

# Integration of Activity Modeller with Bayesian network based recommender for business processes\*

Szymon Bobek , Grzegorz J. Nalepa, Olgierd Grodzki

AGH University of Science and Technology,  
al. A. Mickiewicza 30, 30-059 Krakow, Poland  
{szymon.bobek,gjn}@agh.edu.pl

**Abstract** Formalized process models help to handle, design and store processes in a form understandable for the designers and users. As model repositories often contain similar or related models, they should be used when modelling new processes in a form of automated recommendations. It is important, as designers prefer to receive and use suggestions during the modelling process. Recommendations make modelling faster and less error-prone because a set of good models is automatically used to help the designer. In this paper, we describe and evaluate a method that uses Bayesian Networks and configurable models for recommendation purposes in process modelling. The practical integration of the recommendation module with a Activity Modeller tool is also presented.

## 1 Introduction

Processes are one of the most popular methods for modelling flow of information and/or control within a sequence of activities, actions or tasks. Among many notations that allow to define and build business process diagrams, the Business Process Modeling Notation (BPMN) [1] is currently considered as a standard. BPMN is a set of graphical elements denoting such constructs as activities, splits and joins, events etc. These elements can be connected using control flow and provide a visual description of process logic [2]. Thus, a visual model is easier to understand than textual description and helps to manage software complexity [3].

Several visual editors were developed to support design of business processes in BPMN, one of which is Activity Modeller <sup>1</sup>. It is a web modeller component that is available as part of the Activiti Explorer web application. The Modeller is a fork of the Signavio Core Components project <sup>2</sup>. The goal of the Activiti Modeller is to support all the BPMN elements and extension supported by the Activiti Engine – a Java process engine that runs BPMN 2 processes natively.

Although visual editors like Activity provide support for building and executing business processes, this support does not include design recommendations. By recommendation we mean suggestions that the system can give to the designer to improve the design process both in terms of quality and time.

---

\* The paper is supported by the *Prosecco* project.

<sup>1</sup> See <http://activiti.org/>

<sup>2</sup> See <http://www.signavio.com/>

Three different types of recommendations can be distinguished depending on the subject of recommendation process [4]. These types are:

- structural recommendations – that allows to suggest structural elements of the BP diagram, like tasks, events, etc,
- textual recommendations –that are used to allow suggestions of names of elements, guard conditions, etc.
- attachment recommendations –that allows to recommend attachments to the BP in a form of decision tables, links, service tasks, etc.

In this paper we focus on structural recommendation, that allows for automated generation of suggestions for the next (forward recommendation), previous (backward recommendation) or missing elements (autocompletion) of the designed BP. Such recommendations improves time needed to build new business process and prevents user from making most common mistakes. What is more, such suggestions allow the designer to interactively learn best practices in designing BPMN diagrams as this practices are encoded into the recommendation model.

In this paper we present the implementation and evaluation of the method for structural recommendation of business processes that uses Bayesian networks and configurable processes. The work presented in this paper is part of the Prosecco project<sup>3</sup>. The objective of the project is to provide tools supporting the management of Small and Medium Enterprises (SMEs) by the introduction of methods for declarative specification of business process models and their semantics. The work described in this article is a continuation of our previous research presented in [5].

The rest of the paper is organized as follows. In Section 2 related work is presented and motivation for our research was stated. Section 3 describes briefly the recommendation method developed. A prototype implementation of the recommendation module, and its integration with Activity Modeller in Section 4. This section provides also an evaluation of the method on a real-case scenario. Section 5 provides summary of the research and open issues that are planned to be solved in a future work.

## 2 Related work and motivation

As empirical studies have proven that users prefer to receive and use suggestions during modelling processes [6], several approaches to recommendations in BP modelling have been developed. They are based on different factors such as labels of elements, current progress of modelling process, or additional pieces of information like process descriptions or annotations.

