition, a repository can store indexes on process models that can be used for efficiently searching or querying models. Finally, a repository can store additional information that is used by, but is not part of, the models. An example of such data is an ontology of terms that are relevant to the domain to which the business process models apply. The terms in the ontology can be used when constructing the models and the relations among the terms can be used when performing a similarity search, using the relations to identify similar terms and thus similar tasks.

3. In this special issue

This section briefly introduces each of the five papers that were selected for this special issue. The first paper presents a management technique for the extraction of recurring fragments, which can provide input to refactoring activities or be reused to design new models. The second and third paper address variant management, where the third paper also presents a management technique for organizing a collection. The fourth and the fifth paper deal with management techniques for similarity search.

3.1. Action Patterns in Business Process Model Repositories

Sergey Smirnov, Matthias Weidlich, Jan Mendling and Mathias Weske define the concept of 'action pattern' in their paper as a recurring combination of business process tasks. *Action patterns* can be used to capture re-usable knowledge about business process models. By facilitating re-use, they also help to improve the quality of collections of business process models. The paper presents a framework that classifies different types of action patterns and it shows how action patterns of different types can be mined in a collection. The paper evaluates the potential use of action patterns by showing how often and with how much confidence action patterns occur in a model collection from practice. The paper also shows the potential for re-using action patterns by evaluating how often patterns that are mined in one collection also appear in another collection of process models.

3.2. Design and Management of Flexible Process Variants using Templates and Rules

Akhil Kumar and Wen Yao propose a novel approach for managing business process variants. Their approach is based on the specification of a single consolidated process, which they call the *template process*, and rules that define how, and under what conditions, variants deviate from that template process. The benefit of their approach is that the template process can be kept relatively simple, compared to other approaches for managing business process variants, while their approach also maintains the traceability between each variant and the template process. The approach consists of a language for specifying the rules that define how and under what conditions a variant deviates from the process template and an individualization algorithm that creates the various process variants, given the process template and the configuration rules.

3.3. Multi-Abstraction Layered Business Process Modeling

The paper by Dieter Van Nuffel and Manu De Backer presents a framework that focuses on modeling the organization of a collection of business process models. The concepts that they present can be used to create different views of a process model collection, represent different process variants and create a classification of business process models according to their functionality and the motivation with which they were modeled. As a proof-of-concept the framework is applied to a collection of process models from the financial services industry.

3.4. A Comparative Survey of Business Process Similarity Measures

Michael Becker and Ralf Laue present a comprehensive survey of the different measures that currently exist for measuring the similarity of business process models. They compare a total of 22 similarity measures, which they identify through literature study and classify them into four classes, depending on the aspects of a process model they operate with. Besides these, they compare the performances of each similarity measure against the same collection of business process models. They also determine to which extent certain theoretical properties hold for the similarity measures under analysis.

3.5. Comparison and Retrieval of Process Models using Related Cluster Pairs

The paper by Michael Niemann, Melanie Siebenhaar, Stefan Schulte and Ralf Steinmetz presents a novel approach for measuring the similarity of business process models. The approach works by first identifying which *clusters* (groups of tasks) in one process model are similar to which clusters in the other process model, Next, by using the similarity of the clusters in the process models they determine the similarity of the process models as a whole. The paper shows that the approach outperforms existing techniques for measuring process model similarity. An added benefit of using this approach is that the similar clusters that are identified by the approach also determine which task in one process model corresponds to which task in the other process model. This is also called *matching*. The ability to match tasks is a prerequisite to many existing business process model analysis techniques, such as equivalence and compatibility analyses.

4. Conclusion

This paper introduced the topic of managing large collections of business process models, by presenting a set of management techniques that have been addressed in the literature. It then introduced the papers of this special issue according to the presented techniques and further elaborated on these techniques. Thus, this paper and this special issue present the state-of-the-art in the area of managing large collections of business process models and we hope that they will trigger further research in this area.

References

[1] D. Fahland, C. Favre, B. Jobstmann, J. Koehler, N. Lohmann, H. Völzer, K. Wolf, Instantaneous soundness checking of industrial business process models, in: Proc. of BPM, Vol. 5701 of LNCS, Springer, 2009, pp. 278–293.

