FINDING REUSABLE BUSINESS PROCESS MODELS BASED ON STRUCTURAL MATCHING

Han G. Woo
Le Moyjne College, 1419 Salt Springs Road, Syracuse NY 13214, U.S.A.

Keywords: Business process models, Reuse, BPMN, Structural data mining.

Abstract: Successfully integrating business processes with information systems has been a critical issue in many organizations. Such integrations should take place throughout the various stages of systems development to manage correct, traceable business process requirements. To support business process management (BPM) activities, many modeling formalisms and tools were proposed. Yet reuse of business process knowledge has been understudied although reuse practice is common, often relying on human recollection and reference models. This research proposes a tool support that assists reuse of business process models such as BPMN, EPC, and UML Activity Diagrams. In the suggested approach, the semantics of these formalisms are preserved in the conceptual graph format along with their instantiations and interrelationships. A structural data mining tool is then used to find reusable process models based on similarities in sequences of events and processes. This study can be applied to many reuse-related situations, namely retrieval of reusable process models given a problem, discovery of sequence patterns among process models, and suggesting the instances of (anti-) patterns for learning purpose.

1 INTRODUCTION

Since the management trend of business process reengineering in the 1990s, business process management (BPM) has been a critical issue in many organizations (Smith and Fingar 2003). A business process is loosely defined as a “set of partially ordered activities intended to reach a goal (Hammer and Champy 1993),” even though there is a great deal of variation among its definitions depending on which aspect of business processes is of interest: social construct, dynamic system, or machine metaphor (Lindsay et al. 2003). BPM becomes more important in the fast-changing digital economy as organizations’ business processes themselves evolve over time to keep up with competitive market pressure. The key aspects of BPM are to fully comprehend an organization’s business processes and to manage necessary changes in these processes to meet the organization’s strategies. For successful integration of the BPM practices with enterprise information systems, various business process (or workflow) modeling formalisms and tools have been suggested (Stohr and Zhao 2001; Weske et al. 2004).

As business process models have been analyzed and accumulated in various projects, reuse of business process models becomes common in practice although it often relies on human recollection and reference models (Thomas et al. 2006). System engineers or subject matter experts recall business process models that were previously constructed in a domain similar to their current project. They search relevant business process models from their memories or a repository of archived documents, and then apply retrieved models to the current context. Reusing business process models, akin to other analysis and design artifacts in software development, can bring various benefits to organizations. Reusable business process models can facilitate communications between system engineers and clients instead of starting from scratch, thereby help organizations capture correct business process requirements and identify possible improvements. In this way, they can expedite the process of business process management to quickly respond to changing business environments. In addition, organizations may obtain best-practice business processes because reusable process models have been already validated and successfully integrated in a similar domain.

G. Woo H. (2009).
FINDING REUSABLE BUSINESS PROCESS MODELS BASED ON STRUCTURAL MATCHING.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 249-255
DOI: 10.5220/0002012002490255

Exploiting potential benefits of BPM appears to have redrawn attention from researchers in recent years. It is mainly because of (1) standardization efforts for business process modeling languages such as BPMN, BPEL, and XPDL (Dreiling et al. 2008) and (2) shifted interests toward Web-based, SOA(Service-oriented architecture) business applications (van der Aalst et al. 2007). Interoperable business process model specifications and packaging them as Web services render reuse of business process not only much easier but more attractive in that there are more reusable assets available to organizations and ready for system integration with slight adaptation.

Yet reuse of business process knowledge has been understudied compared to all the progress made in BPM research in the last two decades (Hidders et al. 2005). Most studies related to process reuse place their focus on utilizing "bigger chunks" of business processes like process templates, domain reference models (Thomas et al. 2006), or interoperable services (Brambilla et al. 2006; Distante et al. 2007; O'Brien et al. 2008; Tarantilis et al. 2008) coupled with business processes, with little attention to how to find reusable business processes based on similarities among actual activity and control sequences.