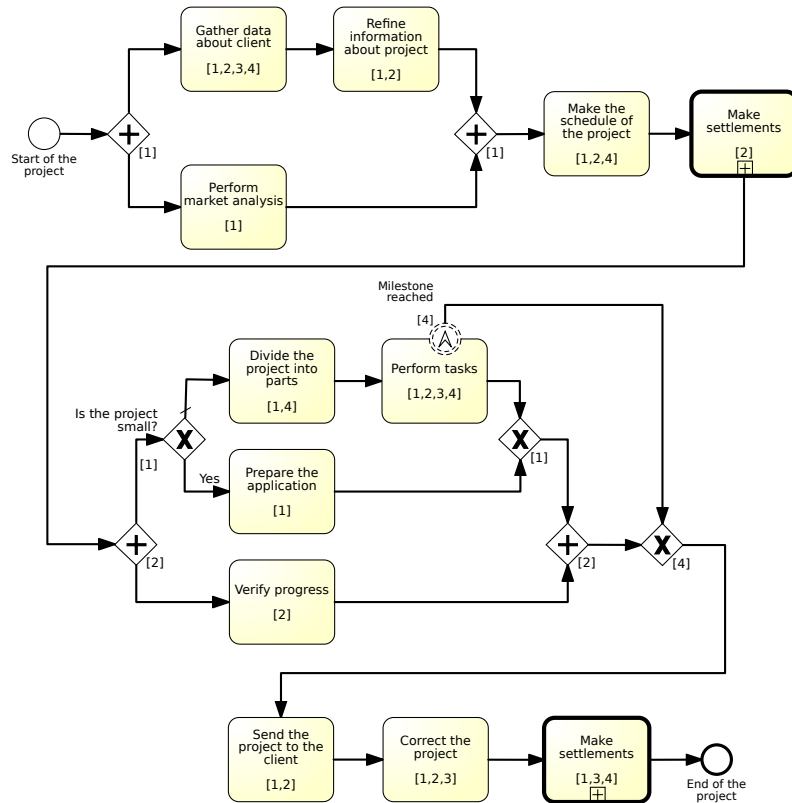
Among attachment recommendations, support with finding appropriate services was proposed by Born et al. [7] and Nguyen et al. [8]. Such a recommendation mechanism can take advantage of context specified by the process fragment [8] or historical data [9]. Approaches that recommend textual pieces of information, such as names of tasks, were proposed by Leopold et al. [10] and extended in [11].

In the case of structural recommendations, Kopp et al. [12] showed how to auto-complete BPMN fragments in order to enable its verification. Although this approach

---

<sup>3</sup> See <http://prosecco.agh.edu.pl>

does not require any additional information, it is very limited in the case of recommendations. The more useful existing algorithms are based on graph grammars for process models [13,14], process descriptions [15], automatic tagging mechanism [16,6], annotations of process tasks [17] or context matching [18].



**Figure 1.** Configurable Business Process [5]

Case-based reasoning for workflow adaptation was discussed in [19]. It allows for structural adaptations of workflow instances at build time or at run time, and supports the designer in performing such adaptations by an automated method based on the adaptation episodes from the past.

The work presented in this paper is a continuation of our previous research presented in [5]. We use Bayesian networks (BN) for recommendation purposes. In this approach a BN is created and learned based on a configurable business process. The motivation for the current work was to evaluate the methods developed in previous research. Therefore this paper focuses on the issues of matching Bayesian network to business processes to allow probabilistic recommendation queries. The BN learning was presented in our previous work and is beyond the scope of this paper. For the eval-

uation environment we decided to use Activity Modeller<sup>4</sup>, which is part of one of the most widely used software bundle for designing and executing BPMN models.

In the following section we present a short overview of the recommendation method, that uses BN for structural recommendations. It also describes an algorithm for mapping BPMN process elements to random variables of Bayesian network.

### 3 Bayesian network based recommendations

Bayesian Network [20] is an acyclic graph that represents dependencies between random variables and provide graphical representation of the probabilistic model. This representation serves as the basis for compactly encoding a complex probability distribution over a high-dimensional space [21].

In the subject of structural recommendations with BN approach, the random variables are BP structural elements (tasks, events, gates, etc). Connections between the random variables are obtained from the configurable process, that captures similarities between two or more BP models and encapsulates them within one meta-model. For the configuration model example, see Figure 1.

