

A framework for business process model repositories

Citation for published version (APA):

Yan, Z., & Grefen, P. W. P. J. (2010). A framework for business process model repositories. In M. Muehlen, zur, & J. Su (Eds.), *Business process management workshops* (pp. 559-570). (Lecture Notes in Business Information Processing). Springer. <https://doi.org/10.1007/978-3-642-20511-8>

DOI:

[10.1007/978-3-642-20511-8](https://doi.org/10.1007/978-3-642-20511-8)

Document status and date:

Published: 01/01/2010

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

A Framework for Business Process Model Repositories

Zhiqiang Yan and Paul Grefen

Eindhoven University of Technology, The Netherlands
`{z.yan,p.w.p.j.grefen}@tue.nl`

Abstract. Large organizations often run hundreds or even thousands of business processes. Managing such large collections of business processes is a challenging task. Intelligent software can assist in that task by providing common repository functions such as storage, search and version management. They can also provide advanced functions that are specific for managing collections of process models, such as managing the consistency of public and private processes and extracting knowledge from existing processes to better design new processes. This paper, by analyzing existing business process model repositories, proposes a framework for repositories that assist in managing large collections of business process models. The framework consists of a management model and a reference architecture. The management model lists the functionality that can be provided by business process model repositories. The reference architecture presents the components that provide this functionality and their interconnections. The framework provides a reference model for analysis and extension of existing repositories and design of new repositories.

1 Introduction

As it becomes more common for organizations to describe their operations in terms of business processes, collections of business process models grow to contain hundreds or even thousands of business process models. For example, the SAP reference model contains over 600 business process models [15] and a collection of business process models for Dutch local government contains a similar number of business process models [7]. Managing such complex process landscapes is a difficult task. Typical issues arise, e.g., being able to find a particular process in a collection, managing different versions of processes and maintaining consistency when multiple people are editing the same process at the same time. In addition to that, the availability of a large collection of processes opens up new possibilities, e.g., extracting knowledge about the operations of the organization from the collection or re-using (best-practice) process fragments from the collection to design new processes.

As a consequence, software tools have been developed to help perform such tasks. These tools have been built as extensions of general database and repository systems. However, they have been specialized for storing business process models by using conceptual models, for example database schemas, that are process specific and by defining process specific interfaces. The interface could, for

example, take the form of a Web service interface or an API that has operations like ‘addProcess’ and ‘searchTask’ and by which process models can be imported or exported in process specific interchange formats like EPML or PNML [13]. We refer to such repositories as BP Model Repositories, which we define as repositories that are structured according to a process-specific conceptual model and/or that have a process-specific interface. In addition to exploiting the functionality that is commonly provided by repository and database management systems [4,14], BP Model Repositories provide functionality that is specific for repositories that contain business process models. Examples of process-specific functionality include: functionality to assist with lifecycle management of business processes, functionality to help maintain consistency between the private view on business processes (which is the view that organizations have internally on their business processes) and the public view on business processes (which is the view on those parts of business processes that companies want to make visible publicly), and functionality to assist with configuration management of business processes as they are composed of (certain versions of) sub-processes and tasks.

To provide an overview of the functionality that should be provided by BP Model Repositories, this paper analyzes and extends existing related works and provides a framework for BP Model Repositories. The contribution of this paper is as follows: it presents a framework for BP Model repositories, which consists of a management model and a reference architecture. The management model lists the functionality that can be provided by BP Model Repositories, while distinguishing between functionality that is provided by general repositories and database management systems [4,14] and functionality that is specific for repositories that contain business process models. The reference architecture presents the components that provide this functionality and their interconnections. The framework serves as a guide for the development of BP Model Repositories.

The remainder of the paper is organized as follows. Section 2 introduces existing BP Model Repositories. Section 3 presents the general BP Model Repository management model, which lists the functionality that can be provided by BP Model Repositories. Section 4 presents a reference architecture for BP Model Repositories. Section 5 concludes the paper.

2 An Overview of Existing BP Model Repositories

This section provides a brief overview of existing BP Model Repositories. Comparisons of existing BP Model Repositories based on the framework are given in our internal report [19]. There are two basic approaches to built BP Model Repositories, i.e., knowledge-based and service-based.

Knowledge-based BP Model Repositories are repositories storing knowledge related to processes. The knowledge can be from different perspectives, e.g., reference models, resources, ontologies, and execution traces. The MIT process handbook [12] and the process reuse architecture [8] take reference models as knowledge, which provide text-based descriptions for processes of different areas.