This research proposes a tool support that assists reuse of business process models such as BPMN, EPC, and UML Activity Diagrams. In the suggested approach, the semantics of these specifications (e.g., event, task, sub-process, gateway, sequence, message, and data object) are preserved in a conceptual graph format, along with their instantiations and interrelationships. A structural data mining tool is then used to find reusable process models based on similarities in sequences of events, processes, and control structures. The structural matching approach can complement other reuse methods like classification based on descriptors or attributes of business processes, and domain models.

2 RELATED WORK

Many commercial tools and academic work implicitly or explicitly provide some reuse support for business processes based on reusable asset management or knowledge management perspective. OMG's RAS (Reusable Asset Specification) standard (Object Management Group 2005) provides guidelines for profiling reusable software asset. Business process models may be managed in many forms: requirements, artifacts, diagram, or services (Park et al. 2007). Another approach to business process reuse can be found in application of reference models. MIT process handbook is an example of such reference models that contains a comprehensive online library of business process knowledge (Malone et al. 2003). Thomas et al. (2006) proposes a reference model management system (RMMS) that facilitates development and management of business process reference models. Some of main ERP vendors also offer BPM tools − for example, SAP's NetWeaver and Oracle's Oracle Workflow − that provide customers with workflow or business scenario templates and process patterns. These templates are normally used for typical business processes such as order processing, those which draw on a reference model of industrial business practice.

It appears that the increasing popularity of SOA (Service-oriented architecture) Web applications has also affected BPM research streams. In SOA, business processes are bundled with service architectures and reused as a form of context-aware services independent of development technologies and platforms (Brambilla et al. 2006; Distante et al. 2007; O'Brien et al. 2008; van der Aalst et al. 2007).

The commonality among these examples of process reuse research is that the guidelines for reuse derive from similar business context, attributes, or descriptors, not detailed process sequences. Therefore, little assistance is available to business process analysts when they seek instance-level exemplars that can be applied to generate alternative business processes or to manage dynamic changes in existing processes.

A few studies in workflow management system (WfMS) research community attempt to assist in workflow reuse based on workflow sequences and control structures. van der Aalst et al. (2003) suggests generic workflow patterns. These domain-independent patterns were initially defined using control flows; the patterns have evolved over time including the observations of various perspectives in workflows: data, resource, and exception handling (N. Russell et al. 2006). Some of the basic control-flow patterns are used in this study to illustrate business process queries.

The process mining tool such as ProM (van der Aalst 2007) aims to discover reusable business processes from *a-posteriori* analysis of event logs that record activities, timestamps, roles, and related data object. It claims that the discovered patterns from process mining are more practical and realistic because it looks at "inside the process" at a very refined level.

Madhusudan et al.'s work (2004) employs case-based reasoning to support workflow modeling and design. Their framework deals with business process model management issues, from storage, retrieval, to reuse and adaptation. To find reusable workflow cases, a similarity-based case retrieval method called Similarity Flooding algorithm is used. This technique shares some characteristics with the tool support presented in this paper in that it finds matching two (query and source) directed graphs based on semantic similarity in node and edge labels and similarity in topology of the graphs. However the representation of business process models and similarity metric in Similarity Flooding are rather simpler; it does not consider detailed process elements expressed in standard modeling specifications.

3 STRUCTURAL MATCHING

This section explains an approach to reuse of business process models based on structural similarity. First, a system engineer or reuse administrator defines or collects business process models in BPMN, EPC, or UML Activity diagram notations. These models are transformed into directed conceptual graphs that consist of vertices and edges. The conceptual graphs are added into a business process library along with the initial models. As an analyst begins defining a business process model or tries to find applications of a certain pattern, the analyst can call on the tool implemented with the data mining algorithm. The algorithm searches for similar structures in the library and returns best matches. The analyst then selects the most relevant business process model and adapts it to the current analysis problem.