The transformation from a configurable model to a BN model is straightforward. Each node in a configurable process has to be modeled as a random variable in BN. Therefore, each node in a configurable process is translated into a node in the network. The flow that is modeled by configurable process represents dependencies between nodes. These dependencies also can be translated directly to the BN model (See Figure 2).

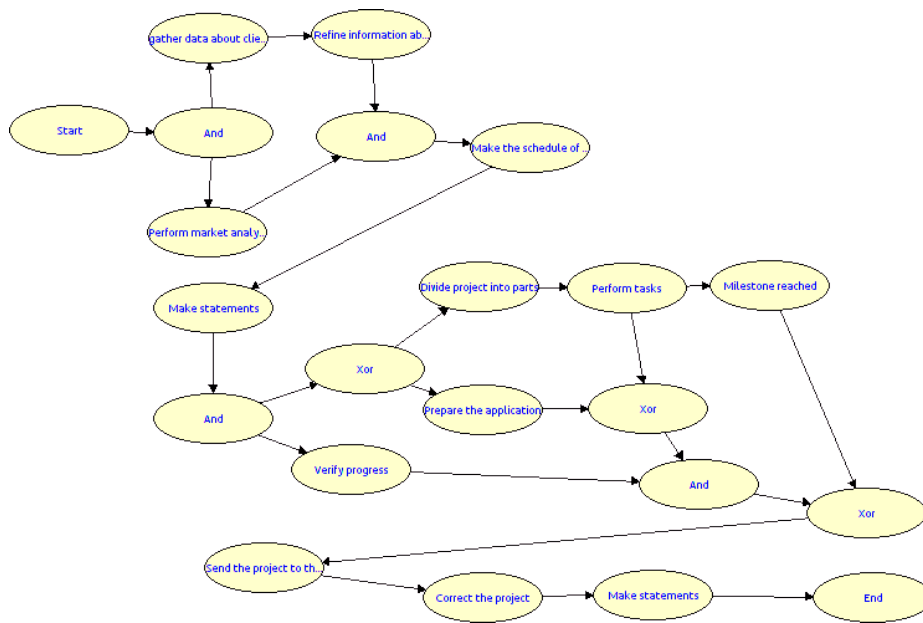
The BN network obtained from the configurable model encodes just the structure of the process. To allow querying the network for the recommendations it is necessary to train it. The comprehensive list and comparison of them can be found in [22]. For the purpose of this paper, we use the Expectation Maximization algorithm to perform Bayesian network training. The software we used to model and train our network is called Samiam<sup>5</sup>. The training data was a configurable process serialized to a CSV file. Each column in the file represents a node in configurable process, whereas each row represents a separate process model that was used to create the former.

#### 3.1 Querying the model for recommendations

We defined three different structural recommendation modes that include forward recommendations, backward recommendation and autocompletion [5]. So far we successfully implemented and evaluated forward recommendations that allows for automated generation of suggestions for the next element of the designed BP. Although the backward recommendations and autocompletion scenarios are not presented in the paper, the overall algorithm remains the same for all three scenarios. The difference between them lays on the implementation rather than conceptual level, and therefore they were skipped for the sake of clarity and simplicity. In this section we describe details of the aforementioned forward recommendation algorithm.

<sup>4</sup> <http://activiti.org>

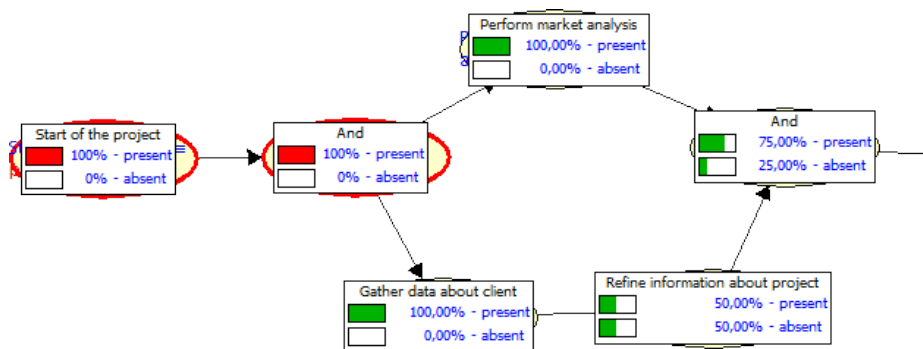
<sup>5</sup> See: <http://reasoning.cs.ucla.edu/samiam>.