[2] T. Curran, G. Keller, SAP R/3 Business Blueprint - Business Engineering mit den R/3-Referenzprozessen, Addison-Wesley, Bonn, Germany, 1999.

[3] Documentair structuurplan, `http://www.model-dsp.nl/` (Accessed: 24 Feb. 2010).

[4] IBM Insurance Application Architecture, `http://www.ibm.com/solutions/sg/insurance/enterprise_aa/tech_details.html` (Accessed: 24 Feb. 2010).

[5] M. La Rosa, M. Dumas, R. Uba, R. M. Dijkman, Business process model merging: An approach to business process consolidation, QUT ePrints 38241, Queensland University of Technology (2010).

[6] A. Awad, Bpmn-q: A language to query business processes, in: Proceedings of EMISA 2007, 2007, pp. 115–128.

[7] I. Choi, K. Kim, M. Jang, An xml-based process repository and process query language for integrated process management, Knowledge and Process Management 14 (4) (2007) 303–316.

[8] T. Jin, J. Wang, N. Wu, M. La Rosa, A. ter Hofstede, Efficient and accurate retrieval of business process models through indexing, in: Proceedings of the 18th CoopIS, Part I, Vol. 6426 of LNCS, Spinger, 2010, pp. 402–409.

[9] C. Beeri, A. Eyal, S. Kamenkovich, T. Milo, Querying business processes, in: Proceedings of the 32nd international conference on Very large data bases (VLDB), 2006, pp. 343–354.

[10] M. Dumas, L. G.-B. nuelos, R. Dijkman, Similarity search of business process models, Bulletin of the Technical Committee on Data Engineering 32 (3) (2009) 23–28.

[11] M. Kunze, M. Weidlich, M. Weske, Behavioral similarity – a proper metric, in: Business Process Management, Vol. 6896 of LNCS, Springer, 2011, pp. 166–181.

[12] M. Kunze, M. Weske, Metric trees for efficient similarity search in process model repositories, in: BPM 2010 Workshops, Vol. 66 of LNBIP, Springer, 2011, pp. 535–546.

[13] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, J. Mendling, Similarity of business process models: Metrics and evaluation, Information Systems 36 (2) (2011) 498–516.

[14] B. Mahleko, A. Wombacher, Indexing business processes based on annotated finite state automata, in: IEEE International Conference on Web Services, IEEE Computer Society, Los Alamitos, CA, USA, 2006, pp. 303–311.

[15] R. Lu, S. Sadiq, G. Governatori, On managing business processes variants, Data & Knowledge Engineering 68 (7) (2009) 642–664.

[16] Z. Yan, R. Dijkman, P. Grefen, Fast business process similarity search with feature-based similarity estimation, in: On the Move to Meaningful Internet Systems: OTM 2010, Vol. 6426 of LNCS, Springer, 2010, pp. 60–77.

[17] S. Kabicher, S. Rinderle-Ma, L. T. Ly, Activity-oriented clustering techniques in large process and compliance rule repositories, in: BPM 2011 Workshops (to appear), LNBIP, Springer, 2012.

[18] M. Qiao, R. Akkiraju, A. J. Rembert, Towards efficient business process clustering and retrieval: combining language modeling and structure matching, in: Proc. of BPM, Vol. 6896 of LNCS, Springer, Berlin, Heidelberg, 2011, pp. 199–214.

[19] M. Weidlich, R. Dijkman, J. Mendling, The ICoP framework: Identification of correspondences between process models, in: Proc. of CAiSE, Vol. 6051 of LNCS, Springer, 2010.

[20] M. Rosemann, W. M. P. van der Aalst, A Configurable Reference Modelling Language, Information Systems 32 (1) (2007) 1–23.

[21] F. Gottschalk, W. M. P. van der Aalst, M. H. Jansen-Vullers, M. L. Rosa, Configurable Workflow Models, International Journal of Cooperative Information Systems 17 (2) (2008) 177–221.

[22] M. La Rosa, M. Dumas, A. ter Hofstede, J. Mendling, Configurable Multi-Perspective Business Process Models, Information Systems 36 (2) (2011) 313–340.

[23] A. Hallerbach, T. Bauer, M. Reichert, Capturing Variability in Business Process Models: The Provop Approach, Journal of Software Maintenance and Evolution: Research and Practice 22 (6-7) (2010) 519–546.