3.1 Structural Data Mining

The tool support suggested in this paper employs an automated relational learner called Subdue (Gonzalez et al. 2000; Joyner et al. 2001) that discovers patterns in structured data sets. In Subdue, information is stored as a graph of vertices and edges. Vertices usually refer to objects, attributes, and their values while edges represent relationships between the objects. The syntax for vertex description is <*v id label*> where *id* is a vertex number and *label is* the name of that vertex. Edges are coded with <*u id1 id2 label*> or <*d id1 id2 label*>. The former represents an undirected edge (*u*) between vertex *id1* and *id2*; the latter means a directed edge (*d*) from vertex *id1* to *id2*. Examples of the Subdue graph are shown in Figure 1.c and Table 1.

Subdue's search algorithm finds repetitive substructures called *concepts* in graphs. The search starts with a uniquely labeled vertex of a graph initializing the search queue. Following the beam search strategy, Subdue expands its search by including adjacent edges and associated vertex in all possible ways, yielding potential substructures. When a repeating substructure is found, it is replaced with a placeholder vertex pointer to its substructure, thereby compressing the whole graph. Each candidate substructure is evaluated by a compression score. The compression score is calculated by *(DL(S) + DL(G|S)) / DL(G)* where *DL(G)* stands for the description length of the input graph, *DL(S)* the description length of the substructure, and *DL(G|S)* the description length of the input graph when compressed by the substructure. This evaluation metric bases its assumption on the Minimum Description Length (MDL) principle. It states that the best concept (substructure) describes the whole data set with a minimal description length, i.e. the length in number of bits of the graph representation when compressed by the substructure (Cook and Holder 2000). When a candidate substructure is found better than others in terms of the compression ability, it is stored in the best substructure queue. Iterating this process results in a hierarchical classification lattice whose lower-level concepts are included in the higher-level concepts. The iteration can be limited by two parameters: breadth of search (*beam*) and number of expansions (*limit*.) The search terminates when it reaches a user specified limit on the number of substructures extended or when the search space is exhausted.

In Subdue, there are two important features relevant to our purpose. One is its ability to find inexact match using threshold value. It is especially useful because finding reusable business processes normally requires a certain degree of tolerance in their variations caused by different styles in authoring models and by differences in naming the same concept. In Subdue, a *threshold* value determines when two structures are similar enough to match. The analyst can set this threshold parameter from 0.0 to 1.0. The value 0.0 means a complete match and 1.0 the maximum tolerance level. The similarity metric of two structures is computed as *transformation cost / structure size*, where *transformation cost* is the number of graph transformations required to make the structures isomorphic. Two structures match when the similarity metric is less than the threshold (Cook and Holder 2000). The other feature is the capability that deals with synonyms. A list of predefined synonyms can substitute different vertex or edge labels that carry the same meaning. This functionality enables Subdue's potential to utilize the benefits of a domain ontology or lexicon.

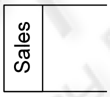
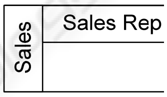
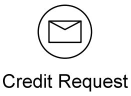
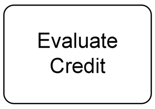
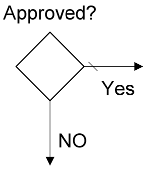
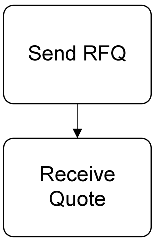
3.2 Representation of Business Process Models