To manage business processes throughout their lifecycle, the repository for integrated process management (IPM) [5] stores different types of knowledge, e.g., resources and business rules. The semantic business process repository [11] considers ontologies as its knowledge and the ontologies include sBPMN, sEPC and sBPEL. The process variant repository (PVR) [10] provides a mechanism to monitor the executions of processes, and stores the execution traces as knowledge to improve processes.

Service-based BP Model Repositories are repositories storing processes that are defined in the context of services. The two main advantages of service-based BP Model Repositories are distribution and collaboration. For example, OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support) [16], combining the ideas of web services and hyperdatabases, to support peer-to-peer process execution by invoking services from distributed repositories. The BPMN repository architecture [17] provides a virtual service platform, which makes it easy to support collaborative scenarios between different organizations.

BP Model Repositories can be both knowledge-based and service-based at the same time. For example, Both the BPEL repository [18] and BP-Mon [3] store process models based on the web services technology; the BPEL repository stores associated metadata as knowledge, while BP-Mon stores execution traces as knowledge.

3 BP Model Repository Management Model

Although, by definition, BP Model Repositories have in common that they have a process-specific conceptual model or interface, they vary with respect to the form of that structure or interaction facility. Also, BP Model Repositories vary with respect to the functions that they provide. This section defines the possible forms of a BP Model Repository structure or interaction facility and the functions that a BP Model Repository can provide.

We consider a BP Model Repository as a specialized repository. According to Bernstein and Dayal [4], a repository is “a shared database of information about engineered artifacts produced or used by an enterprise”. Consequently, it should provide common database management services for data model creation and adaption, data retrieval, enabling data views, integrity management, access management and state management. It should also provide services that are specific for managing objects as opposed to data in general: object check-out/checkin, version management, configuration management, notification management, context management and workflow management. The functionality for general repositories, as it is summarized by Bernstein and Dayal [4] and by Sagawa [14], can be specialized and extended to develop repositories that are specific for storing and managing business process models. We developed such an extension by taking the work of Bernstein and Dayal [4] and Sagawa [14] as a starting point and specializing and extending it, based on functionality that can be observed in existing BP Model Repositories, as described in Section 2.

The resulting BP Model Repository management model is shown in table 1. It consists of three parts: the process data model, the process function model and the process management model.

3.1 Process Data Model

The process data model prescribes what kinds of business process models and related data can be stored in the BP Model Repository. It consists of the meta-model, the storage model and the index model.

The *meta-model* prescribes what information can be and must be stored in the BP Model Repository by defining the concepts that are used in the repository and the relations between those concepts. Each BP Model Repository potentially supports a large number of concepts. We classify those concepts by identifying the *process aspects* and the *process types* that are supported by a BP Model Repository. We distinguish the following process aspects.

- The activity aspect (A) contains concepts to describe the activities that are performed in the context of a process, e.g., [12].
- The control-flow aspect (CF) contains concepts to describe the control-flow relations between activities, e.g., [3,5].
- The data aspect (D) contains concepts to describe the information that is used and changed during the execution of a process, e.g., [3,5].
- The resource aspect (R) contains concepts to describe physical resources that are required to execute (activities in) a process, including human resources, e.g., [3,5].
- The authorization aspect (Au) contains concepts to describe who is authorized to perform which part of a process, e.g., [3,5].
- The organization aspect (O) contains concepts to describe the organizational structure, as it consists of people and organizational units, related to a collection of processes, e.g., [3,5].
- The strategic goals aspect (G) contains concepts to describe the hierarchy of strategic goals and to describe the relations of those goals to the processes that are meant to achieve them, e.g., [8].
- The monitoring aspect (M) contains concepts to define how the performance of a process should be monitored, e.g., [3,5].
- The management control (MC) aspect contains concepts to define the management controls that are implemented by (parts of) processes, e.g., [17].

We distinguish the following process types.

- A company specific process (C) is a process that is designed by a specific company to describe its own operations, e.g., [3,5].
- A reference process (Re) is an abstract and standard process that can be reused and adapted to develop company specific processes. If a reference process contains pre-defined configuration options, it is also called a configurable reference process, e.g., [12,8].
- A process pattern (P) is a partial process that describes a best practice summarized from former experience, e.g., [8].

- A process instance (I), or case, is an execution of a process for a customer, e.g., [3,10].
- Historical information (H) consists of logs that contain information about executions of the process instances, e.g., [3,10].