[24] J. Becker, P. Delfmann, R. Knackstedt, Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models, in: J. Becker, P. Delfmann (Eds.), Reference Modeling: Efficient Information Systems Design Through Reuse of Information Models, Springer, 2007, pp. 27–58.

[25] K. Czarnecki, M. Antkiewicz, Mapping Features to Models: A Template Approach Based on Superimposed Variants, in: Proceedings of the 4th Int. Conference on Generative Programming and Component Engineering, Vol. 3676 of LNCS, 2005, pp. 422–437.

[26] A. Schnieders, F. Puhlmann, Variability Mechanisms in E-Business Process Families, in: W. Abramowicz, H. Mayr (Eds.), Proceedings of the 9th International Conference on Business Information Systems (BIS'06), Vol. 85 of LNI, GI, 2006, pp. 583–601.

[27] I. Reinhartz-Berger, P. Soffer, A. Sturm, Organizational Reference Models: Supporting an Adequate Design of Local Business Processes, International Journal of Business Process Integrations and Management 4 (2) (2009) 134–149.

[28] I. Machado, R. Bonifácio, V. Alves, L. Turnes, G. Machado, Managing variability in business processes: an aspect-oriented approach, in: Proceedings of the 2011 international workshop on Early aspects, ACM, New York, NY, USA, 2011, pp. 25–30.