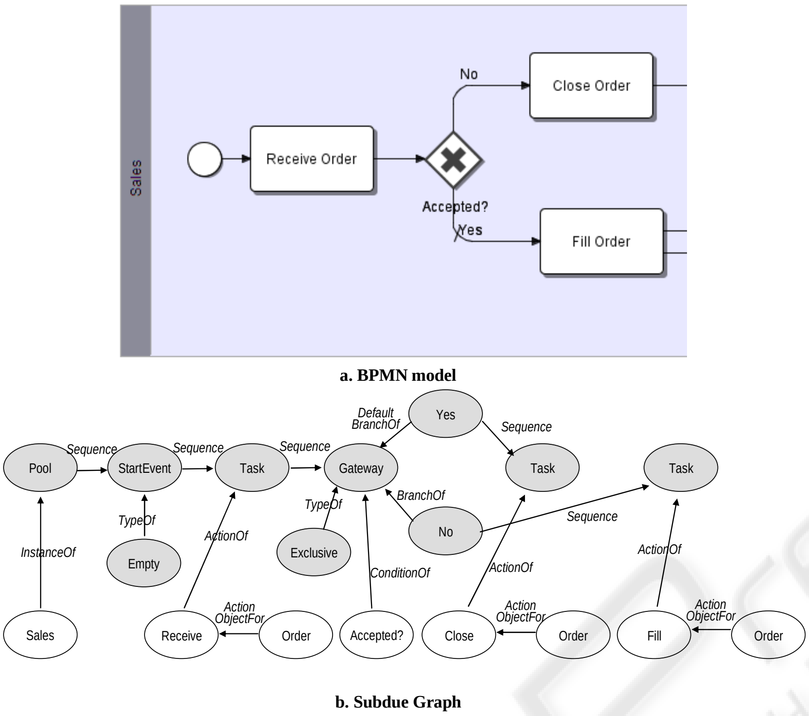
As described in the previous section, business process models need to be represented as directed conceptual graphs in order for the structural data mining algorithm to find a similar match. The transformation of business process models into conceptual graphs takes place at two different abstraction levels: metamodel and instance level. Since key elements in metamodel types in BPMN, EPC models, and UML activity diagrams share similar semantics, converted process graphs can be used together regardless of differences in the modeling notations.

Let us consider an example of BPMN models. BPMN core semantics include swimlanes, events, activities, gateways, and other artifacts (Object Management Group 2008). Table 1 summarizes the representation scheme for BPMN notations. These metamodel elements are coded with vertices in a conceptual graph. Each vertex has a label like *Pool, Lane, StartEvent, IntermediateEvent, EndEvent, Task, SubProcess, Gateway, DataObject,* etc. The connecting objects such as sequence flows and message connectors are coded as edges between two vertices that represent these metamodel elements. A pool or data object instance also becomes a vertex with its name as a vertex label. Then the instance and its metamodel vertex are connected with a directed edge labeled *InstanceOf.* A task instance has two vertices describing the nature of the task with a verb and an object. They are linked with *ActionOf* and *ActionObjectFor* edges respectively. Gateway instances are coded in a similar fashion. For example, if there is an exclusive (XOR) gateway with two branches, a gateway name becomes a vertex pointing to the metamodel vertex *Gateway* with an edge *ConditionOf.* The two branches are represented as vertices and connected with edges, *DefaultBranchOf* or *BranchOf.* Figure 1 illustrates a detailed example of the transformation. Figure 1.a shows a fragment of the BPMN model, *Process Order* while Figure 1.b illustrates the translated vertices and edges in the conceptual graph format. In Figure 1.b, the gray-highlighted part represents the structural information of the process, and the white-colored elements are the instance-level information. Figure 1.c is the actual Subdue text graph used in the algorithm.

The transformation coding scheme is designed with two guidelines: (1) separation between metamodel elements and instances and (2) maintaining an atomic value for each vertex and edge. These guidelines allow the search algorithm to focus more on structural aspects of business process models and to handle naming differences by populating synonyms. This way, an intermediate language between natural language and the formality of first-order logic makes it possible to perform classification, aggregation, and generalization (Greenspan and Mylopoulos 1982).

Table 1: Transformation of BPMN into Subdue Graph.