Table 1. BP Model Repository Management Model

Process data model	Process meta model	Process aspect Process type Process notation
	Process storage model	External process data model Internal process data model Process related data model
	Process index model	Process classifications Other process indices
Process function model	Storage functions	Create Delete Update Import Export
	Retrieval functions	Navigate Search Query
	Integration functions	[Depend on external tools]
Process management model	Process-specific management	Version management Configuration management Lifecycle management Process view management
	General repository management	Access management Integrity management Transaction management Checkin/out management Dispatch management Notification management Context management

The meta-model also prescribes how information that is stored in a BP Model Repository is presented to the end-user, by associating a notation with its concepts. For example, a BP Model Repository can store the information for activities and control-flow relations between those activities, but that information can be presented to the user in (structured) natural language, in a standardized graphical notation like EPC or BPMN, or in a proprietary graphical notation. It is also common for a BP Model Repository not to prescribe a notation, but focus solely on defining its conceptual model and/or interchange format, e.g., the MIT process handbook [12].

The *storage model* prescribes how the original information about the process must be technically provided to the BP Model Repository (*external data model*) and how it must be internally stored by the BP Model Repository (*internal data model*). The external and the internal model can be the same, for example each process can be stored as an XML file that is also used to exchange the process between the BP Model Repository and related tools, or they can differ, for example processes can be exchanged using XML but stored in a relational database. Other than that process related data, which is data that is used by, but not part of, the processes can be stored in the repository. Process related data includes: descriptors of web services that are used by the processes and ontologies that are used to relate terms from different processes. For example, the IBM BPEL repository [18] also stores WSDL web-service descriptors.

The *index model* prescribes the indices that are kept for process models, to allow both the user and the repository manager itself to quickly browse or search the collection of processes. An index that is commonly used is a *classification* of process models in terms of the business functions for which they are available. For example, we can classify processes into processes for: sales, procurement, production, finance and support. Subsequently, we can distinguish different classes of procurement processes, like procurement of product related materials and procurement of non-product related materials, etceteras.

3.2 Process Function Model

A BP Model Repository should support a series of basic functions to effectively manipulate the processes that it stores. We identify storage functions, retrieval functions and integration functions.

The *storage functions* are the functions to create, update and delete processes or parts of processes, by creating, updating or deleting instances of the concepts that are defined in the process meta model. In addition to that functions exist to import complete processes into the repository, using the interchange format from the external data model, and to export complete processes from the repository using that interchange format.

The *retrieval functions* can be used to obtain the required process according to some criteria. There are three methods for retrieving processes: navigate, query and search. Navigation is the method of manually scanning processes in a list, or by using a classification or some other index. Search provides the function to get processes that match criteria that are given as keywords. Query provides more advanced functions to specify search criteria using a query language, such as IPM-PQL [5] or BPMN-Q [2]. Queries and query languages can have a focus on one or more process aspects or process types. Awad distinguishes the following foci for process query languages [2]; languages that focus on retrieving (elements of) processes (company specific or reference), languages that focus on retrieving (elements of) process instances and languages that focus on retrieving (elements of) process execution history.

The *integration functions* can be used to integrate a process repository with external tools. Integration varies, depending on the types of tools a repository

integrates. In the BP Model Repositories that we have studied, we have observed integrations with the following types of tools.

- Process modeling tools, which can be used to visually create, retrieve, update and delete processes, e.g., OSIRIS [16] provides them.
- Report generators, which can be used to generate reports about (monitoring information of) processes and their properties, e.g., BP-Mon [3] and PVR [10] provides them.
- Process analysis tools, which can be used to analyze correctness, selected properties or performance of processes, e.g., IPM [5] provides them.
- Workflow engines, which can be used to execute business processes by performing activities, or notifying human resources that activities must be performed, according to the order specified by the control-flow relations. When executing a business process a process instance is created and monitoring information is generated, e.g., BP-Mon [3] and PVR [10] provide them.
- Process administration and monitoring tools, which can be used to manage or monitor the (executions of) processes, e.g., BP-Mon [3] provides them.
- Collaboration tools, which can be used to establish business collaborations based on processes in the repository, e.g., the BPMN repository architecture [17] provides them.

Within the set of BP Model Repositories that we studied, there was no strict separation with respect to what is considered internal functionality of the repository and what is considered external functionality that can be integrated with the repository. For example, query tools have been proposed as external tools [2], but at the same time tools for establishing collaborations between organizations, based on their processes, have been proposed as internal parts of the repository [17]. We made the separation between internal and external functionality above, based on what we most frequently observed in the analyzed BP Model Repositories.