**Figure 2.** Bayesian Network representing the configurable process from Figure 1

Figure 3 describes a possible query for forward recommendation. The red circle denotes observed states (so called evidence) that represents BPMN elements that were already included by the designed in the model. In the case presented in the Figure 3, the only observed evidence is a *Start* element. The remaining circles denotes possible configuration of BPMN blocks with probabilities assigned to them. For instance probability, that the block *Perform market analysis* will be present in the model is 25%.

The forward recommendation algorithm will scan the Bayesian network starting from the last observed block in a topological order, and return three blocks with the probabilities of presence in the model greater than 50%. The most challenging task in this algorithm was mapping the nodes from BPMN process to nodes in Bayesian network. This was particularly difficult because the BPMN elements are identified by the unique IDs that are different every time a new process is created.



**Figure 3.** Forward recommendations of the process for the model presented in Fig. 2

Therefore, we distinguished several possible paths for matching the BN model to the process that is designed:

- graph-based metrics, that allows to compare structures of two networks and identify areas that may correspond to the same elements [23],
- text-based metrics, that allows to compare elements based on their labels [24], and
- semantic-based comparisons, that provides more advanced matching based on the elements labels, taking into consideration semantics of the labels [25].

Because the recommendation module should work in a real-time, we decided to use the second approach which is more efficient comparing to graph-based approaches and requires less implementation effort than the third option. This choice was motivated also by the fact that BPMN elements usually have very informative labels and hence, the text comparison should give good results. As the metric for comparing nodes labels, a Levenshtein distance [26] was used. Mathematically, the Levenshtein distance between two strings  $a, b$  is given by the equation 1, where where  $1_{(a_i \neq b_j)}$  is the indicator function equal to 0 when  $a_i = b_j$  and equal to 1 otherwise.

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \quad (1)$$

Such text-based matching performs well until two or more nodes have similar or the same labels. For instance in Figure 1 there are four *And* nodes and two *Make settlements* nodes. Hence, when the user puts the block with a label *And*, the recommendation algorithm has to decide to which of the blocks in the Bayesian network it corresponds. This is performed by the neighborhood scanning algorithm.

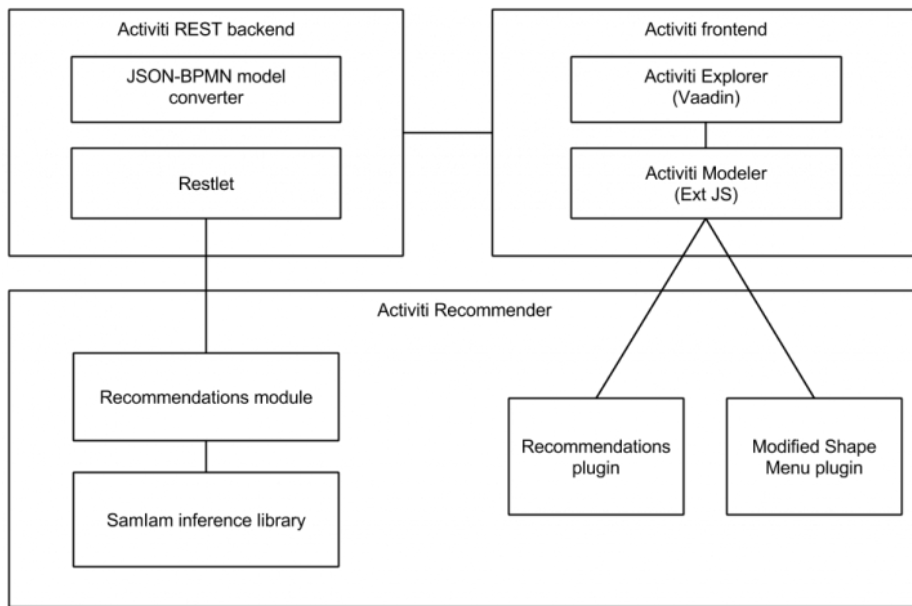
The algorithm performs a breadth-first search on the currently designed model, and try to match the neighborhood of the node from BPMN diagram to the neighborhoods of the ambiguous nodes in the Bayesian networks. The node from the Bayesian network which neighborhood matches the most of the nodes from BPMN diagram is chosen. For instance in the example from Figure 3, the user choses to include the *And* gateway in the diagram. The recommendation algorithm has to decide which *And* node from the Bayesian network presented in the Figure 2 should be treated as a reference point for the next recommendation. Because in the BPMN diagram the neighborhood of the *And* node is just one element called *Start*, the neighborhood scanning algorithm will search in the BN for the *And* node with a *Start* element as a neighbor. If the ambiguity cannot be resolved by the first level neighborhood scanning, the algorithm continues the process in a breath-search manner.