BPMN Semantics	BPMN Example	Graph Example
Pool		V 1 Pool *V 2 Sales* D 2 1 InstanceOf
Lane		V 1 Pool … V 3 Lane *V 4 Sales Rep* D 4 3 InstanceOf D 3 1 Within
MessageStartEven t		V 1 StartEvent V 2 Message *V 3 Credit Request* D 2 1 TypeOf D 3 2 newStateOf
Task		V 1 Task *V 2 Evaluate* *V 3 Credit* D 2 1 ActionOf D 3 2 ActionObjectFor
Gateway		V 1 Gateway *V 2 Approved?* *V 3 Exclusive* *V 4 Yes* *V 5 No* D 2 1 ConditionOf D 3 1 TypeOf D 4 1 DefaultBranchOf D 5 1 BranchOf
Sequence Flow		V 1 Task *V 2 Send* *V 3 RFQ* D 2 1 ActionOf D 3 2 ActionObjectFor V 4 Task *V 5 Receive* *V 6 Quote* D 2 1 ActionOf D 3 2 ActionObjectFor *D 1 4 Sequence*



a. **BPMN model**

b. **Subdue Graph**

% Pool
V 1 Pool
V 2 Sales
D 2 1 InstanceOf
% Event
V 3 StartEvent
V 4 Empty
D 4 3 TypeOf
% Task (Receive Order)
V 5 Task
V 6 Receive
V 7 Order
D 6 5 ActionOf
D 7 6 ActionObjectFor
% Gateway
V 8 Gateway
V 9 Exclusive
V 10 Accepted?
V 11 Yes
V 12 No
D 9 8 TypeOf
D 10 8 ConditionOf
D 11 8 DefaultBranchOf
D 12 8 BranchOf
% Task (Close Order)
V 13 Task
V 14 Close
V 15 Order
D 14 13 ActionOf
D 15 14 ActionObjectFor
...
% Sequence Flow
D 1 3 Initiate
D 3 5 Sequence
D 5 8 Sequence
D 11 13 Sequence
...

c. **Graph Text**

Figure 1: Example of Transformation from BPMN to Subdue Graph.

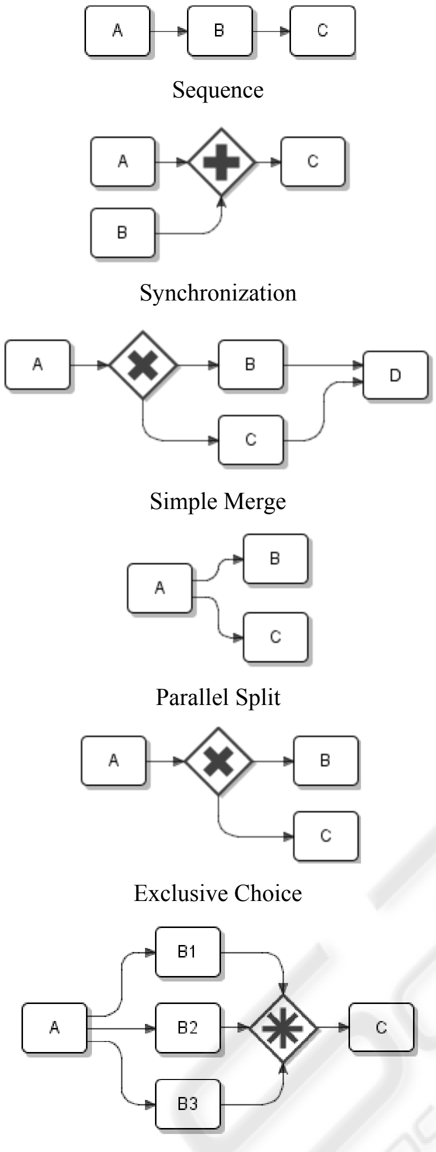
4 PRELIMINARY EVALUATION

To demonstrate the feasibility of the tool support suggested in the paper, a simple evaluation was taken with a relatively small process library. It consists of 37 partially or fully completed business process models in BPMN notations, yielding 1089 vertices and 1151 edges. The business process models were collected from the specification documents, examples available on related Web sites, and textbooks.

Since the main purpose of this case study is to see whether the tool can assist in finding similar business process structures, seven queries were presented to the library. The queries are borrowed from the workflow patterns in (van der Aalst et al. 2003), specifically *Sequence*, *Parallel Split*, *Synchronization*, *Exclusive Choice*, *Simple Merge*, and *N out of M Join*. These query models are shown in Figure 3. In addition, the model in Figure 2.a (part of order processing) is included for a more complicated query.