3.3 Process Management Model

Advanced management functions can be subdivided into functions that are provided by general repositories and functions that are provided only by BP Model Repositories.

The *process specific management functions* are: version management, configuration management, lifecycle management and view management. Although version management, configuration management and view management are also general repository functions (or even general database functions in the case of view management), these functions have been specialized to meet process specific requirements [1,5,6,9,20]. The version management function enables multiple versions of the same process or activity to be maintained. The configuration management function makes it possible to maintain the relation between (a version of) a process and the (versions of) subprocesses and activities that it consists of. Although version and configuration management are also general

repository functions, specialized functionality is added to support requirements in the context of BP Model Repositories. For example, when a process is being executed and a new version of that process or a part of that process is created, a decision must be made as to whether the new version will be put into effect for process instances that are already running or not and, if so, for which process instances. The lifecycle management function maintains the stage in its lifecycle that a process is currently in. For example, a process can be under design, validation and current. Depending on the stage that it is, some operations can be performed on it and others cannot. For example, a new version cannot be created of a version that is under design, nor can a process that is still under validation be executed. The view management function makes it possible to create multiple views on a process. Although view management is a general database function, specialized functionality is added to support requirements in the context of BP Model Repositories. For example, it is common to keep a private view on a process, which represents the process as it is performed inside an organization. At the same time a public view (also called service) can be provided of what the behavior of the process to the outside world will be like, therewith preserving company secrets of how services are internally implemented and not bothering clients with details that do not concern them. To support the generation of the public view from the private view and to keep the two views consistent, BP Model Repository specific functionality is needed.

The *general repository management functions* are: access management, integrity management, transaction management, checkin/out management, dispatch management, notification management and context management. The access management function ensures that people only have access to the objects in the repository that they are authorized to view. The integrity management function ensures that the repository cannot get into an inconsistent state. Transaction management ensures that multiple operations on a repository can be performed in a transactional manner (i.e.: either all at once or not at all). Checkin/out management allows a user to check-out objects from the repository, therewith locking them so others cannot change them, make the desired changes and then check them in again by releasing the lock. Optionally, multiple people can be allowed to check-out an object at the same time, in which case check-in management should ensure that changes that are made to the same object by multiple people are properly merged. Dispatch management makes it possible to associate a work-order with an object, such that it is forwarded to people in the order specified in the work-order along with notes about what these people have to do with the object. (This is usually called ‘workflow management’ in repositories. We call it dispatch management to avoid the confusion with workflow tools that execute the processes in a BP Model Repository.) Notification management enables notifications to be generated in case an object in the repository is changed. The context management function allows collections of repository objects, called ‘contexts’, (also called ‘projects’ or ‘workspaces’) to be created and manipulated. Contexts can be stored persistently.

4 BP Model Repository Reference Architecture

This section presents a reference architecture for BP Model Repositories. The reference architecture is obtained by analyzing the architectures of existing BP Model Repositories, as described in Section 2, and by integrating the analysis results into a cohesive architecture. Figure 1 shows the reference architecture. It has five layers: the presentation layer, the process repository management layer, the repository management layer, the database management layer and the storage layer.

The presentation layer provides GUIs for users to interact with a BP Model Repository, so the users can easily interact with the functions provided by the repository. Not all concrete BP Model Repositories have a presentation layer.

The process repository management layer provides repository functions that are specific for BP Model Repositories. The functions are described in detail in the previous section. Although general database management systems and repositories implement general functions, such as querying and checkin/checkout, most BP Model Repositories choose to implement these functions themselves, because this allows them to at least provide a facade that applies the functions specifically to processes instead of general repository objects or database tables. The functions may, or may not, use general functionality provided by a general database management system or repository to implement the BP Model Repository-specific functions.

The repository and database management layers provide the functions that are generally provided by repository and database management systems, respectively. Most BP Model Repositories that we studied do not distinguish the repository management layer from the layer with process specific functionality. Instead, they will have a single layer that contains all management functionality. We introduce the distinction between the layers here, to clearly show that there is an architectural choice between implementing these layers in the process repository or obtaining them from general purpose repositories or database management systems.