The following section presents details of the implementation of the recommendation module and presents brief evaluation of the approach.

## 4 Activity recommender

In [5] the process of transforming a configurable process into a Bayesian Network is performed manually. In order to automate this process 3 auxiliary modules have been implemented.

The first module is the converter, which creates a Bayesian Network file from a BPMN file. The second module generates a training data file based on the information about each block's occurrence in each of the processes that the configurable process is composed of. The third module takes the untrained Bayesian Network file and the training data file as input and trains the network using the EM algorithm.



**Figure 4.** Architecture of the Activity recommender extension.

The output of this three modules is an input for the Activity recommender presented in the following section.

### 4.1 Implementation

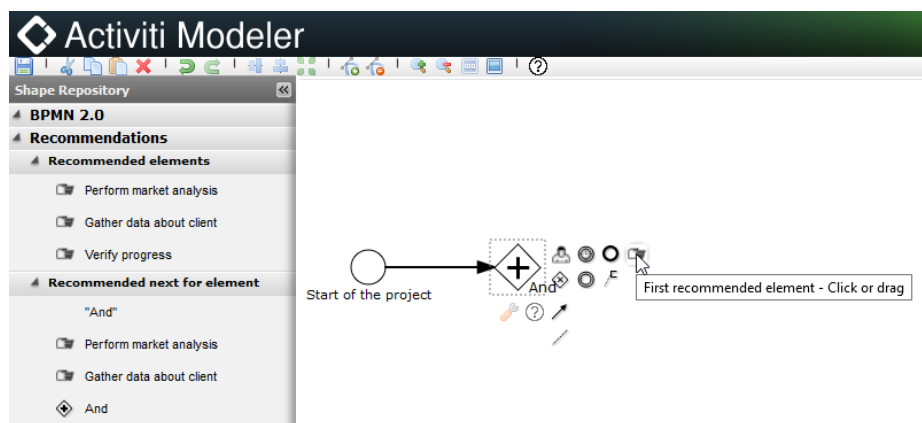
The architecture of the solution consists of four elements as depicted in Figure 4. These elements are:

- Recommendation module – an element responsible for providing recommendations based on the given BN model and evidences. It executes the recommendations queries described in Section 3.
- SamIam inference library<sup>6</sup> – A library that allows for probabilistic reasoning that is used by the recommendation module. The probabilities of occurrence of elements in designed BPMN diagram are calculated by this element.

<sup>6</sup> <http://reasoning.cs.ucla.edu/samiam/>

- Recommendation plugin – A user interface element, that presents the recommendations to the designer and allows to query the recommendation module.
- Shape Menu plugin – A plugin that is a set of icons that surround a selected block providing shortcuts for the most commonly used operations. In this case, a plugin allows to insert the recommended element just after the selected one (see Figure 5).

The Recommendation module and SamIam inference library were encapsulated into a webservice restlet to fit the Activity software architecture. Recommendation plugin and Shape Menu plugin have been implemented as a frontend plugins for Activity modeller. The communication between frontend and backend is based on the JSON exchange format.