For each query, the threshold value is initially set as 0.0 and then incremented by 0.1 until 0.9. Table 2 summarizes the retrieval results for the queries. In each trial, relevant, best matched graphs were retrieved at the threshold between 0.3 and 0.9. For simple queries like *Sequence, Parallel Split*, and *Synchronization*, the tool was able to find similar business process models at the relatively low threshold values. Since the query graphs used simple labels such as A, B, B1, etc., there was no exact match found. In the first query Sequence, there are too many instances found at the threshold > 0.7, suggesting almost all sequential chains in the library. For the complicated queries, *N out of M Join* query fins only one match at 0.8. The retrieved process fragment contains an exclusive gateway instead of a complex gateway. For *Process Order* query, the completed business process model was intentionally prepopulated with a few modifications on vertex and edge labels, and it was found at 0.8.

The results of the preliminary evaluation suggest that structural matching technique can be applied to find relevant, reusable business process models. Yet, in order for the tool to be practical, each search must be tuned with a proper threshold. It should be also noted that for complicated queries with a high threshold, the computation time may exceed more than a minute in a personal computer CPU environment because the algorithm itself is polynomial. This concern can be resolved by adjusting other search options such as limit, beam, number of vertices in a structure, etc.



Sequence

Synchronization

Simple Merge

Parallel Split

Exclusive Choice

N out of M Join

Figure 2: Workflow Patterns used as Queries.

Table 2: Query Results.

Query	Best match found at threshold	Number of Instances
Sequence	0.3 – 0.6	> 45
Parallel Split	0.3 – 0.7	> 23
Synchronization	0.3 – 0.7	> 7
Exclusive Choice	0.4 – 0.7	> 11
Simple Merge	0.6 – 0.8	> 9
N out of M Join	0.8 – 0.9	1
Process Order	0.7 – 0.9	1

5 DISCUSSION

This research proposes automated tool support for business process reuse that exploits rich semantics of business process modeling formalisms. Business process models are translated as conceptual graphs that comprise vertices and edges. The coding scheme is quite flexible and extensible; it can express core semantics of existing business process specifications. By applying the structural matching technique, the tool support can deal with a certain degree of informality inherent in business process models while looking at similar sequence patterns. This study can be applied to many reuse-related situations, namely retrieval of reusable process models given a problem, uncovering sequence patterns among process models, and suggesting the instances of (anti-) patterns for learning purpose.

Future work includes developing the prototype of the tool support, validating its effectiveness in a field or lab experiment setting. The prototype of the tool support is under development as a plug-in to SOA Tools Platform on Eclipse (http://www.eclipse.org/stp/bpmn.) This approach is being tested with a bigger collection of business process models for ERP systems in order to support ERP configuration with business process modeling. Additional methods for search tuning also need to be explored to increase search performance, including ontological support of important concept matching and Subdue's supervised learning feature.

REFERENCES

Moore, R., Lopes, J., 1999. Paper templates. In *TEMPLATE'06, 1st International Conference on Template Production*. INSTICC Press.

Smith, J., 1998. *The book*, The publishing company. London, 2nd edition.

Brambilla, M., Ceri, S., Fraternali, P., and Manolescu, I. "Process modeling in Web applications," *ACM Transactions on Software Engineering Methodologies.* (15:4) 2006, pp.360-409.

Cook, D.J., and Holder, L.B. "Graph-Based Data Mining," *IEEE Intelligent Systems* (15:2) 2000, pp.32-41.

Distante, D., Rossi, G., and Canfora, G. "Modeling business processes in web applications: an analysis framework," in *Proceedings of the 2007 ACM symposium on Applied computing*, ACM, Seoul, Korea, 2007, pp.1677-1682.

Dreiling, A., Rosemann, M., van der Aalst, W.M.P., and Sadiq, W. "From conceptual process models to running systems: A holistic approach for the configuration of enterprise system processes," *Decision Support Systems* (45:2), 2008/5 2008, pp.189-207.