The storage layer stores the process models, the related data and indices or classifications to enable fast querying, searching and navigation of the BP Model Repository. Process models can be stored both in an internal format, for example as rows in database tables, and in their original external format. In that case the internal format is used for fast and unified processing of process models in spite of their external format and the external format is used to maintain the relation with the original models. In most cases the storage layer is implemented by a general database management system. Relational (e.g., [12,8,3,5]), object-oriented (e.g., [17,18]) and XML databases (e.g., [8,3,5,17,18]) have all been observed in the concrete BP Model Repositories that were studied. Alternative implementations that have been observed are implementations using general repositories, of which one using a distributed repository, and an implementation in which the data is stored as files in a filesystem.

Well-defined interfaces should exist between the different layers. In most BP Model Repositories well-defined interfaces exist between the presentation

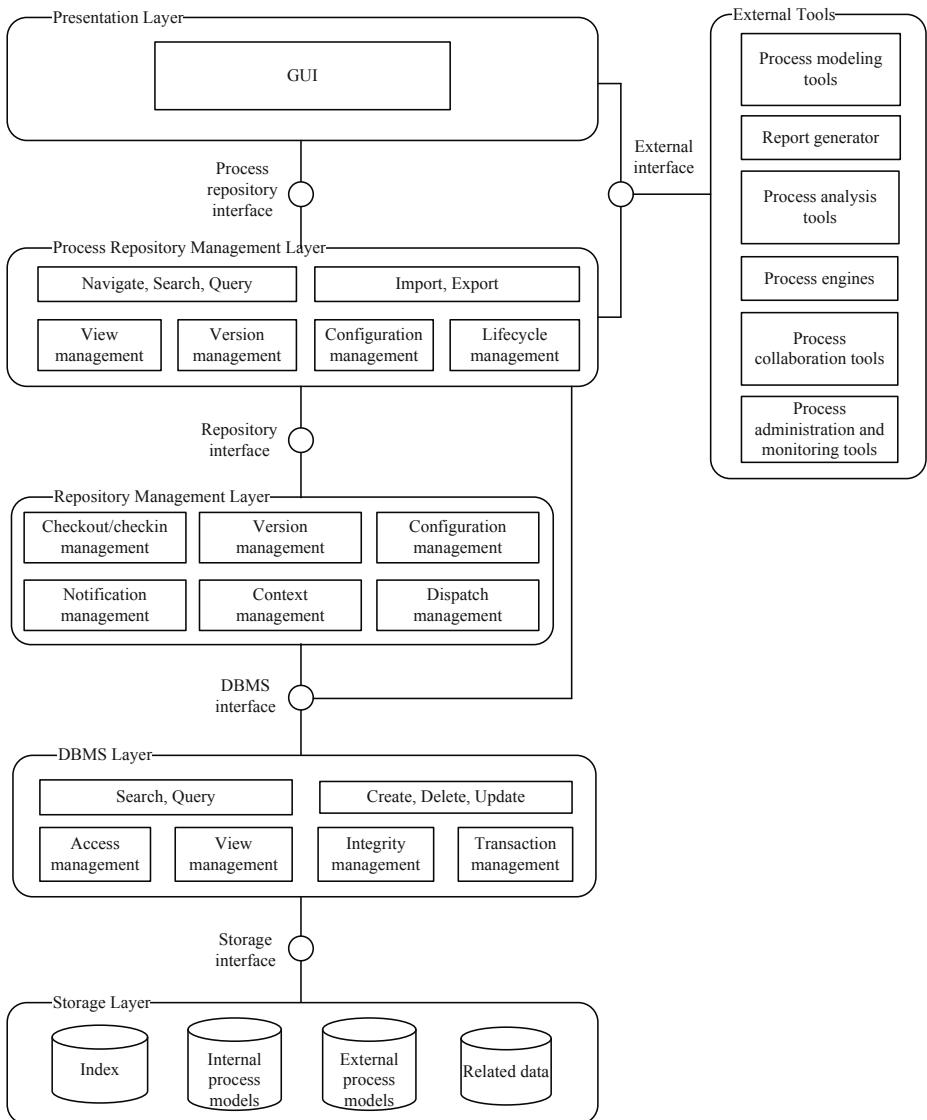


Fig. 1. The BP Model Repository Reference Architecture

layer and the process repository management layer and between the repository management layer and the database management layer. The technology that is used to implement the interfaces varies. The process repository interface can be implemented using a programming language API, but also using remote method invocation or even using web-services. The DBMS interface can be implemented using a (standard) API, but we have also observed concrete BP Model

Repositories that added an additional layer that abstracts from the storage technology that was used, to allow different storage technology to be used without having to implement the repository functions.