**Figure 5.** An example of a recommendation process in Activity modeller

To better visualize this process of recommendation performed by the Activity recommender, let's assume that the previously trained Bayesian network was deployed into the Activity recommender system. When a designer queries the system for the recommendation, all the BPMN elements labels that were already placed by the designer into the model are treated as an evidence. The currently selected element is treated as a reference element for which the forward recommendation should be performed. All the evidence are packaged into the JSON format and sent to the backend, where the recommendation module performs a query to the BN and returns the recommended elements back to the frontend. In the frontend the recommendation plugin and shape menu plugin present the recommendations to the designer and the process continues.

The following section describes a brief evaluation of the described solution on the simple use-case scenario.

## 4.2 Evaluation

The evaluation of the Activity recommender was performed on the simple model presented in Figure 2. The model was learnt from the configurable process presented in Figure 1 with an auxiliary modules described briefly at the beginning of this section.



The Figure 5 presents the beginning of the design process, when only two elements are inserted into the diagram: the *Start of the project* element and the *And* gateway. The Bayesian network representing this state was depicted in the Figure 3. If the user selects the node and presses the button depicted with a question mark icon that is located on the top bar of the modeller, the recommendation query will be send to the recommendation module. The module will then perform forward recommendation starting from the element that was selected by the designer (in this case the *And* gateway). The results of the query are presented in the sidebar on the left side of the Activity modeller, and are also accessible through shape menu plugin, which is activated when the user hover the mouse over the element.

As presented in Figure 5, the recommendations are consistent with the probabilities calculated from the Bayesian network presented in Figure 3. It is worth noting, that although in the Bayesian network there exist four *And* gateways, the correct gateway was chosen for the recommendation reference point, thanks to the neighborhood scanning algorithm described in Section 3.

## 5 Summary and future work

In this paper we presented an implementation and evaluation of the structural recommendation module for BPMN diagrams. We integrated one of the most popular BPMN modeller called Ativity with our recommendation module providing practical tool for structural recommendation of BP models. We also presented an approach that supports matching the similar areas of two graphs that is based on the Levenshtein metric and neighbor scanning algorithm. The evaluation was presented on a simple use-case scenario that was part of the Prosecco project.

The future works assumes implementing remaining two recommendation modes that are: backward recommendation and autocompletion. It is also considered to compare the solution based on the Bayesian networks to the other approach that originates from phrase prediction algorithms [27].

## References

1. OMG: Business Process Model and Notation (BPMN): Version 2.0 specification. Technical Report formal/2011-01-03, Object Management Group (2011)
2. Allweyer, T.: BPMN 2.0. Introduction to the Standard for Business Process Modeling. BoD, Norderstedt (2010)
3. Nalepa, G.J., Kluza, K.: UML representation for rule-based application models with XTT2-based business rules. *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)* **22** (2012) 485–524
4. Kluza, K., Baran, M., Bobek, S., Nalepa, G.J.: Overview of recommendation techniques in business process modeling. In Nalepa, G.J., Baumeister, J., eds.: *Proceedings of 9th Workshop on Knowledge Engineering and Software Engineering (KESE9) co-located with the 36th German Conference on Artificial Intelligence (KI2013)*, Koblenz, Germany, September 17, 2013. (2013)
5. Bobek, S., Baran, M., Kluza, K., Nalepa, G.J.: Application of bayesian networks to recommendations in business process modeling. In Giordano, L., Montani, S., Dupre, D.T., eds.: *Proceedings of the Workshop AI Meets Business Processes 2013 co-located with the 13th Conference of the Italian Association for Artificial Intelligence (AI\*IA 2013)*, Turin, Italy, December 6, 2013. (2013)