[29] M. La Rosa, W. M. P. van der Aalst, M. Dumas, A. Hofstede,

Questionnaire-based Variability Modeling for System Configuration, Software and Systems Modeling(forthcoming).
[30] C. Li, Mining process model variants: Challenges, techniques, examples, Ph.D. thesis, University of Twente, The Netherlands (2010).
[31] E. Pascalau, A. Awad, S. Sakr, M. Weske, On maintaining consistency of process model variants, in: Business Process Management Workshops, LNBIP, Springer, 2010, pp. 289–300.
[32] W. Derguech, G. Vulcu, S. Bhiri, An indexing structure for maintaining configurable process models, in: Proc. of BMMDS/EMMSAD, Vol. 50 of LNBIP, Springer, 2010, pp. 157–168.
[33] C. C. Ekanayake, M. La Rosa, A. H. ter Hofstede, M.-C. Fauvet, Fragment-based version management for repositories of business process models, Vol. Proc. of CoopIS of LNCS, Springer, 2011, pp. 20–37.
[34] M. Weidlich, J. Mendling, M. Weske, A foundational approach for managing process variability, in: CAiSE, Vol. 6741 of LNCS, Springer, 2011, pp. 267–282.
[35] F. Gottschalk, W. M. P. van der Aalst, M. H. Jansen-Vullers, Merging event-driven process chains, in: Proc. of CoopIS, Vol. 5331 of LNCS, Springer, 2008, pp. 418–426.
[36] H. A. Reijers, R. S. Mans, R. A. van der Toorn, Improved Model Management with Aggregated Business Process Models, Data and Knowledge Engineering 68 (2) (2009) 221–243.
[37] M. La Rosa, M. Dumas, R. Uba, R. Dijkman, Merging business process models, in: On the Move to Meaningful Internet Systems: OTM 2010, Vol. 6426 of LNCS, Springer, 2010, pp. 96–113.
[38] S. Sun, A. Kumar, J. Yen, Merging workflows: A new perspective on connecting business processes, Decision Support Systems 42 (2) (2006) 844–858.
[39] J. Mendling, C. Simon, Business process design by view integration, in: Proc. of BPM Workshops, Vol. 4103 of LNCS, Springer, 2006, pp. 55–64.
[40] C. Li, M. Reichert, A. Wombacher, The minadept clustering approach for discovering reference process models out of process variants, Int. J. Cooperative Inf. Syst. 19 (3-4) (2010) 159–203.
[41] J. Polpinij, A. K. Ghose, H. K. Dam, Business rules discovery from process design repositories, in: IEEE Congress on Services, 2010, pp. 614–620.
[42] J. Buijs, B. van Dongen, W. M. P. van der Aalst, Towards cross-organizational process mining in collections of process models and their executions, in: 1st Int. Workshop on Process Model Collections (PMC 2011), LNBIP, Springer, 2012 (to appear).
[43] X. Zhao, Y. Zou, J. Hawkins, B. Madapusi, A business-process-driven approach for generating e-commerce user interfaces, in: Model Driven Engineering Languages and Systems, Vol. 4735 of LNCS, Springer, 2007, pp. 256–270.
[44] X. Zhao, A business process driven approach for automatic generation of business applications, Ph.D. thesis, Queen's University, Canada (2011).
[45] K. van Hee, M. La Rosa, Z. Liu, N. Sidorova, Discovering characteristics of stochastic collections of process models, in: Proc. of BPM 2011, Vol. 6896 of LNCS, Springer, 2011, pp. 298–312.
[46] B. Weber, M. Reichert, J. Mendling, H. A. Reijers, Refactoring large process model repositories, Computers in Industry 62 (5) (2011) 467–486.
[47] R. Uba, M. Dumas, L. Garcia-Banuelos, M. La Rosa, Clone detection in repositories of business process models, in: Proc. of BPM, Vol. 6896 of LNCS, Springer, 2011, pp. 248–264.
[48] R. Dijkman, B. Gfeller, J. Küster, H. Völzer, Identifying refactoring opportunities in process model repositories, Information and Software Technology 53 (9) (2011) 937–948.
[49] J. Guo, Y. Zou, Detecting clones in business applications, in: Reverse Engineering, Working Conference on, IEEE Computer Society, Los Alamitos, CA, USA, 2008, pp. 91–100.
[50] J. Guo, K. C. Foo, L. Barbour, Y. Zou, A business process explorer: recovering and visualizing e-commerce business processes, in: Proceedings of the 30th international conference on Software engineering, ACM, New York, NY, USA, 2008, pp. 871–874.
[51] B. Weber, M. Reichert, Refactoring process models in large process repositories, in: Advanced Information Systems Engineering, Vol. 5074 of LNCS, Springer, 2008, pp. 124–139.
[52] M. La Rosa, A. H. M. ter Hofstede, P. Wohed, H. A. Reijers, J. Mendling, W. M. P. van der Aalst, Managing process model complexity via concrete syntax modifications, IEEE Transactions on Industrial Informatics 7 (2) (2011) 255–265.
[53] M. La Rosa, P. Wohed, J. Mendling, A. H. M. ter Hofstede, H. A. Reijers, W. M. P. van der Aalst, Managing process model complexity via abstract syntax modifications, IEEE Transactions on Industrial Informatics 7 (4) (2011) 614–629.
[54] N. Peters, M. Weidlich, Automatic generation of glossaries for process modelling support, in: Enterprise Modelling and Information Systems Architectures (EMISA), Vol. 6, 2011, pp. 30–46.
[55] A. Weissgerber, Semantically-enriched business process modeling and management, Ph.D. thesis, Universität des Saarlandes, Germany (2011).
[56] S. Smirnov, M. Weidlich, J. Mendling, M. Weske, Action patterns in business process models, in: Service-Oriented Computing, Vol. 5900 of LNCS, Springer, 2009, pp. 115–129.
[57] S. Smirnov, M. Weidlich, J. Mendling, M. Weske, Object-sensitive action patterns in process model repositories, in: Proceedings of BPM 2010 Workshops, Vol. 66 of LNBIP, Springer, 2010, pp. 251–263.
[58] M. Gambini, M. La Rosa, S. Migliorini, A. ter Hofstede, Automated error correction of business process models, in: Proc. of BPM, Vol. 6896 of LNCS, Springer, 2011, pp. 148–165.
[59] A. Koschmider, M. Song, H. A. Reijers, Social software for business process modeling, Journal of Information Technology 25 (3) (2010) 308–322.
[60] T. Madhusudan, J. Zhao, B. Marshall, A case-based reasoning framework for workflow model management, Data & Knowledge Engineering 50 (1) (2004) 87–115.
[61] I. Markovic, A. C. Pereira, Towards a formal framework for reuse in business process modeling, in: Proceedings of the 2007 international conference on Business process management, Vol. 4714 of LNCS, Springer, 2007, pp. 484–495.
[62] G. Pignatelli, G. Motta, G. Germani, Ict portfolio: Mapping business and ict services, in: E-Government Ict Professionalism and Competences Service Science, Vol. 280 of IFIP Advances in Information and Communication Technology, Springer, 2008, pp. 299–307.
[63] T. Malone, K. Crowston, G. Herman, Organizing business knowledge: the MIT process handbook, the MIT Press, 2003.
[64] D. Jacobs, P. Kotze, A. Van Der Merwe, Towards an enterprise repository framework, in: 1st International Workshop on Advanced Enterprise Repositories (AER 2009), 2009, pp. 77–89.
[65] S. Rinderle, M. Reichert, M. Jurisch, U. Kreher, On representing, purging, and utilizing change logs in process management systems, Business Process Management (2006) 241–256.
[66] T. Malone, et. al., Tools for inventing organizations: Toward a handbook of organizational processes, Manage. Sci. 45 (1999) 425–443.
[67] T. Kurniawan, A. Ghose, L.-S. Le, H. K. Dam, On formalizing inter-process relationships, in: 1st Int. Workshop on Process Model Collections (PMC 2011), LNBIP, Springer, 2012 (to appear).
[68] P. Fettke, P. Loos, J. Zwicker, Business process reference models: Survey and classification, in: BPM Workshops, Vol. 3812 of LNCS, Springer, 2006, pp. 469–483.
[69] C. Liu, X. Lin, X. Zhou, M. Orlowska, Building a repository for workflow systems, in: Proceedings of Technology of Object-Oriented Languages and Systems, 1999, pp. 348–357.
[70] Z. Ma, B. Wetzstein, D. Anicic, S. Heymans, Semantic business process repository, in: Proceedings of SBPM, 2007, pp. 92–100.
[71] D. Schumm, D. Karastoyanova, O. Kopp, F. Leymann, M. Sonntag, S. Strauch, Process Fragment Libraries for Easier and Faster Development of Process-based Applications, Journal of Systems Integration 2 (1) (2011) 39–55.