Gonzalez, J.A., Jonyer, I., Holder, L.B., and Cook, D.J. "Efficient Mining of Graph-Based Data," *Proceedings of the AAAI Workshop on Learning Statistical Models from Relational Data*, 2000, pp.21-28.

Greenspan, S.J., and Mylopoulos, J. "Capturing More World Knowledge in the Requirements Specification," *Proceedings of the Proceedings of the 6th International Conference on Software Engineering*,, IEEE CS Press, Los Alamitos, CA, 1982, pp.225-234.

Hammer, M., and Champy, J. *Re-engineering the Corporation; A Manifesto for Business Revolution*, Harper Business, New York, 1993.

Hidders, J., Dumas, M., Aalst, W.M.P.v.d., Hofstede, A.H.M.t., and Verelst, J. "When are two workflows the same?" in *Proceedings of the 2005 Australasian symposium on Theory of computing,* Australian Computer Society, Inc., Newcastle, Australia, 2005 pp.3-11

Joyner, I., Cook, D.J., and Holder, L.B. "Discovery and Evaluation of Graph-Base Hierarchical Conceptual Clusters," *Journal of Machine Learning Research* (2) 2001, pp.19-43.

Lindsay, A., Downs, D., and Lunn, K. "Business processes--attempts to find a definition," *Information and Software Technology* (45:15), 2003, pp.1015-1019.

Malone, T.W., Crowston, K., and Herman, G.A. (eds.) *Organizing Business Knowledge: The MIT Process Handbook*. MIT Press, Cambridge, MA, 2003.

N. Russell, A.H.M., ter Hofstede, W.M.P., Aalst, v.d., and Mulyar, N. "Workflow Control-Flow Patterns: A Revised View.," BPM Center Report BPM-06-22, BPMcenter.org, 2006.

Object Management Group, Reusable Asset Specification, V 2.2, 2005.

Object Management Group, Business Process Modeling Notation, V1.1, 2008.

O'Brien, L., Brebner, P., and Gray, J. "Business transformation to SOA: aspects of the migration and performance and QoS issues," in *Proceedings of the 2nd international workshop on Systems development in SOA environments*, ACM, Leipzig, Germany, 2008, pp.35-40.

Park, S., Park, S., and Sugumaran, V. "Extending reusable asset specification to improve software reuse," in *Proceedings of the 2007 ACM symposium on Applied computing*, ACM, Seoul, Korea, 2007, pp.1473-1478.

Smith, H., and Fingar, P. *Business Process Management: The Third Wave*, Meghan-Kiffer Press, Tampa, FL, 2003.

Stohr, E.A., and Zhao, J.L. "Workflow Automation: Overview and Research Issues," *Information Systems Frontiers* (3:3), 2001, pp.281-296.

Tarantilis, C.D., Kiranoudis, C.T., and Theodorakopoulos, N.D. "A Web-based ERP system for business services and supply chain management: Application to real-world process scheduling," *European Journal of Operational Research* (187:3), 2008, pp.1310-1326.

Thomas, O., Horiuchi, M., and Tanaka, M. "Towards a reference model management system for business engineering," in *Proceedings of the 2006 ACM symposium on Applied computing*, ACM, Dijon, France, 2006, pp.1524-1531.

van der Aalst, W.M.P. "Trends in Business Process Analysis: From Verification to Process Mining," *Proceedings of the the 9th International Conference on Enterprise Information Systems (ICEIS 2007)*, Medeira, Portugal, 2007, pp.12-22.

van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F., and Verbeek, E. "Business process management: Where business processes and web services meet," *Data & Knowledge Engineering* (61:1), 2007/4 2007, pp.1-5.

van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., and Barros, A.P. "Workflow Patterns," *Distributed and Parallel Databases* (14:3), 2003, pp.5-51.

Weske, M., van der Aalst, W.M.P., and Verbeek, H.M.W. "Advances in Business Process Management," *Data & Knowledge Engineering* (50:1) 2004, pp.1-8.