6. Koschmider, A., Hornung, T., Oberweis, A.: Recommendation-based editor for business process modeling. *Data & Knowledge Engineering* **70** (2011) 483 – 503
7. Born, M., Brelage, C., Markovic, I., Pfeiffer, D., Weber, I.: Auto-completion for executable business process models. In Ardagna, D., Mecella, M., Yang, J., eds.: *Business Process Management Workshops*. Volume 17 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg (2009) 510–515
8. Chan, N., Gaaloul, W., Tata, S.: Context-based service recommendation for assisting business process design. In Huemer, C., Setzer, T., eds.: *E-Commerce and Web Technologies*. Volume 85 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg (2011) 39–51
9. Chan, N., Gaaloul, W., Tata, S.: A recommender system based on historical usage data for web service discovery. *Service Oriented Computing and Applications* **6** (2012) 51–63
10. Leopold, H., Mendling, J., Reijers, H.A.: On the automatic labeling of process models. In Mouratidis, H., Rolland, C., eds.: *Advanced Information Systems Engineering*. Volume 6741 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2011) 512–520
11. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. *Information Systems* **37** (2012) 443–459
12. Kopp, O., Leymann, F., Schumm, D., Unger, T.: On bpmn process fragment auto-completion. In Eichhorn, D., Koschmider, A., Zhang, H., eds.: *Services und ihre Komposition*. Proceedings of the 3rd Central-European Workshop on Services and their Composition, ZEUS 2011, Karlsruhe, Germany, February 21/22. Volume 705 of *CEUR Workshop Proceedings*., CEUR (2011) 58–64
13. Mazanek, S., Minas, M.: Business process models as a showcase for syntax-based assistance in diagram editors. In: *Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems*. MODELS '09, Berlin, Heidelberg, Springer-Verlag (2009) 322–336
14. Mazanek, S., Rutetzki, C., Minas, M.: Sketch-based diagram editors with user assistance based on graph transformation and graph drawing techniques. In de Lara, J., Varro, D., eds.: *Proceedings of the Fourth International Workshop on Graph-Based Tools (GraBaTs 2010)*, University of Twente, Enschede, The Netherlands, September 28, 2010. Satellite event of ICGT'10. Volume 32 of *Electronic Communications of the EASST*. (2010)
15. Hornung, T., Koschmider, A., Lausen, G.: Recommendation based process modeling support: Method and user experience. In: *Proceedings of the 27th International Conference on Conceptual Modeling*. ER '08, Berlin, Heidelberg, Springer-Verlag (2008) 265–278
16. Koschmider, A., Oberweis, A.: Designing business processes with a recommendation-based editor. In Brocke, J., Rosemann, M., eds.: *Handbook on Business Process Management 1*. *International Handbooks on Information Systems*. Springer Berlin Heidelberg (2010) 299–312
17. Wieloch, K., Filipowska, A., Kaczmarek, M.: Autocompletion for business process modelling. In Abramowicz, W., Maciaszek, L., Węcel, K., eds.: *Business Information Systems Workshops*. Volume 97 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg (2011) 30–40
18. Chan, N., Gaaloul, W., Tata, S.: Assisting business process design by activity neighborhood context matching. In Liu, C., Ludwig, H., Toumani, F., Yu, Q., eds.: *Service-Oriented Computing*. Volume 7636 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2012) 541–549
19. Minor, M., Bergmann, R., Görg, S., Walter, K.: Towards case-based adaptation of workflows. In Bichindaritz, I., Montani, S., eds.: *ICCBR*. Volume 6176 of *Lecture Notes in Computer Science*., Springer (2010) 421–435
20. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning* **29** (1997) 131–163
21. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. MIT Press (2009)
22. Neapolitan, R.E.: *Learning Bayesian Networks*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2003)

23. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In Dayal, U., Eder, J., Koehler, J., Reijers, H., eds.: *Business Process Management*. Volume 5701 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2009) 48–63
24. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. *Information Systems* **36** (2011) 498 – 516 Special Issue: Semantic Integration of Data, Multimedia, and Services.
25. Sigman, M., Cecchi, G.A.: Global organization of the wordnet lexicon. *Proceedings of the National Academy of Sciences* **99** (2002) 1742–1747
26. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* **10** (1966) 707
27. Nandi, A., Jagadish, H.V.: Effective phrase prediction. In: *Proceedings of the 33rd International Conference on Very Large Data Bases. VLDB '07, VLDB Endowment* (2007) 219–230