[72] S. Gao, J. Krogstie, A repository architecture for business process characterizing models, in: Proceedings of PoEM 2010, Vol. 68 of LNBIP, Springer, 2010, pp. 162–176.
[73] M. Hipp, B. Mutschler, M. Reichert, Navigating in process model collections: A new approach inspired by google earth, in: 1st Int. Workshop on Process Model Collections (PMC 2011), LNBIP, Springer, 2012 (to appear).
[74] P. Bernstein, C. Dellarocas, T. Malone, J. Quimby, Software tools for a process handbook, IEEE Data Engineering Bulletin 18 (1) (1995) 41–48.
[75] S. Fiorini, J. Leite, C. Lucena, Process reuse architecture, in: Proceedings of CAiSE 2001, 2001, pp. 284–298.
[76] G. Yang, Process library, Data & Knowledge Engineering 50 (2004) 35–62.
[77] D. Vanderhaeghen, P. Loos, Distributed model management platform for cross-enterprise business process management in virtual enterprise networks, International Journal of Intelligent Manufacturing 18 (2007) 553–559.
[78] G. Decker, H. Overdick, M. Weske, Oryx : An open modeling platform for the bpm community, in: Proceedings of BPM 2008, 2008, pp. 382–385.
[79] A. Wasser, M. Lincoln, R. Karni, Processgene query - a tool for querying the content layer of business process models, in: Proceedings of BPM 2006, 2006, pp. 1–8.
[80] J. Vanhatalo, J. Koehler, F. Leymann, Repository for business processes and arbitrary associated metadata, in: Proceedings of BPM 2006, 2006, pp. 25–31.
[81] M. Elhadad, M. Balaban, Effective business process outsourcing: The prosero approach, International Journal of Interoperability in Business Information Systems 3 (1).
[82] R. Weber, C. Schuler, P. Neukomm, H. Schuldt, Web service composition with o'grape and osiris, in: Proceedings of VLDB 2003, 2003, pp. 1081–1084.
[83] H. J. Wang, H. Wu, Supporting process design for e-business via an integrated process repository, Information Technology Management 12 (2011) 97–109.
[84] M. La Rosa, H. A. Reijers, W. M. P. van der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, L. Garcia-Banuelos, Apromore: An advanced process model repository, Expert Systems with Applications 38 (6) (2011) 7029–7040.
[85] Z. Yan, P. Grefen, A framework for business process model repositories, in: Business process management workshops, LNBIP, Springer, 2010, pp. 559–570.