In addition to the interfaces between the layers, interfaces can exist between the BP Model Repository and external tools. The presence and implementation of these interfaces varies largely. However they are all defined either to interact with the presentation layer, with the process repository management layer or with both. Interaction with the presentation layer enables the BP Model Repository to open a tool from the GUI of the BP Model Repository. Interaction with the process repository management layer enables an external tool to directly invoke the functions that are provided by the BP Model Repository.

5 Conclusion

This paper defines a Business Process Model Repository (BP Model Repository) as a repository that is structured according to a process-specific conceptual model and/or has a process-specific interface. It presents a framework for BP Model Repositories, which consists of a list of functions that such repositories can provide and a reference architecture that is an abstraction of the architectures that they observe.

By observing existing BP Model Repositories, we conclude that complete repositories, as described in the framework, are not yet available. Therefore, the framework is a basis of future works. We will try to implement a complete BP Model Repository in the future.

Acknowledgement

The research reported in this paper is supported by the China Scholarship Council (CSC).

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
2. Awad, A.: BPMN-Q: A language to query business processes. In: Proceedings of EMISA 2007, Nanjing, China, pp. 115–128 (2007)
3. Beeri, C., Eyal, A., Milo, T., Pilberg, A.: BP-Mon: Query-based monitoring of BPEL business processes. SIGMOD Record 37(1), 21–24 (2008)
4. Bernstein, P.A., Dayal, U.: An overview of repository technology. In: Proceedings of VLDB 1994, Santiago de Chile, Chile, pp. 707–713 (1994)
5. Choi, I., Kim, K., Jang, M.: An xml-based process repository and process query language for integrated process management. Knowledge and Process Management 14(4), 303–316 (2007)
6. Dijkman, R.M., Quartel, D.A.C., van Sinderen, M.J.: Consistency in Multi-Viewpoint Design of Enterprise Information Systems. Information and Software Technology (IST) 50(7-8), 737–752 (2008)

7. Documentair structuurplan, <http://www.model-dsp.nl> (retrieved February 20, 2009)
8. Fiorini, S., Leite, J., Lucena, C.: Process reuse architecture. In: Dittrich, K.R., Gepert, A., Norrie, M.C. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 284–298. Springer, Heidelberg (2001)
9. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-Driven Design and Configuration Management of Business Processes. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 246–261. Springer, Heidelberg (2007)
10. Lu, R., Sadiq, S.: Managing process variants as an information resource. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 426–431. Springer, Heidelberg (2006)
11. Ma, Z., Wetzstein, B., Anicic, D., Heymans, S.: Semantic business process repository. In: Proceedings of SBPM 2007, Innsbruck, Austria, pp. 92–100 (2007)
12. Malone, T.W., Crowston, K., Herman, G.A.: Organizing Business Knowledge: The MIT Process Handbook. MIT Press, Cambridge (2003)
13. Mendling, J., Neumann, G., Nüttgens, M.: A Comparison of XML Interchange Formats for Business Process Modelling. In: Proceedings of EMISA 2004, Luxembourg, pp. 129–140 (2004)
14. Sagawa, J.M.: Repository manager technology. IBM Systems Journal 29(2), 209–227 (1990)
15. Curran, T.A., Keller, G.: SAP R/3 Business Blueprint - Business Engineering mit den R/3-Referenzprozessen. Addison-Wesley, Reading (1999)
16. Schuler, C., Weber, R., Schuldt, H., Schek, H.-J.: Peer-to-peer process execution with OSIRIS. In: Orlowska, M.E., Weerawarana, S., Papazoglou, M.P., Yang, J. (eds.) ICSOC 2003. LNCS, vol. 2910, pp. 483–498. Springer, Heidelberg (2003)
17. Theling, T., Zwicker, J., Loos, P., Vanderhaeghen, D.: An architecture for collaborative scenarios applying a common BPMN-repository. In: Kutvonen, L., Alonistioti, N. (eds.) DAIS 2005. LNCS, vol. 3543, pp. 169–180. Springer, Heidelberg (2005)
18. Vanhatalo, J., Koehler, J., Leymann, F.: Repository for business processes and arbitrary associated metadata. In: Proceedings of BPM 2006, Vienna, Austria, pp. 25–31 (2006)
19. Yan, Z., Dijkman, R.M., Grefen, P.W.P.J.: Business Process Model Repositories - Framework and Survey. BETA Working Paper WP-292, Eindhoven University of Technology, Eindhoven, The Netherlands (2009)
20. Zhao, X., Liu, C.: Version Management in the Business Process Change Context. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 198–213. Springer, Heidelberg (2007)